

MANUAL **AUTOFORMATIVO**

ARQUITECTURA DEL COMPUTADOR

Pedro Yuri Márquez Solís

Cada autor es responsable del contenido de su propio texto.

De esta edición:

© Universidad Continental S.A.C 2013

Jr. Junin 355, Miraflores, Lima-18

Teléfono: 213 2760

Derechos reservados

Primera Edición: Junio 2013

Tiraje: 500 ejemplares

Autora: Carolina Gladys Caballero Sáenz

Oficina de Producción de Contenidos y Recursos

Impreso en el Perú

en los talleres de Rebelars S.A.C

Los Bosques 555 - El Tambo - Huancayo

Fondo Editorial de la Universidad Continental

Todos los derechos reservados.

Esta publicación no puede ser reproducida, en todo ni en parte, ni registrada en o transmitida por un sistema de recuperación de información, en ninguna forma ni por ningún medio sea mecánico, fotoquímico, electrónico, magnético, electroóptico, por fotocopia, o cualquier otro sin el permiso previo por escrito de la Universidad.



INTRODUCCIÓN	7
DIAGRAMA DE PRESENTACIÓN DE LA ASIGNATURA	9
COMPETENCIA DE LA ASIGNATURA	9
UNIDADES DIDÁCTICAS	9
TIEMPO MÍNIMO DE ESTUDIO	9
UNIDAD I: EL COMPUTADOR ELECTRÓNICO DIGITAL	11
DIAGRAMA DE PRESENTACIÓN DE LA UNIDAD I	11
ORGANIZACIÓN DE LOS APRENDIZAJES	11
TEMA N° 1: INTRODUCCIÓN A LA ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS	12
1.1 Arquitectura de computadoras y Procesamiento electrónico de datos	12
1.2 Clasificación de los computadores	13
1.3 Lenguajes de Programación:	16
1.4 Computación en tiempo real:	17
TEMA N° 2: EL COMPUTADOR ELECTRÓNICO DIGITAL	18
2.1 Definiciones	18
2.2 Comparación Analógico digital	18
2.3 Estructura de un computador	19
2.4 Arquitectura Jhon Von Neuman:	20
2.5 Arquitectura hardvard	20
LECTURA SELECCIONADA N° 1	21
“Organización y arquitectura de computadoras – Diseño buscando mejores prestaciones” - Williams Stallings – Pag 40	
TEMA N° 3: PSICOFÍSICA	24
3.1 Estructura Generica de un sistema Digital	24
3.2 Los Ciclo de Captacion y Ejecución :	25
3.3 Ciclo de instrucción: Diagrama de estado	27
3.4 Interrupciones	28
LECTURA SELECCIONADA N° 2	31
Arturo Meza Huaman (10-08-2012)	
Computación con 32,64 y 128 bits. Recuperado de: https://plus.google.com/118041000851955320233/posts	
ACTIVIDAD N° 1	32
CONTROL DE LECTURA N° 1	32
GLOSARIO DE LA UNIDAD I	32
BIBLIOGRAFÍA DE LA UNIDAD I	34
AUTOEVALUACIÓN DE LA UNIDAD I	35

UNIDAD II: EL PROCESADOR	37
DIAGRAMA DE PRESENTACIÓN DE LA UNIDAD II	37
ORGANIZACIÓN DE LOS APRENDIZAJES	37
TEMA N°1: LA UNIDAD CENTRAL DE PROCESO	38
1.1 Estructura y elementos	38
1.2 LA ARQUITECTURA X86	40
1.3 Características y parámetros de los procesadores	40
LECTURA SELECCIONADA N° 1	46
Xataka.com (19 -11-2012) Los próximos Intel Atom. Recuperado de: http://www.xataka.com/componentes-de-pc/bay-trail-t-los-proximos-intel-atom	
ACTIVIDAD N° 1	47
TEMA N° 2: LA MEMORIA Y LOS BUSES	48
2.1 Definición de Memoria Ram	48
2.2 Elementos de Interconexión	48
2.3 Jerarquía de la Memoria	54
2.4 Características por las cuales se clasifican las memorias	55
2.5 Tipos de Memoria RAM	55
2.6 Memoria ROM(Read Only Memory)	57
2.7 Memoria cache	58
2.8 Memoria Virtual:	61
TEMA N° 3: ENTRADA Y SALIDA	62
3.1 LA PLACA PRINCIPAL:	62
3.2 Función de la entrada /salida	71
3.3 PERIFÉRICOS:	72
3.4 Métodos de programación de Entrada/Salida.	72
LECTURA SELECCIONADA N° 1 : DIEZ LENGUAJES EXPERIMENTALES QUE PODRÍAN IMPACTAR EN UN FUTURO	78
http://noticiasdelaciencia.com/not/5535/nuevas_y_revolucionarias_memorias_electroquimicas_deordenador/	
ACT	
ACTIVIDAD N° 1	79
TAREA ACADÉMICA N° 1	79
GLOSARIO DE LA UNIDAD II	79
BIBLIOGRAFÍA DE LA UNIDAD II	81
AUTOEVALUACIÓN DE LA UNIDAD II	61
UNIDAD III: “LENGUAJE ENSAMBLADOR”	85
DIAGRAMA DE PRESENTACIÓN DE LA UNIDAD III	85



ORGANIZACIÓN DE LOS APRENDIZAJES	85
TEMA N° 1: LENGUAJE ENSAMBLADOR Y LA ORGANIZACIÓN DEL COMPUTADOR	86
2.1 El CPU de la familia 8086:	86
2.2 Registros de 16 y 32 bits	86
LECTURA SELECCIONADA N° 1	88
Los microcontroladores y microprocesadores. Mecatrónica: Que es un microcontrolador? ingenio-upp (10-03-2008). Recuperado de: http://ingenio-upp.blogdiario.com/	
ACTIVIDAD N° 1	91
TEMA N° 2: LENGUAJE ENSAMBLADOR BÁSICO	91
1.1 Instrucciones básicas de Ensamblador y estructura de un programa	91
1.2 Directivas de datos	94
1.3 Interrupciones	95
1.4 Interrupciones más comunes	97
1.5 Empleando el Emulador 8086	103
LECTURA SELECCIONADA N° 2	105
Ventajas y desventajas del Lenguaje Ensamblador. (6/02/2012), Cómputo integrado. Recuperado de: http://computo-integrado.blogspot.com/2012/02/lenjuage-assembly.html	
ACTIVIDAD N° 1	106
CONTROL DE LECTURA N° 2	106
GLOSARIO DE LA UNIDAD	106
BIBLIOGRAFÍA DE LA UNIDAD III	107
AUTOEVALUACIÓN DE LA UNIDAD III	108
UNIDAD IV: LENGUAJE ENSAMBLADOR Y ARQUITECTURAS AVANZADAS	111
DIAGRAMA DE PRESENTACIÓN DE LA UNIDAD IV	111
ORGANIZACIÓN DE LOS APRENDIZAJES	111
TEMA N° 1: ESTRUCTURAS DE CONTROL Y SUBPROGRAMAS	112
1.1 Instrucciones de control de Flujo	112
1.2 Salto Incondicional.	113
1.3 Salto Condicional	113
1.4 Subprogramas	114
1.5 Instrucciones Aritméticas	117
1.6 Instrucciones Lógicas.	119
LECTURA SELECCIONADA N° 1 : DIEZ LENGUAJES EXPERIMENTALES QUE PODRÍAN IMPACTAR EN UN FUTURO	120
ACTIVIDAD N° 2	122
TEMA N° 2: ARQUITECTURAS AVANZADAS	122
2.1 Procesamiento Paralelo.	123



2.2 Arquitectura Superscalar	128
LECTURA SELECCIONADA N° 1 : ¿POR QUÉ MÁQUINAS VECTORIALES?.	130
JOHN PAUL SHEN y MIKKO H. LIPASTI (2005-2006) Arquitectura de Computadores: Fundamentos de los procesadores superscalares . México	
ACTIVIDAD N° 1	132
CONTROL DE LECTURA N° 1	132
GLOSARIO DE LA UNIDAD	132
BIBLIOGRAFÍA DE LA UNIDAD IV	132
AUTOEVALUACIÓN DE LA UNIDAD IV	133



INTRODUCCIÓN

Bienvenido al mundo de las Computadoras, un mundo paralelo Virtual que influye en nuestra forma de vida. Imagina nuestro modo de vida actual sin computadores, sin la posibilidad de usar los beneficios de estos equipos electrónicos, el resultado sería que no gozaríamos de las comodidades actuales, sólo por citar un ejemplo, la edición, impresión y prensa de este manual hubiera demorado bastante tiempo.

Si seguimos analizando estos beneficios, digamos el pago por el consumo de energía eléctrica, en nuestro modo de vida actual, tienes varias opciones, las cuales serían pagar desde el Internet, o con débito automático o en alguna tienda autorizada para realizar el cobro y además en un rango de fechas muy amplio.

En este manual buscamos explicarte el funcionamiento de un computador como un todo, así como las interrelaciones de sus componentes y su organización, tratando de tomar en cuenta su aplicación inmediata en el ámbito informático. Abarcamos en detalle el estudio del computador; iniciamos con la evolución de los computadores, analizando su funcionamiento, el Procesador (CPU) y su trabajo con la memoria tanto RAM y ROM, el papel de los elementos de comunicación como las Placas Principales, y los principios de trabajo de las unidades de almacenamiento. También estudiamos las arquitecturas superescalar y paralelas, analizamos sus respectivas características y sobre todo las ten-

dencias en estas ramas de la computación de alta performance.

Si en algún momento del estudio encuentras curiosidad por ampliar o profundizar, un tema, podemos enviarte material adicional para tu investigación; recuerda que vivimos en un mundo cada vez más competitivo, por lo tanto, más exigente para responder con claridad, precisión y experticia, así que, esperamos establecer una relación mancomunada para crecer y desarrollarnos conjuntamente.

A la par con este material, encontrarás diversas actividades que te ayudarán en tu auto evaluación, ponte el reto de calificarte propia y personalmente. Finalmente, las actividades propuestas intentan generar nuevos aprendizajes, a partir de la práctica; anímate a completarlas

Carolina Gl. Caballero Sáenz





DIAGRAMA DE PRESENTACIÓN DE LA ASIGNATURA

COMPETENCIA DE LA ASIGNATURA

Analiza y describe el proceso de funcionamiento de un computador considerando términos y restricciones relacionados a los recursos de hardware que se pudieran requerir en una organización.

UNIDADES DIDÁCTICAS

UNIDAD I	UNIDAD II	UNIDAD III	UNIDAD IV
El computador electrónico digital	El procesador	Lenguaje Ensamblador	Lenguaje Ensamblador y arquitecturas avanzadas

TIEMPO MÍNIMO DE ESTUDIO

UNIDAD I	UNIDAD II	UNIDAD III	UNIDAD IV
1ª Y 2ª Semana 16 horas	3ª Y 4ª Semana 16 horas	5ª Y 6ª Semana 16 horas	7ª Y 8ª Semana 16 horas







UNIDAD I: EL COMPUTADOR ELECTRÓNICO DIGITAL



DIAGRAMA DE PRESENTACIÓN DE LA UNIDAD I



ORGANIZACIÓN DE LOS APRENDIZAJES

CONOCIMIENTOS	PROCEDIMIENTOS	ACTITUDES
Tema N° 1: Introducción a la Organización Y Arquitectura De Computadoras 1.1 Arquitectura de computadoras y Procesamiento electrónico de datos. 1.2 Clasificación de los computadores. 1.3 Lenguajes de Programación. 1.4 Computación en tiempo real. Tema N° 2: El computador electrónico digital 2.1 Definiciones. 2.2 Comparación Analógica digital. 2.3 Estructura de un computador. 2.4 Arquitectura Jhon Von Neuman. 2.5 Arquitectura Harvard. Lectura Seleccionada N° 1 “Organización y arquitectura de computadoras – Diseño buscando mejores prestaciones” - Williams Stallings – Pag 40 Tema N° 3: Funcionamiento del computador 3.1 Estructura Genérica de un sistema Digital. 3.2 Los Ciclo de Captación y Ejecución. 3.3 Ciclo de instrucción: Diagrama de estado. 3.4 Interrupciones. Lectura seleccionada N°2 Arturo Meza Huaman (10-08-2012) Computación con 32,64 y 128 bits. Recuperado de: https://plus.google.com/118041000851955320233/posts Autoevaluación de la unidad I	1. Explica el funcionamiento y estructura general de un computador 2. Explica los cambios de diseño para lograr mejores prestaciones Actividad N° 1 Completar el organizador referente a la clasificación de las computadoras. 3. Analiza el funcionamiento detallado del procesamiento de instrucciones por el procesador 4. Explica el uso de los ciclos de captación y el empleo de las interrupciones Actividad N° 2 Elabora una presentación de PowerPoint que explique el ciclo de captación y ejecución que se da en un computador Control de lectura N°1 Responde un cuestionario de los temas 1, 2 y 3	Muestra interés por entender y explicar el funcionamiento de un computador



TEMA N° 1: INTRODUCCIÓN A LA ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS

1.1 Arquitectura de computadoras y Procesamiento electrónico de datos

Según William Stallings (2006), la Arquitectura de Computadores se refiere a los atributos de un sistema que son visibles a un programador; para decirlo de otra manera, a aquellos atributos que tienen un impacto directo en la ejecución lógica de un programa. Como ejemplos de atributos arquitectónicos se pueden citar el conjunto de instrucciones, el número de bits usados para representar varios tipos de datos (números, caracteres), los mecanismos de Entrada/Salida, y las técnicas para direccionamiento de memoria

Como veremos a continuación un computador requiere de elementos de entrada, Procesamiento y salida, esto dicho de forma simple, el detalle de la organización de cada uno de estos elementos son los que estudiaremos a continuación. Arquitectura de computadoras y Procesamiento electrónico de datos. Se puede definir la arquitectura de computadores como el estudio de la estructura, funcionamiento y diseño de computadores. Esto incluye, sobre todo a aspectos de hardware, pero también afecta a cuestiones de software de bajo nivel. Ahora podemos ampliar el concepto de Computador: **Dispositivo electrónico capaz de recibir un conjunto de instrucciones y ejecutarlas realizando cálculos sobre los datos numéricos, o bien compilando y correlacionando otros tipos de información, entregando finalmente los resultados**

1.1.1 Funcionamiento y estructura del computador

Las funciones básicas que debe realizar un computador se muestran en la Figura 1,

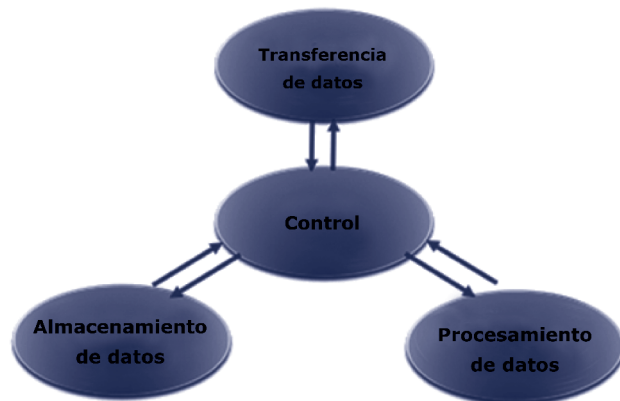


Figura Nro 1. Funciones Básicas de un computador –Stallings(2006:10)

El computador debe procesar datos, a pesar de la diversidad de datos a procesar, y del amplio rango de requisitos de proceso, existen únicamente pocos métodos fundamentales de procesado. También es esencial que el computador almacene datos, inclusive si el computador está procesando datos al vuelo es imprescindible que almacene datos al menos temporalmente, podríamos entonces empezar a hablar de almacenamiento temporal y permanente.

El computador debe ser capaz de transferir datos entre el mismo y el mundo exterior, así cuando se reciben o envían datos a un dispositivo que está directamente conectado al computador se está hablando de un proceso llamado de Entrada/Salida, en cambio si la

transferencia es a largas distancias se habla ya de Comunicación de datos. Finalmen-

¹ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p.8

te debe haber un control de estas tres funciones, El control es ejercido por el ente que proporciona al computador instrucciones.

Dentro del computador la Unidad de Control gestiona los recursos del computador en respuesta a las instrucciones que se generen. A pesar de que esta explicación parece absolutamente generalizada, pero es suficiente para tener la funcionalidad de los computadores de hoy en día, citando a [SIEW82]: “Hay, sorprendentemente, muy pocas formas de estructuras de computadoras que se ajusten a la función que va a ser llevada a cabo. En la raíz de esto subyace el problema de la naturaleza de uso general de los computadores, en la cual toda la especialización funcional se tiene cuando se programa y no cuando se diseña”

Estructura: La figura 2 es la representación de la estructura de un computador, con 4 componentes estructurales principales: Unidad Central de Proceso, Memoria Principal, Interconexión de la CPU y Elementos de entrada/salida



Figura Nro 2. Estructura del Computador –Stallings(2006:13)

1.2 Clasificación de los computadores

1.2.1 Por su potencia de procesamiento:

a. Microcomputadoras:

b. Es una computadora que posee como CPU un microprocesador, con la característica que su funcionamiento interno está basado en el uso de un microprocesador para conseguir una serie de prestaciones para cubrir la gama más baja de necesidades en el mundo de la informática en lo referente a potencia, manejabilidad, portabilidad, precio, y otros. Son equipos monousuario. Es el tipo de computador más difundido. Se pueden diferenciar a su vez dos configuraciones importantes dentro de este grupo: La Estación de Trabajo y la computadora personal.

Estación de Trabajo.

(Workstation): Es una microcomputadora de gran potencia pudiéndose considerar la gama más alta de las microcomputadoras. Permite la conexión en una red con una computadora de mayor potencia y que es utilizada en los negocios o trabajos de ingeniería y diseño gráfico de alta resolución.

Computadora Personal.

(Personal computer - PC): es una microcomputadora de fácil uso y grandes prestaciones. Es el grupo de computadoras para el cual el mercado tiene disponible la mayor gama de equipos hardware y de aplicaciones software.

Computadoras Móviles.

Dentro de las computadoras personales, atendiendo a su tamaño y prestaciones

³ <http://educacion.relacionarse.com/index.php/146304>

pueden diferenciarse otros modelos además del clásico que son:

Portátil o transportable: que es una microcomputadora con características que le adecuan para su fácil transporte, manteniendo las cualidades de una computadora personal.

Laptop: Es una computadora personal más manejable, de tamaño pequeño y gran potencia. Su peso oscila entre 1 y 2 Kilogramos. **Notebook:** Similar al laptop, pero más pequeña, con menor peso. Permite funciones de computadora personal con capacidad de cálculo y de agenda evolucionadas para aplicaciones comerciales.

Notebook: Similar al laptop, pero más pequeña, con menor peso. Permite funciones de computadora personal con capacidad de cálculo y acceso al internet, agenda y aplicaciones comerciales.

c. Minicomputadoras:

Surgen en los años sesenta como una opción para realizar adquisición de datos o controlar procesos industriales que resultaban muy costosos si se realizaban con un mainframe; se requería ser operado por cuadrillas de operadores (situación que incrementaba los costos), la minicomputadora permitía la interacción directa entre la máquina y el usuario, con la consecuente disminución de costos. Cabe mencionar que las primeras minicomputadoras fueron de 8 y 12 bits de longitud de palabra, pero a finales de los setentas ya casi todas las minicomputadoras eran de 16 bits. Con el auge de la tecnología y el surgimiento de las microcomputadoras, la minicomputadora se convirtió en un sistema multiusuario o compartido que manipulaba más dispositivos periféricos y contaba con memorias más grandes. Con la llegada de las minicomputadoras de 32 bits estas características crecieron. A la minicomputadora también se le conoce como sistema de rango medio.

d. Mainframes:

Aunque hay algunas confusiones sobre los límites entre las minicomputadoras y las computadoras mainframes (también llamadas macrocomputadoras), éstas permanecen como una clase diferente de computadoras.

Una de las diferencias reside en la velocidad de las dos clases, no obstante esto puede resultar ambiguo ya que la instrucción típica en una máquina podría hacer más que la instrucción de la otra, pero si consideramos que son comparables los conjuntos de instrucciones de las computadoras mainframe y de las minis, entonces la comparación es válida.



Figura Nro 3. MacroComputadora Honeywell- Bull DPS 7(web.okonline.co.uk)

Otra diferencia es el precio. Para dejar más claras las diferencias entre estas dos clasificaciones vea la tabla comparativa de las características de algunas computadoras contemporáneas.

Hoy en día estas máquinas realizan tareas que se creyeron que podían realizarse con una red de minicomputadoras, como es el caso del procesamiento de datos distribuido, también se emplean para soportar grandes bases de datos de empresas transnacionales y gobiernos, las cuales necesitan un almacén central de datos que se pueda manejar y controlar de manera central.

e. Supercomputadoras:

Aunque el desempeño de los mainframes de propósito general continúa mejorando sin descanso, existen algunas aplicaciones que están más allá del alcance de los mainframes contemporáneos. En disciplinas como la aerodinámica, la sismología y la física atómica, nuclear y de plasma, existen problemas que se caracterizan por la necesidad de alta precisión y un programa que realice en forma repetitiva operaciones de punto flotante sobre grandes arreglos de números, y que se engloban en la categoría conocida como simulación de campo continuo. Para manejar estos tipos de problemas, se ha desarrollado la supercomputadora máquina capaz de manejar miles de millones de operaciones de punto flotante por segundo y su costo está en el rango los millones de dólares. Una supercomputadora puede hacer aplicaciones de propósito general, pero su arquitectura esta optimizada para el tipo de cálculo numérico que involucra arreglos. La supercomputadora tiene un uso limitado y, debido a su precio, un mercado también limitado.



SuperComputador Cray xK7. <http://www.madboxpc.com>

1.2.2 Clasificación según su funcionamiento:

- a. **Computador digital:** Todo proceso tiene características particulares y el proceso de la percepción no es la excepción.
- b. **Computador analógico:** tienen semejanza con instrumentos de medida tales como amperímetro, voltímetro, termómetro, etc.; los cuales controlan la medida de una magnitud de forma continua.
- c. **Computador híbrido:** posee características de los dos anteriores, habitualmente, los cálculos los realizan de forma analógica mientras que la entrada y salida de datos se hace de modo digital. La utilización del computador híbrido es frecuente en el control de procesos industriales, en ellos las funciones principales son controladas por un computador digital, mientras que la entrada de datos y salida de resultados se hace empleando computadores analógicos conectados con el digital. Clasificación según la finalidad
- d. **De propósito general:** cuando están dedicados a resolver cualquier tipo de apli-





cación dependiendo del programa que se utilice, como por ejemplo los computadores del hogar, se emplean para procesadores de texto, graficar, visualizar películas, comunicaciones mediante el Internet.

- e. **De propósito especial:** cuando únicamente pueden resolver un tipo concreto y particular de problemas como el computador de un coche o de una lavadora. Considerando que estudiamos computadores de uso general, debemos entender que para lograr la generalidad se deben crear programas que se usen en estos computadores, el problema reside en que el computador únicamente procesa secuencias de unos y ceros, mientras que para el humano es muy complicado seguir estas secuencias, por ello es común que se emplee en la actualidad lenguajes de Programación de alto nivel.

1.3 Lenguajes de Programación:

1.3.1 Lenguaje de Programación de Bajo Nivel - Assembler:

El computador dispone de un conjunto de instrucciones que son reconocidas y ejecutadas por el procesador. Estas instrucciones se expresan, al igual que los datos, en forma digital binaria, si bien para reconocerlas mejor y evitar errores se les asignan unos nombres mnemotécnicos que permiten recordar sus funciones.

Estas instrucciones constituyen el lenguaje de máquina del computador, y suelen ser diferentes en función del fabricante del procesador. Por su estrecha relación con el hardware, a los lenguajes de máquina se les llama lenguajes de bajo nivel. En consecuencia, la tarea de escribir programas en lenguaje de máquina es tediosa y está sujeta a muchos errores.

1.3.2 Lenguaje de Programación de Alto Nivel

Ante la necesidad de escribir programas cada vez más complejos y fiables, los informáticos desarrollaron lenguajes con niveles más elevados de abstracción, que incluían operaciones y datos más complejos a los que llamaron lenguajes de alto nivel, los cuales emplean instrucciones normalmente en Idioma inglés que son relativamente fáciles de entender. Una de las grandes ventajas de estos lenguajes radica en que la traducción del programa escrito en lenguaje de alto nivel (programa fuente) al lenguaje de máquina (programa objeto) que, como recordamos, es el único que puede ser ejecutado por el computador, es automática y se realiza por un programa traductor.

En la actualidad la mayoría de los programas se escriben en lenguajes de alto nivel, reservándose los lenguajes de bajo nivel para aquellas aplicaciones donde se necesita una elevada velocidad o un acceso directo a los órganos del computador; por ejemplo, en los programas traductores o en programas de gestión del computador.

Por lo tanto en los computadores modernos se llevan a cabo procesos de compilación y traducción, los cuales normalmente no son notorios porque son realizados automáticamente por el sistema operativo, en ocasiones se realiza a una velocidad que basta con un instante para poder visualizar el programa requerido listo en pantalla, en otras ocasiones si el programa es muy pesado podemos apreciar una pantalla que nos indica que ese proceso de Compilación, traducción, carga y linkeo se está efectuando.

En la figura Nro 5, podemos observar esta secuencia:

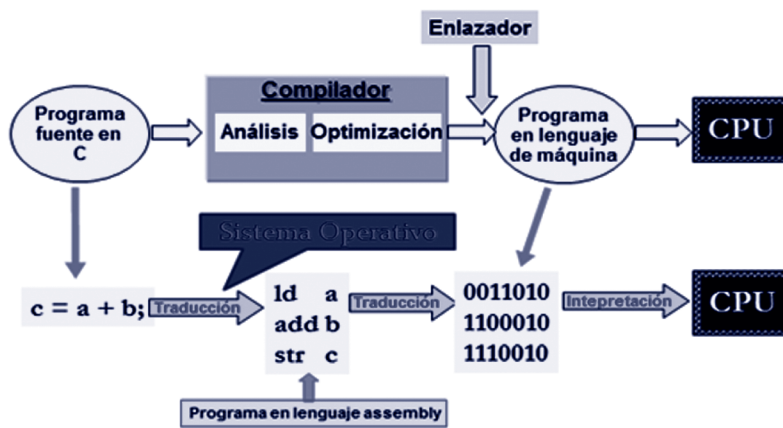


Figura Nro 5. Secuencia de compilación y traducción (D Mery: 2006)

En detalle luego de obtener el programa objeto o ejecutable, se efectúan las siguientes acciones:

- **Compilación:** Tomar cada instrucción del programa (exe), convertir a lenguaje máquina.
- **Traducir:** ubicar una zona de memoria para cada una de las variables del programa que se va a ejecutar.
- **Cargar:** Llevar las instrucciones ya compiladas en lenguaje maquina a la RAM.
- **Linkear:** es compilar las librerías que el programa va emplear.

Por qué tenemos que usar programas?:

Uno de los motivos para emplear programas es que los programas son flexibles, es decir podemos ir agregando líneas de programación, mejorando una o todas las instrucciones. Como dice Tanenbaum (2000) : “Las primeras computadoras digitales solo tenían dos niveles el ISA, en el que se efectuaba toda la programación y el de lógica digital que ejecutaba esos programas. Los circuitos del nivel de lógica digital eran complicados, difíciles de entender y construir, y poco confiables.

En 1951 Maurice Wilkes. Investigador de la Universidad de Cambridge, sugirió la idea de diseñar una computadora de tres niveles a fin de simplificar drásticamente el hardware (Wilkes, 1951). Esta máquina tendría un intérprete inmutable integrado (el microprograma), cuyo misión sería ejecutar programas de nivel ISA por interpretación. Puesto que ahora el hardware solo tendría que ejecutar microprogramas, que tienen un conjunto limitado de instrucciones”

1.4 Computación en tiempo real:

Los requisitos de la computación de tiempo real se manifiestan en la restricción temporal a la que está sometido; por ejemplo se tienen tareas periódicas de control, que vienen impuestas por un algoritmo de control, el que debe entregar respuestas a pesar de no tener completos todos los datos. En el caso de tratamiento de datos, tareas gráficas o de supervisión, tareas de comunicación las restricciones temporales no son tan estrictas y muchas veces el diseñador de las aplicaciones dispone de un margen donde elegir.

Estas restricciones temporales también implican prioridad de ejecución, siendo comúnmente las tareas dedicadas al control las más frecuentes y por lo tanto las que se deben ejecutar con mayor prioridad, interrumpiendo a todas las demás tareas.

TEMA N° 2: EL COMPUTADOR ELECTRÓNICO DIGITAL

2.1 Definiciones

Como ya habíamos mencionado un computador es una maquina electrónica que procesa información siguiendo las instrucciones de un programa registrado de acuerdo a los datos proporcionados y luego entrega estos resultados, en el caso del Computador digital la información que se procesa en el (programas, datos y resultados) esta expresada en forma digital binaria, combinando ceros y unos, a nivel de circuitos Nivel alto de Voltaje y Nivel Bajo de voltaje. En consecuencia, tanto los programas como los datos y resultados deben codificarse en este formato para poder ser procesados. Una vez obtenidos los resultados, estos tienen que ser decodificados para mostrarlos al usuario.

Un computador se compone de partes claramente diferenciadas: una física, constituida por circuitos electrónicos, y elementos electromecánicos, llamado hardware, y otra parte que constituida por los programas y datos, llamada software en inglés. Ambas partes están íntimamente relacionadas de forma que una no puede operar sin la otra y viceversa. Además debemos considerar el Firmware, que es un circuito integrado que almacena un programa, normalmente este programa requiere muy poca compilación porque está prácticamente en lenguaje Máquina.

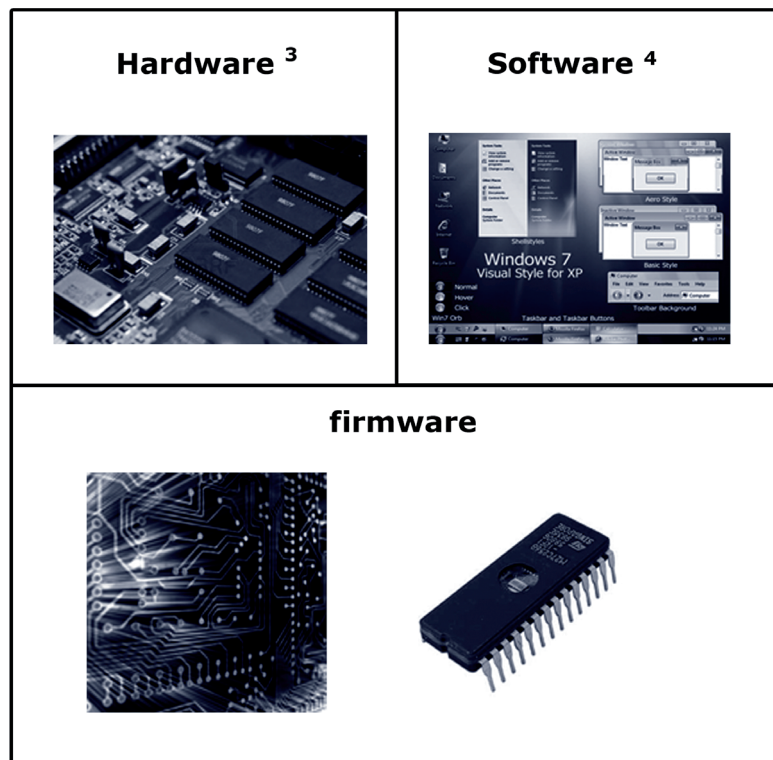


Figura Nro 6. Elementos del computador

Para poder entender mejor estos conceptos debemos definir también los términos analógico y digital, temas que se tratan a continuación

2.2 Comparación Analógico digital

2.2.1 Sistemas analógicos

² WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 57

³ Mikhail Tchkhaidze (09/12/2013). *Blue Cerca de los circuitos electrónicos. Recuperado de 123rf.COM* : http://es.123rf.com/photo_5243627_blue-cerca-de-los-circuitos-electronicos-up.html

⁴ Windows 7 Visual Style:

⁵ Guillem(10/12/2013). *Windows*. *Recuperado de:* <http://www.mastermagazine.info>

Los sistemas analógicos están formados por un conjunto de componentes que interactúan entre sí para lograr el mismo objetivo o utilidad de ese sistema. Un sistema para que sea analógico debe usar variables de entrada y de salida analógicas. Es decir se emplean valores continuos como 2.0, 2.3434353453, 100002323, 5.00001, lo que luego provoca que la transmisión y copia de estos valores no sea exacta.

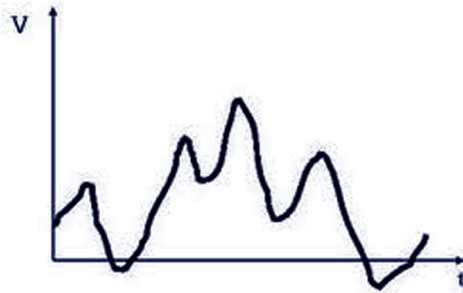


Figura Nro 7. Gráfico V/t de señal analógica

2.2.2 Los Sistemas Digitales (SD)

Los sistemas digitales se caracterizan porque sus variables de entrada y salida sólo tienen dos estados posibles. Estos estados pueden tomar los siguientes valores:

Verdaderos y falsos
Alto Bajo
Cerrado Abierto
5 voltios 0 voltios
1 0



Figura Nro 8. Gráfico señal digital

2.3 Estructura de un computador

El computador es un sistema electrónico compuesto de circuitos de alta complejidad y elevada escala de integración, cuya finalidad es efectuar una secuencia de operaciones mediante un programa previamente establecido

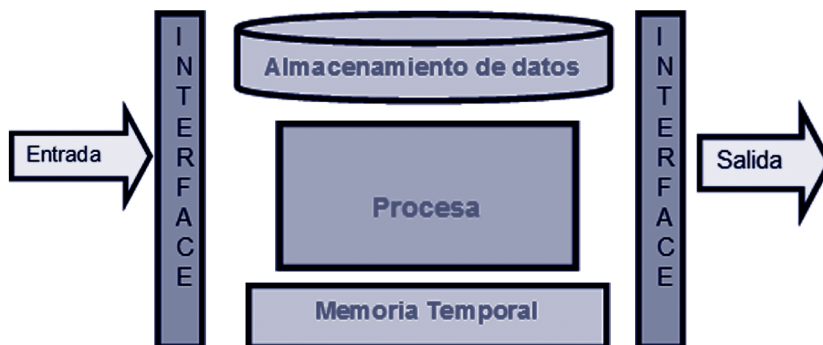


Figura Nro 9. Estructura funcional del Computador.

Con respecto a la imagen el computador mediante sus elementos de Entrada recibe normalmente señales analógicas las cuales deben ser transformadas y adecuadas al idioma del Procesador, en este caso a una representación en binario. Las interfaces luego de obtener las señales en binario, hacen que la información se sitúe en la Memoria Ram del computador, desde la cual son tomadas en forma secuencial una a una por el procesador, el cual en base a su interpretación las procesa, los resultados son dejados en la memoria ram, los cuales nuevamente son tomados por las interfaces y entregados a los elementos de salida. Finalmente el dispositivo de salida recibe la información en binario y la transforma a señales analógicas, representación que los humanos podemos perfectamente entender. Como mencionamos



Verdadero falso
Alto bajo
Cerrado abierto
5 voltios 0 voltios
1 0

antes un computador debe realizar cuatro funciones: Procesamiento de datos, almacenamiento de datos, transferencia de datos y control; Para realizar estas funciones es necesario un mecanismo de control que gobierne el flujo de datos entre los diversos módulos.

2.4 Arquitectura Jhon Von Neuman:

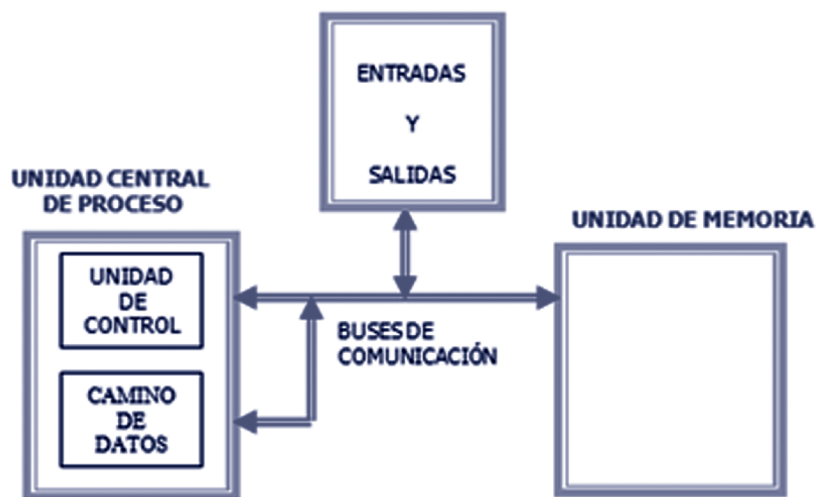


Figura Nro 10. Arquitectura Jhon Von Neumann.

La arquitectura de un computador, fue definida por uno de los mejores matemáticos de la historia John Von Neumann, que propuso una arquitectura en la cual la CPU (Unidad Central de proceso) está conectada a una única memoria donde se guardan conjuntamente instrucciones (programas) y datos (con los cuales operan estos programas). Además existe un módulo de entradas y salidas para permitir la comunicación de la máquina con los periféricos extremos que maneja el usuario.

La longitud de las instrucciones está limitada por la longitud de los datos, por lo tanto el procesador se ve obligado a hacer varios accesos a memoria para buscar instrucciones complejas. La velocidad de operación está limitada por el efecto cuello de botella, que significa que un bus único para datos e instrucciones impide superponer ambos tipos de acceso.

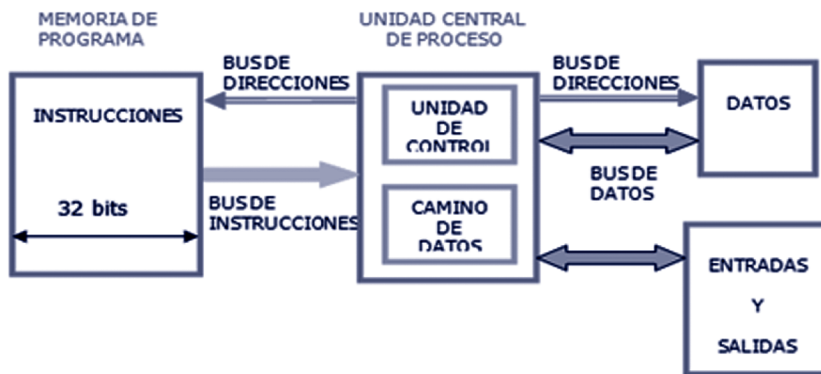
2.5 Arquitectura hardvard

Arquitectura moderna (1990 - hoy) Propone modificaciones en la arquitectura del equipo físico y mejoras y nuevas prestaciones en el tiempo lógico. Un ejemplo en el primer aspecto es la arquitectura Harvard, que está especialmente diseñada para atacar las debilidades de la arquitectura Von Neumann, la solución, se construye un procesador que está unido a dos tipos de memoria diferentes por medio de dos buses independientes.

Figura Nro 11. Arquitectura Harvard

La memoria de datos y la memoria de instrucciones son independientes, almace-

⁶ Josue Bilbao Eguia(2008). *Arquitectura de ordenadores*. España: Editorial Independiente. p. 4



nándose en ellas los datos y el programa, respectivamente. Para un procesador de tipo RISC “Computador de Juego de Instrucciones Reducido”, el conjunto de instrucciones y el bus de la memoria de programa pueden diseñarse de manera tal que todas las instrucciones tengan la misma longitud que la posición de la memoria y lo mismo con los datos. Además, como los buses de ambas memorias son independientes, la CPU puede estar accediendo a los datos para completar la ejecución de una instrucción, y al mismo tiempo estar leyendo la próxima instrucción a ejecutar. Una forma de potenciar el aislamiento entre las instrucciones y los datos es la incorporación de memorias caché ultra rápidas, que como sucede en los modelos de Intel (Pentium IV), una se encarga de guardar los datos que va a precisar la CPU y otra las instrucciones.



LECTURA SELECCIONADA N° 1

“Organización y arquitectura de computadoras – Diseño buscando mejores prestaciones”- Williams Stallings – Pag 40

⁷ Josue Bilbao Eguia(2008). *Arquitectura de ordenadores*. España: Editorial Independiente. p. 4



MEJORAS EN LA ORGANIZACIÓN Y ARQUITECTURA DE CHIPS

“Además del reto de los diseñadores de equilibrar las prestaciones del procesador con la memoria principal, y con otros componentes del computador, persiste la necesidad de aumentar la velocidad del procesador.

Hay 3 formas de conseguir incrementar la velocidad del procesador Incrementando la velocidad del hardware del procesador esto se consigue fundamentalmente disminuyendo el tamaño de las puertas lógicas del chip del procesador de forma que se puedan encapsular más compuertas, más cercanas y así incrementar la frecuencia del reloj. Con las compuertas más juntas. El tiempo de propagación de las señales se reduce significativamente. Haciendo posible un aumento de la velocidad del procesador.

Un incremento en la velocidad del reloj implica que las operaciones individuales se ejecuten más rápidamente Incrementar el tamaño y la velocidad de las cachés interpuesta entre el procesador y la memoria principal. En particular dedicando una parte del chip del procesador a la caché. El tiempo de acceso de la caché disminuye considerablemente

Haciendo cambios en la organización y arquitectura del procesador de forma que se incremente la velocidad efectiva de ejecución de una instrucción. Usualmente, esto implica utilizar el paralelismo de una forma u otra

Tradicionalmente, el factor dominante en la ganancia de prestaciones se debe al aumento de la velocidad del reloj y la densidad de la lógica. En la Figura 2.12 se muestra esta tendencia en los procesadores de Intel. Sin embargo; al aumentar la velocidad de reloj y la densidad de la lógica, una serie de obstáculos se hacen más significativos :

Potencia: a medida que la densidad de la lógica y la velocidad del reloj de un chip aumentan también lo hace la densidad de potencia (vatios/cm²). La dificultad en disipar el calor generada por la gran densidad y la alta velocidad en el chip se convierte en un problema de diseño serio

Retardo RC: la velocidad a la que los electrones pueden fluir en un chip entre transistores está limitada por la resistencia y capacidad de los hilos metálicos que los conectan; concretamente el retardo aumenta al hacerlo el producto RC. Como los componentes del chip disminuyen de tamaño los hilos que los interconectar son más finos. Aumentando su resistencia. Además los hilos están más juntos aumentando la capacitancia. **Latencia de memoria:** La velocidad de la memoria ralentiza la velocidad del procesador, como se ha visto anteriormente. Por lo tanto: es necesario hacer más énfasis en la organización y arquitectura para mejorar las prestaciones.

La Figura 2.12 destaca los principales cambios que se han hecho a lo largo de los años para aumentar el paralelismo y, por lo tanto la eficiencia computacional de los procesadores. “

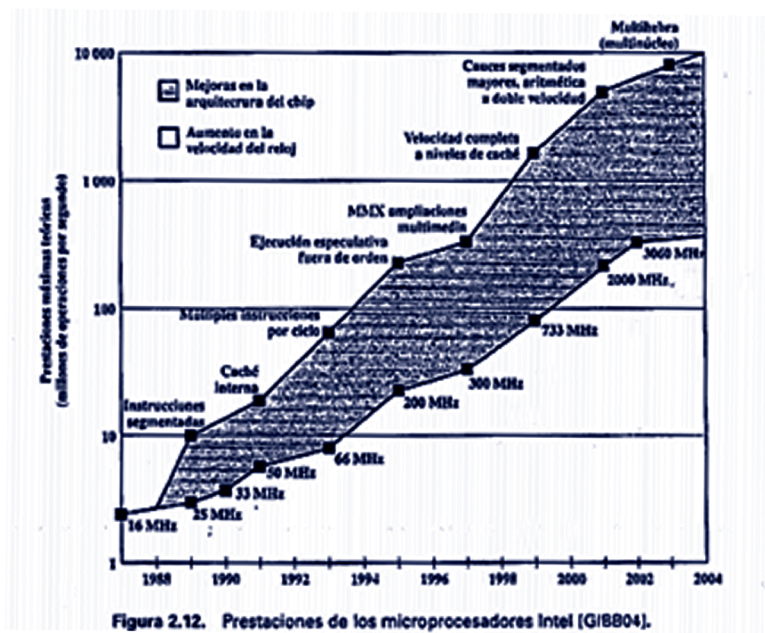


Figura Nro 12. Evolución de las prestaciones de los Microprocesadores Intel



ACTIVIDAD N° 1

Esta actividad puede consultarla en su aula virtual.

⁸ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores* –España: Pearson Education p. 57

TEMA N° 3: FUNCIONAMIENTO DEL COMPUTADOR

Es de suma importancia tener una idea de alto nivel de la arquitectura de un computador, esto nos permite entender su funcionamiento, así como determinar posibles cuellos de botella que se pudieran generar. Inclusive la mejora en las prestaciones de potencia y tolerancia a fallas se han satisfecho con cambios en el diseño de la arquitectura. Como ya se mencionó los componentes están diseñados bajo conceptos de la arquitectura de John Von Neumann, y se basa en tres conceptos clave:

- Los datos y las instrucciones se almacenan en una sola memoria de lectura-escritura.
- Los contenidos de esta memoria se direccionan indicando su posición, sin considerar el tipo de dato contenido en la misma.
- La ejecución se produce siguiendo una secuencia de instrucción tras instrucción (a no ser que dicha secuencia se modifique explícitamente).

1ra opción: Se pueden interconectar diversos componentes lógicos para realizar una determinada función. El programa resultante es **HARDWARE** y se le denomina **HARDWIRED PROGRAM** (programa cableado)

2da opción: Emplear **HARDWARE** de uso general, este realiza distintas funciones según las señales de control aplicadas. En el segundo caso, se pueden generar nuevos programas proporcionando nuevas señales de control, es decir solo es necesario cambiar el programa.

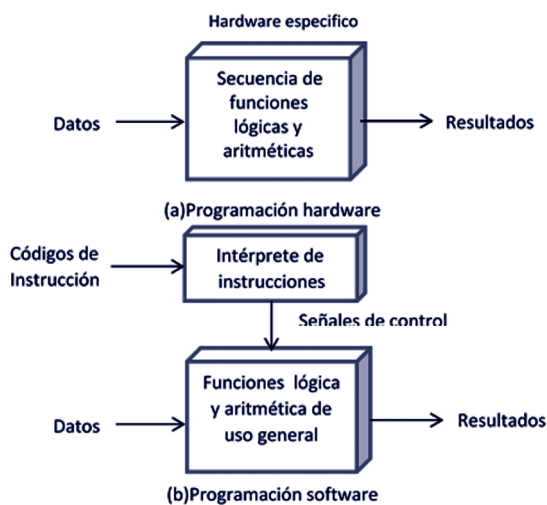


Figura Nro 13. Alternativas de hardware y software.

La pregunta sería ¿Cómo se entregan nuevas instrucciones? Simplemente un programa es: Es una secuencia de Instrucciones que procesan datos. En cada paso, se realiza una operación aritmética o lógica. Para cada paso, se requiere un conjunto de señales de control. La Solución consiste en Asociar un código específico a cada posible conjunto de señales de control, y agregar hardware que interprete cada instrucción y genere las señales de control a partir de las señales de código. En la Figura Nro 12 se muestran los Componentes esenciales: Un interprete de instrucciones y un módulo de uso general para las funciones aritméticas y lógicas, ambos constituyen el CPU.

3.1 Estructura Genérica de un sistema Digital

Los datos e instrucciones deben ingresarse al sistema. Se requiere algún tipo de módulo de entrada y un módulo de salida para mostrar resultados. Se requiere almacenar temporalmente datos e instrucciones en una Memoria.

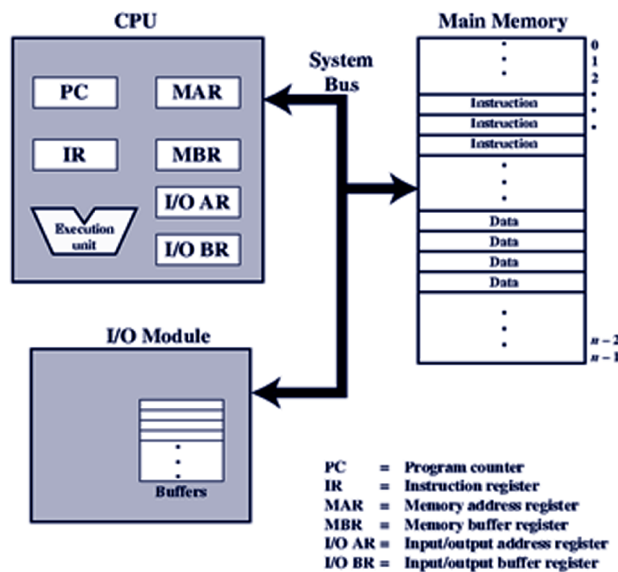


Figura Nro 14. Componentes del Computador: esquema de 2 niveles.

La función básica que realiza un computador es la ejecución de un programa que está constituido por un conjunto de instrucciones almacenadas en memoria. El procesador es el que se encarga de ejecutar las instrucciones especificadas en el programa. El procesador lee (capta) la instrucción de memoria y la ejecuta. La ejecución del programa es la repetición del proceso de captación de instrucción y ejecución de la instrucción. En el procesamiento que requiere una instrucción se denomina ciclo de instrucción.

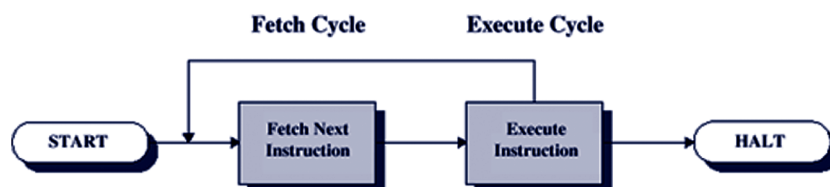


Figura Nro 15. Ciclo de captación y ejecución

3.2 Los Ciclos de Captación y Ejecución :

Al comienzo de cada ciclo la CPU capta una instrucción de memoria, esto empleando un registro llamado Contador de programa (program counter) PC. El PC posee la dirección de la próxima instrucción a ejecutar. El procesador mueve la instrucción apuntada por PC al Registro de Instrucción (RI). Se incrementa en 1 el contenido del PC.

⁹ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p. 57

¹⁰ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p. 61

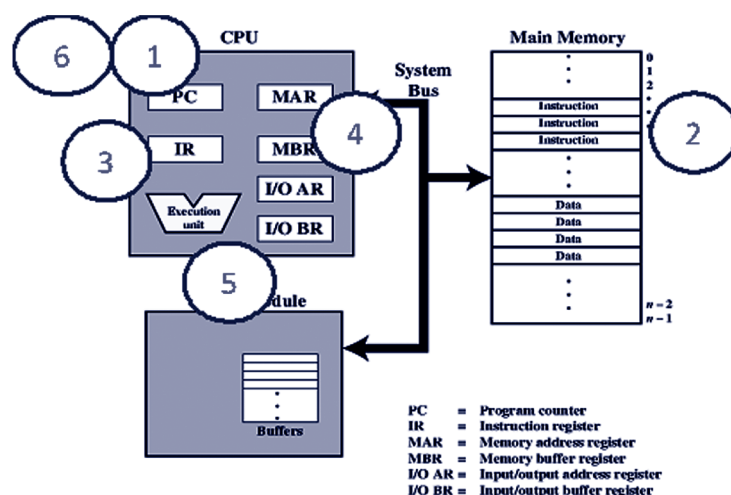


Figura Nro 16. Componentes del computador: Esquema de 2 niveles

Se interpreta la instrucción y se envían las órdenes de control. La instrucción interpretada puede ser de cuatro tipos (Proceso Memoria, Proceso- E/S, Procesamiento de datos y Control)

En este caso el procesador posee un único registro de datos llamado acumulador (AC). Tanto las instrucciones como los datos son de 16 bits. Así, es conveniente organizar la memoria utilizando posiciones de 16 bits, o palabras. El formato de instrucción indica que puede haber $2^4 = 16$ códigos de operación (codops) diferentes, y se pueden direccionar directamente hasta $2^{12} = 4096$ (4K) palabras de memoria.

El Figura 16 ilustra la ejecución de una parte de un programa, mostrando las partes relevantes de la memoria y los registros de la CPU. El fragmento de programa suma el contenido de la palabra de memoria en la dirección 940 con el contenido de la palabra de memoria en la dirección 941 y almacena el resultado en esta última posición. Se requieren tres instrucciones, que consumen tres ciclos de captación y tres de ejecución.

1. El contador de programa (PC) contiene el valor 300, la dirección de la primera instrucción (IR). Obsérvese que este proceso implicaría el uso del registro de dirección de memoria (MAR) y el registro de datos de memoria (MBR). Por simplicidad, se han ignorado estos registros intermedios.
2. Los primeros cuatro bits de IR (primer dígito hexadecimal) indican que el acumulador (AC) se va a cargar. Los restantes 12 bits (tres dígitos hexadecimales) especifican la dirección (940) que se va a cargar.
3. El registro de PC se incrementa, y se capta la siguiente instrucción (5941) desde la dirección 301.
4. El contenido anterior AC y el de la posición de memoria 941 se suman, y el resultado se almacena en AC.
5. El registro de PC se incrementa, y se capta la siguiente instrucción (294) desde la posición 302.
6. El contenido de AC se almacena en la posición 941. En este ejemplo, se necesitan tres ciclos de instrucción, cada uno con un ciclo de captación y un ciclo de

¹¹ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores* –España: Pearson Education p. 61

¹² WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores* –España: Pearson Education p. 62

ejecución, para sumar el contenido de la posición 940 y el contenido de la 941. Con un conjunto de instrucciones más complejo, se hubieran necesitado menos ciclos. Así, en algunos procesadores más antiguos se incluían instrucciones con más de una dirección. De esta forma, el ciclo de ejecución de una instrucción generaría más de una referencia a memoria. Además, en lugar de referencias a memoria, una instrucción puede especificar una operación de E/S.

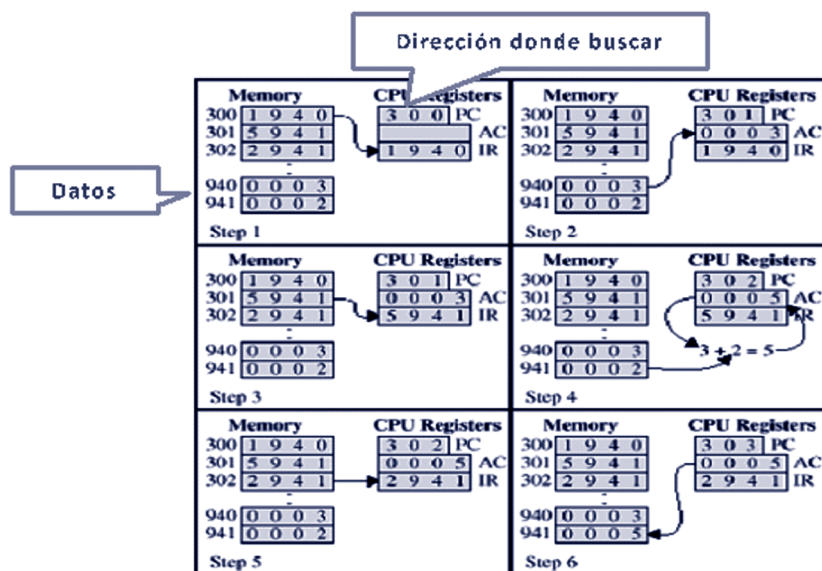


Figura Nro 17. Ejemplo de ejecución de un programa.

3.3 Ciclo de instrucción: Diagrama de estado

Así, el ciclo de ejecución de una instrucción particular puede ocasionar más de una referencia a memoria. Además, en lugar de referencias a memoria, una instrucción puede especificar una operación de E/S. Con estas consideraciones adicionales en mente, la Figura 17 proporciona una visión más detallada del ciclo de instrucción básico de la Figura 17. La figura tiene la forma de un diagrama de estados. Para un ciclo de instrucción dado, algunos estados pueden no darse y otros pueden visitarse más de una vez. Los estados se describen a continuación:

- **Cálculo de la dirección de la instrucción** (IAC, Instruction Address Calculation): Determina la dirección de la siguiente instrucción a ejecutar. Normalmente, esto implica añadir un número fijo a la dirección de la instrucción previa. Por ejemplo, si las instrucciones tienen un tamaño de 16 bits y la memoria se organiza en palabras de 16bits, se suma 1 a la dirección previa. En cambio, si la memoria se organiza en bytes (8bits) direccionables individualmente, entonces hay que sumar 2 a la dirección previa.
- **Captación de instrucción** (if, Instruction Fetch): La CPU lee la instrucción desde su posición en memoria.
- **Decodificación de la operación indicada en la instrucción** (IOD, Instruction Operation decoding): Analiza la instrucción para determinar el tipo de operación a realizar y el (los) operando(s) a utilizar.
- **Cálculo de la dirección del operando** (OAC, Operand Address Calculation): si la instrucción implica una referencia a un operando en memoria o disponible mediante E/S, determina la dirección del operando.

¹³ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p. 61

- Captación de operando (OF, Operand Fetch): capta el operando desde memoria o se lee desde el dispositivo de E/S.

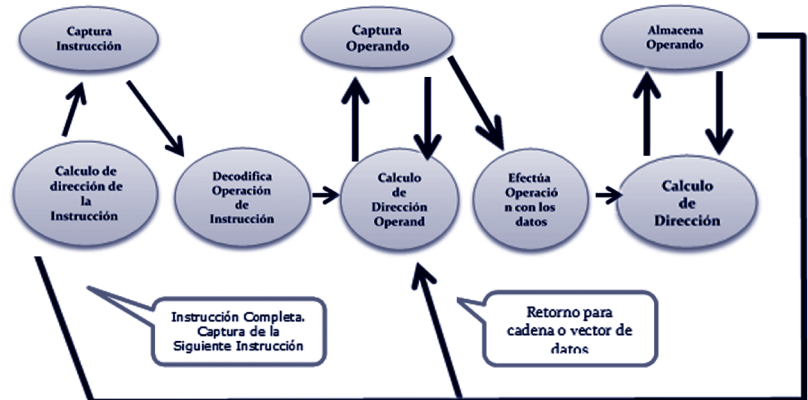


Figura Nro 18. Diagrama de estados del Ciclo de Instrucción.

3.4 Interrupciones

Los computadores poseen Mecanismos por los cuales los módulos pueden “interrumpir” el procesamiento normal de la CPU

Tipos de Interrupciones :

Programa	Generadas por alguna condición que se produce como resultado de la ejecución de una instrucción. Ejemplos, Desbordamiento aritmético (overflow), división por cero, intento de ejecutar una instrucción, máquina inexistente e intento de acceder fuera del espacio de memoria permitido por el usuario .
Temporización	Generadas por un temporizador interno al procesador. Este permite al sistema operativo realizar ciertas funciones de manera regular
Entrada/Salida	Generadas por un controlador de Entrada/Salida, para indicar la finalización sin problemas de una operación o para avisar de ciertas condiciones de error.
Fallo de Hardware	Generadas por un fallo tal como la falta de potencia de alimentación o un error de paridad en la memoria.

Figura Nro 19. Clases de Interrupciones.

En primer lugar las interrupciones proporcionan una forma de mejorar la eficiencia del procesador . Por ejemplo, la mayoría de los dispositivos externos son mucho más lentos que el procesador.

Supóngase que el procesador está transfiriendo datos a una impresora utilizando el esquema del ciclo de instrucción de la Figura 3.3. Después de cada operación de escritura, el procesador tendrá que parar y permanecer ocioso hasta que la impresora complete la escritura. La longitud de esta pausa puede ser del orden de muchos cientos o incluso miles de ciclos de instrucción que no implican acceso a memoria. Claramente, esto supone un derroche en el uso del procesador

La Figura 18 ilustra la situación del ejemplo referido en el párrafo precedente. El programa de usuario realiza una serie de llamadas de escritura (WRITE) entre mezcladas con el procesamiento. Los segmentos de código 1, 2, y 3 corresponden a secuencias de instrucciones que no ocasionan operaciones de E/S. Las llamadas de escritura (WRITE) corresponden a llamadas a un programa de E/S que es una de las utilidades del sistema operativo y que se encarga de la operación de E/S considerada. El programa de E/S está constituido por tres secciones:

¹⁴ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 64

- Una secuencia de instrucciones, rotulada con 4 en la figura, de preparación para la operación de E/S a realizar. Esto puede implicar la copia del dato que se va proporcionar en un registro intermedio (buffer) especial, y preparar los parámetros de control del dispositivo de E/S.
- La orden de E/S propiamente dicha. Si no se utilizan interrupciones, una vez que se ejecuta esta orden, el programa debe esperar que el dispositivo de E/S complete la operación solicitada. El programa esperaría comprobando repetidamente una condición que indique si se ha realizado la operación de E/S.
- Una secuencia de instrucciones, rotulada con 5 en la figura, que terminan la operación de E/S. Estas pueden incluir la activación de un indicador (flag) que señale si la operación se ha completado correctamente o con errores. Debido a que la operación de E/S puede necesitar un tiempo relativamente largo, el programa de E/S debe detenerse a esperar que concluya dicha operación; por consiguiente, el programa de usuario estará parado en las llamadas de escritura (WRITE) durante un periodo de tiempo considerable.

3.4.1 Las Interrupciones y el ciclo de instrucción:

Con el uso de interrupciones, el procesador puede dedicarse a ejecutar otras instrucciones mientras una operación de E/S está en curso. Considérese el flujo de control de la Figura 20b. Como antes, el programa de usuario llega. A un punto en el que realiza una llamada al sistema para realizar una escritura (WRITE). El programa de E/S al que se llama en este caso está constituido solo por el código de preparación y la orden de E/S propiamente dicha. Después de que estas pocas instrucciones se hayan ejecutado, el control se devuelve al programa de usuario. Mientras tanto, el dispositivo externo está ocupado aceptando el dato de la memoria del computador e imprimiéndolo. Esta operación de E/S se realiza concurrentemente con la ejecución de instrucciones del programa de usuario. Cuando el dispositivo externo pasa a estar preparado para actuar es decir, cuando está listo para aceptar más datos del procesador, el módulo de E/S de este dispositivo externo emite una señal de pendónce interrupción al procesador. El procesador responde suspendiendo la operación del programa que estaba ejecutando y salta a un programa, conocido como gestor de interrupciones.

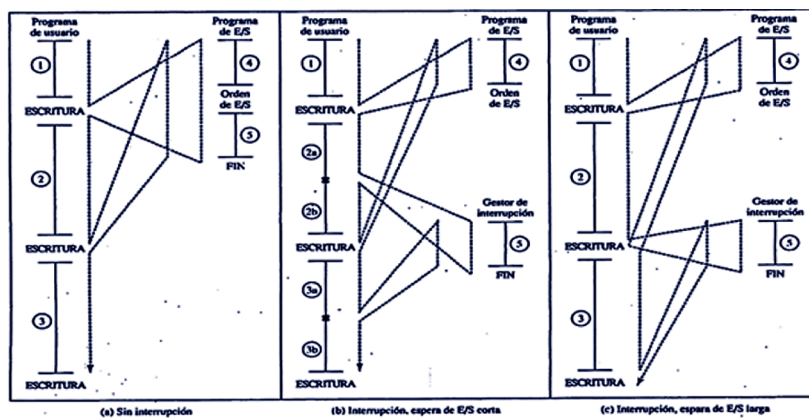


Figura Nro 20. Flujo de Control de un programa sin y con interrupcion.

Ejemplo: La mayoría de los dispositivos externos son mucho más lentos que el procesador. Supóngase que el procesador está transfiriendo datos a una impresora utilizando el esquema del ciclo de instrucción. Después de cada operación de escritura, el procesador tendrá que parar y permanecer

¹⁵ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p. 73

¹⁶ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p. 66

ocioso hasta que la impresora complete la escritura. La longitud de esta pausa puede ser del orden de muchos cientos o incluso miles de ciclos de instrucción que no implican acceso a memoria. Claramente, esto supone un derroche en el uso del procesador

El programa de usuario realiza una serie de llamadas de escritura (WRITE) entremezcladas con el procesamiento

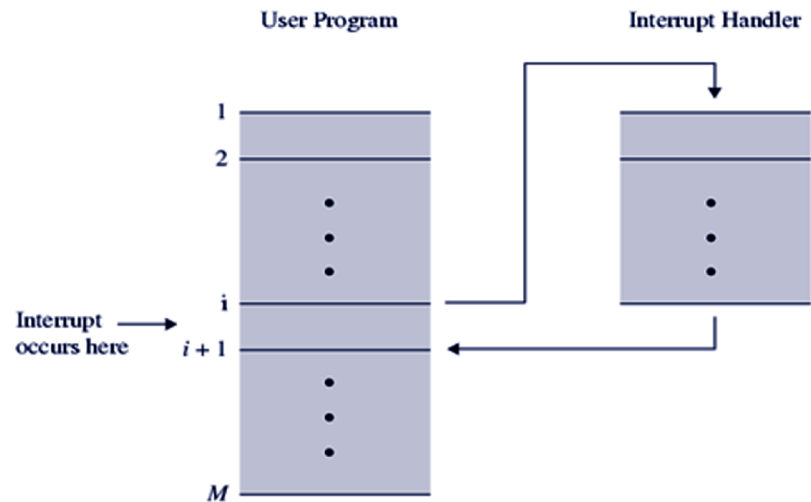


Figura Nro 21. Transferencia del Control debido a una opción.

Desde el punto de vista del programa de usuario, una interrupción es precisamente eso: una interrupción en la secuencia normal de funcionamiento. Cuando el procesamiento de la interrupción se completa, la ejecución prosigue (Figura 21). Así, el programa de usuario no tiene que incluir ningún código especial para posibilitar las interrupciones; el procesador y el sistema operativo son los responsables de detener el programa de usuario y después permitir que prosiga en el mismo punto.

Para permitir el uso de interrupciones, se añade un ciclo de interrupciones al ciclo de instrucción. Como muestra la Figura 22 En el ciclo de interrupción, el procesador comprueba si se ha generado.

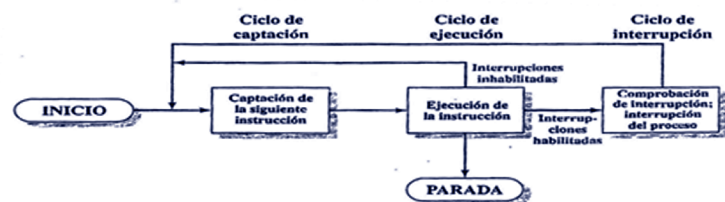


Figura Nro 22. Ciclo de Instrucción con interrupciones.

3.4.2 Interrupciones Múltiples

Hasta ahora únicamente se ha discutido la existencia de una sola interrupción. Supóngase, no obstante, que se puedan producir varias interrupciones. Por ejemplo, un programa puede estar recibiendo datos a través de una línea de comunicación e imprimiendo resultados. La impresora generará interrupciones cada vez que complete una operación de escritura. El controlador de la línea de comunicación generará una interrupción cada vez que llegue una unidad de datos. La unidad de datos puede ser un carácter o un bloque, según el protocolo de comunicación.

En Prioridades Interrupciones pueden ser interrumpidas por otras de mayor prioridad Al terminar la de mayor prioridad, se continua ejecutando la anterior

Transferencia de control con varias interrupciones anidadas.

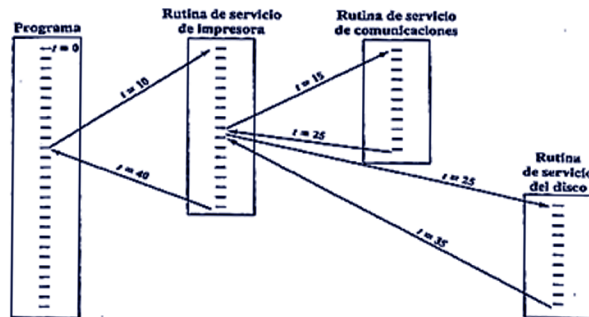


Figura Nro 23. Ejemplo de secuencia temporal de varias interrupciones.



LECTURA SELECCIONADA N° 2

“Computación con 32,64 y 128 bits“ Arturo Meza Huamán– <https://plus.google.com/118041000851955320233/posts>

¿Que Nos Ofrece Un Procesador de 64 BITS?

No se trata de una guerra por una tecnología nueva, de una innovación recién creada, ni de algo nunca antes visto. De hecho la tecnología de procesadores de 64 bits tiene más de diez años en el mercado con productos como el SUN SPARC, Digital Alpha, IBM AS/4000 y MIPS, pero estos procesadores pertenecen al segmento de los servidores y estaciones de trabajo, equipos con alto poder de cómputo y también un costo bastante elevado, razón por la cual su volumen de ventas es reducido y poca gente puede acceder a dicha tecnología.

Sin duda alguna, el segmento que tiene el mayor volumen de ventas es el de los equipos de cómputo personal, aquéllos que están en nuestra oficina, en nuestro hogar o que, incluso, podemos transportarlos con nosotros a todas partes, y es a este segmento al que ha llegado finalmente la tecnología de 64 bits.

Procesador

¿Qué significa esto de 64 bits? En principio, entendamos la diferencia con los procesadores de escritorio actuales. Hoy día tenemos procesadores de 32 bits que, incluso, corren a más de 3 Ghz, esos 32 bits podríamos decir que son como si tuviéramos una carretera de cuatro carriles donde los autos corren a 100 Km/hr; con 64 bits, esos mismos autos correrían a la misma velocidad pero ahora en una autopista de ocho carriles, con lo que podrían transitar más autos que en una carretera de sólo cuatro carriles.

Para los procesadores de 64 bits, esto significa que pueden trabajar el doble de información en el mismo ciclo de reloj (un hertz), pueden acceder a mayor capacidad de memoria y procesar archivos más grandes. Actualmente, un CPU de 32 bits puede controlar 4 GB de memoria en el caso de los procesadores de Intel y AMD, y 2 GB para los Apple (IBM); mientras que un procesador de 64 bits tiene la capacidad de controlar 16 exabytes de memoria, es decir, 16 mil millones de GB, una cantidad bastante sorprendente.

En cuanto a los cálculos matemáticos también habrá ventajas, ya que un procesador

¹⁷ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 68

¹⁸ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 69



actual de 32 bits puede representar números desde 0 hasta 4,294,967,295; con el nuevo cómputo de 64 bits, se incrementará la capacidad logrando que se puedan representar números desde 0 hasta 18,446,744,073,709,551,615. Obviamente esto significa que las computadoras podrán hacer operaciones con cantidades mayores y que los cálculos con cantidades pequeñas sean más eficientes.

Conclusión.

- Ya pasamos de 8 bits a 16 bits, y 32 bits ahora le llega la hora a los 64 bits los cuales tendrán varias décadas para que los 128 bits lleguen.
- Todos los procesadores actuales en el mercado son compatibles con 64 bits.
- En Pocos años tanto el Hardware como el Software tendrá el estándar de 64 bits.
- Un Procesador de 64 bits nos permite hacer buen uso de más de 4GB de memoria ram.
- Un Procesador de 64 bits tiene el doble de capacidad de procesamiento pero no significa que sea el doble de rápido.
- Para disfrutar las capacidades que nos brinda un procesador de 64 bits hace falta tener Software de 64 bits.
- Software de 32 bits corriendo en un procesador de 64 bits funcionara igual que si estuviese en un procesador de 32 bits.
- Hay poco Software compatible con 64 bits.
- Las aplicaciones de 16 bits no son compatibles con los procesadores de 64 bits.
- Las limitaciones de versiones de 64 bits también afectan a los sistemas operativos cada sistema operativo tiene ciertas limitaciones habría que analizar las de cada uno.



ACTIVIDAD N° 1

Esta actividad puede consultarla en su aula virtual



CONTROL DE LECTURA N° 1

Esta actividad puede consultarla en su aula virtual



GLOSARIO DE LA UNIDAD

APU: Un Accelerated processing unit en español Unidad de Procesamiento Acelerado, Unidad de Cómputo Acelerado o simplemente APU, dichos microprocesadores combinan un CPU multinúcleo, una GPU y un bus de interconexión de alta velocidad que permite transferir información a mayores velocidades, debido a que se encuentran en el mismo chip el CPU la GPU y controlador de memoria entre otros, esto es posible gracias a la miniaturización de los chips actuales procesos de fabricación de entre 40 y 32 nm.

El término unidad de procesamiento acelerado o APU fue utilizado por primera vez en un contexto público en el año 2006.

ALU: Unidad Aritmético Lógica. Es la encargada de las operaciones aritméticas y lógicas: suma, resta, multiplicación, división, mayor, menor, diferente, igual, retornando como resultado de su operación falso o verdadero etc.

BIOS: Basic Input Output System. Es un programa informático inscrito en compo-

¹⁹ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 69

²⁰ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 75

nentes electrónicos de memoria Flash (ejemplo clásico de FIRMWARE) existentes en la placa base. Este programa controla el funcionamiento de la placa base y de dichos componentes. Se encarga de realizar las funciones básicas de manejo y configuración del computador.

BIT: Unidad elemental de información que equivale a 0 o 1. Acrónimo de Binary Digit

BYTE: Conjunto de 8 bits. Se relaciona directamente con un carácter de la tabla de códigos ASCII.

CHIPSET: Es un conjunto de componentes electrónicos construidos en un circuito integrado que gestiona el flujo de datos entre el procesador, la memoria y los dispositivos periféricos. Por lo general se encuentra en la placa base de un ordenador, están diseñados para trabajar con una familia específica de procesador.

CODIFICACIÓN: Transformación de la representación de la información.

CONEXIÓN ATX: Es la conexión que se hace de la fuente de poder a la Placa. Compuesta de 20 pines o 24 pines.

CPU: Unidad Central de Procesamiento. Su nombre proviene del hecho que se encarga de procesar datos de acuerdo a las instrucciones que se le indican en memoria, finalmente presenta los resultados en alguna forma desalida.

DATOS: Símbolos para representar un número, objeto, letra etc.

DDR SDRAM : DDR permite a ciertos módulos de memoria RAM compuestos por memorias síncronas (SDRAM), disponibles en encapsulado DIMM, la capacidad de transferir simultáneamente datos por dos canales distintos en un mismo ciclo de reloj. Los módulos DDR soportan una capacidad máxima de 1 GiB (1 073 741 824 bytes).

DVI: Digital Visual Interface. Es una interfaz de vídeo diseñada para obtener la máxima calidad de visualización posible en pantallas digitales.

ESTACIONES DE TRABAJO: Computador de altas prestaciones destinado para trabajo técnico o científico. En una red de computadoras, es una computadora que facilita a los usuarios el acceso a los servidores y periféricos de la red.

HARDWARE: Elementos electrónicos y mecánicos del computador

IDE: Tecnología para la conexión de dispositivos de almacenamiento como el disco duro y la unidad de CD mediante un bus de datos de 40 hilos, relacionada con el estándar PATA(Parallel ATA).

INSTRUCCIONES: Pasos a realizar por la computadora para la ejecución de un programa.

LENGUAJE DE PROGRAMACIÓN: Instrucciones que conforman sentencias las cuales siguen unas reglas precisas.

LENGUAJES DE ALTO NIVEL: Instrucciones de entendimiento para los humanos fácilmente.

MACROCOMPUTADORAS (MAINFRAMES): Computadores de uso industrial, tienen gran procesamiento y al igual que las supercomputadoras puede tener cientos de usuarios conectados al mismo mainframe (multiusuario).

MEGAHERTZ: Medida de la frecuencia de Reloj, equivalente a 1024 KHz.





MEMORIA AUXILIAR: Dispositivos de almacenamiento permanente y externos como discos duros, USB, DVD, CD etc.

MEMORIA RAM: Random Access Memory. Memoria en la que cada posición direccionable tiene un único mecanismo de direccionamiento. El tiempo de acceso a una posición dada es independiente de la secuencia de acceso previa, es empleada por el procesador, desde la cual el procesador va tomando instrucciones y datos según la dirección que indique el Program Counter(PC).

MEMORIA ROM: Read Only Memory. Normalmente empleada como FIRMWARE, para almacenar programas que no requieran ser compilados.

NANOCOMPUTADORA: Computadora con una circuitería tan pequeña que sólo puede verse a través de un microscopio. Las nanocomputadoras pueden ser electrónicas (donde la nanolitografía se usa para crear los circuitos microscópicos), bioquímica u orgánica. Las nanocomputadoras se componen de materiales a nivel molecular y son la promesa de crear computadoras cada vez más pequeñas y rápidas, un concepto muy importante en el mundo de la computación

ORDENADOR: Terminó empleado en España para designar a las Computadoras.

PERIFÉRICOS: Son las unidades de entrada y salida.

SATA: Serial Advanced Technology Attachment. Emplea una conexión serial, por ello emplea una sola interfaz para cada dispositivo.

SISTEMA OPERATIVO: Programa principal para controlar la ejecución de todos los programas a su vez hace la labor de interfaz entre el usuario y el Hardware.

SLOTS: Ranuras donde se insertan tarjetas de expansión: Sonido, Video.

SOCKET: Elemento físico diseñado para insertar el procesador.

SOFTWARE: Son los programas.

UC: Unidad de control. Elemento de la CPU encargada de interpretar las instrucciones que se toman desde la RAM y de acuerdo a este resultado determina el flujo siguiente a efectuar.

UNIDADES DE ENTRADA: Dispositivos externos del computador que reciben la información del usuario: Mouse, Teclado, Scanner.

UNIDADES DE SALIDA: Dispositivos externos del computador que envían la información al usuario: Monitor, Parlantes, Impresora.



BIBLIOGRAFIA DE LA UNIDAD I

1. Andrew S. Tanenbaum (2000). *Organización y arquitectura de Computadoras: Un enfoque estructurado*. México: Pearson Education.
2. Domingo Mery Quiroz. (2006) *Arquitectura de computadoras*. Chile: Escuela de Ingeniería. Departamento de Ciencia de la Computación. IIC 2342. Semestre.
3. Josu Bilbao Eguía (2008). *Arquitectura de ordenadores*. España: Editorial Independiente
4. WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores*

–España: Pearson Education p.



AUTOEVALUACIÓN DE LA UNIDAD I

I. EN LAS SIGUIENTES PREGUNTAS ELIGE LA ALTERNATIVA ADECUADA.

1. El procesamiento en lotes permitió :
 - a) Acelerar el trabajo con memoria RAM
 - b) Reducir los tiempos muertos del Procesador
 - c) Eliminar los tiempos muertos del procesador
 - d) Mejorar los tiempos de respuesta
2. Un computador tiene como característica:
 - a) Ejecutar muchos procesos al mismo tiempo
 - b) Ejecutar millones de cálculos en un segundo
 - c) Controlar adecuadamente el uso de memoria y CPU
 - d) Todas las Anteriores.
3. Elementos del computador encargado de intercomunicar a los diversos componentes
 - a) ALU
 - b) Memoria RAM
 - c) Bus
 - d) Conectores
4. Digital se refiere a:
 - a) Solo se emplean dos valores posibles
 - b) Se emplean dígitos arábigos
 - c) Solo requiere ser digitado
 - d) Al empleo de muchos valores en el tiempo
5. No es una característica de Los computadores de la primera generación:
 - a) No poseían sistema operativo
 - b) Se programaban mediante comandos en ingles
 - c) Empleaban gran cantidad de energía
 - d) No se programaban mientras estaban computando.

II. Indica la verdad o falsedad de las siguientes afirmaciones

1. El reemplazo de transistores en vez de tubos al vacío mejora los tiempos de respuesta por la menor distancia entre los componentes	(V) (F)
2. En la primera generación de computadores se empleaba programación cableada	(V) (F)
3. La máquina de Von Neuman se basaba en leer sus instrucciones desde una memoria en forma incremental	(V) (F)
4. La miniaturización tiene como objetivo único hacer equipos portátiles	(V) (F)
5. La traducción hace que desde un programa (bytecode) se obtenga lenguaje maquina	(V) (F)







UNIDAD II: EL PROCESADOR



DIAGRAMA DE PRESENTACIÓN DE LA UNIDAD II



ORGANIZACIÓN DE LOS APRENDIZAJES

CONOCIMIENTOS	PROCEDIMIENTOS	ACTITUDES
Tema N° 1: La Unidad Central de Proceso 1.1 Estructura y Elementos. 1.2 La arquitectura x86 1.3 Características y Parámetros de los procesadores Lectura Seleccionada N° 1 : Xataka.com (19 -11-2012) Los próximos Intel Atom. Recuperado de: http://www.xataka.com/componentes-de-pc/bay-trail-t-los-proximos-intel-atom Tema N° 2: La Memoria Ram y los Buses 2.1 Definición de Memoria Ram 2.2 Elementos de Interconexión 2.3 Jerarquía de la Memoria 2.4 Características por las cuales se clasifican las memorias 2.5 Pueden leerse para detectar su estado. 2.6 Memoria ROM(Read Only Memory) 2.7 Memoria cache 2.8 Memoria Virtual Tema N° 3: ENTRADA /SALIDA 3.1 La placa principal 3.2 Función de la entrada/salida 3.3 Periféricos 3.4 Métodos de Programación de Entrada/salida Lectura Seleccionada N° 2 : Amazings.com (02/11/2012) “Nuevas y revolucionarias memorias” Recuperado de: http://noticiasdelaciencia.com/not/5535/nuevas_y-revolucionarias_memorias_electroquimicas_de_ordenador/	1. Identifica cada uno de los elementos de un procesador. 2. Clasifica un computador de acuerdo a las características de la CPU. Actividad N° 1 Analizar y diferenciar las características de los procesadores Intel y AMD. 3. Explica la estructura y forma de trabajo de una memoria RAM, su relación con la memoria Virtual. 4. Esquematiza el flujo de proceso en una placa. Actividad N° 2 Elabora un cuadro comparativo de Memorias RAM, para clasificarlas por sus tiempos de respuesta y frecuencia de reloj Tarea Académica N° 1 Crear un cuadro comparativo de placas de diversas marcas.	Muestra interés por entender y explicar el funcionamiento de un computador



TEMA N° 1: LA UNIDAD CENTRAL DE PROCESO

1.1 Estructura y elementos

1.1.1 Estructura del Procesador

Anteriormente en este manual vimos que un computador posee elementos básicos como son CPU, unidad Central de Proceso, Memoria Principal, Dispositivos de Entrada/Salida y elementos de Interconexión, esto influenciado por la propuesta de arquitectura de Jhon Von Neumann. Veremos en detalle la unidad Central de Proceso, como se observa en la figura 24, los elementos básicos del procesador son:

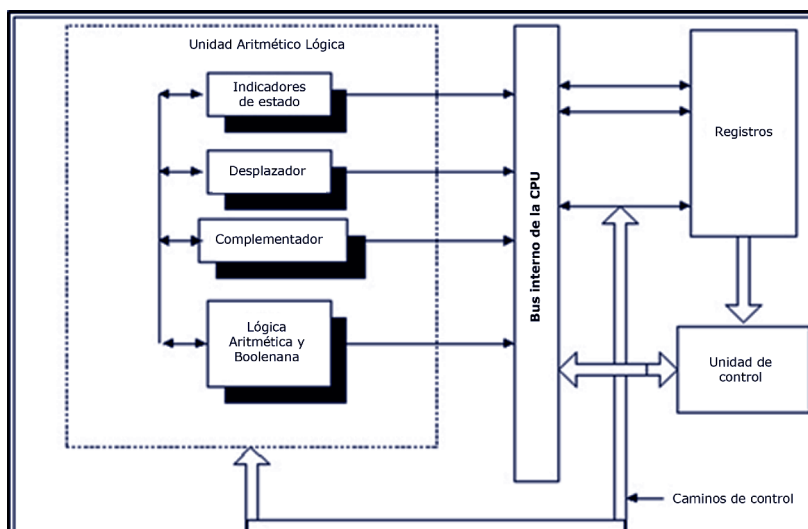


Figura Nro 24. Estructura Interna del Procesador

1.1.2 Elementos del Procesador

1.1.2.1 Unidad De Control

Es el centro nervioso del computador, ya que desde ella se controlan y gobiernan todas las operaciones. Se efectúan las operaciones aritméticas (suma, resta, y a veces producto y división) y lógicas (and, or, not, etc.). Cómo función básica tiene:

- Tomar-Capturar las instrucciones de memoria
- Decodificar o interpretar las instrucciones : Se interpretan las instrucciones que van llegando y que componen el programa
- Ejecutar las instrucciones

Para realizar su función, la unidad de control consta de los siguientes elementos:

- Contador de programa
- Registro de instrucciones
- Decodificador
- Reloj y
- Secuenciador

1.1.2.2 Unidad Aritmético Lógica (ALU):

²¹ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 437

²² WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 440

Esta unidad se encarga de realizar las operaciones elementales de tipo aritmético (sumas, restas, productos, divisiones) y de tipo lógico (AND, OR, NOT, comparaciones, desplazamientos). A través de un bus interno se comunica con la unidad de control la cual le envía los datos y le indica la operación a realizar.

La ALU está formada a su vez por los siguientes elementos:

- Circuito operacional,
- Registros de entrada
- Registro acumulador
- Registro de estado (flags).

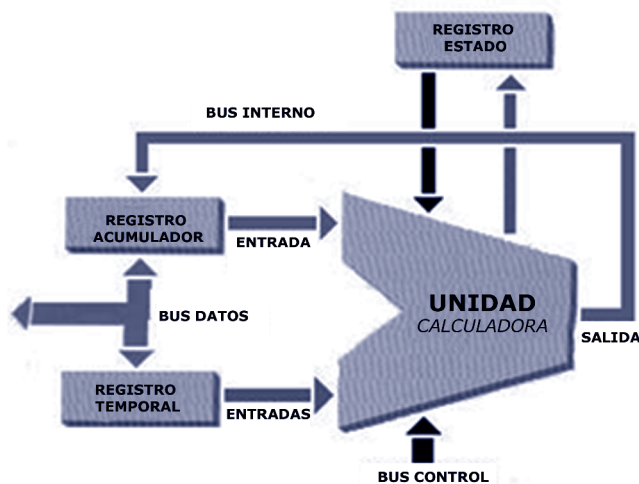


Figura Nro 25. Unidad Aritmética Lógica.

1.1.2.3 Registros:

Los registros son celdas de memoria en donde queda almacenado un dato temporalmente. Existe un registro especial llamado de indicadores, estado o flags, que refleja el estado operativo del microprocesador.

1.1.2.4 Buses:

Se poseen tanto buses internos y externos, además de una unidad de control de buse que supervisa todo el proceso de transferencias de información dentro del microprocesador y fuera de él.

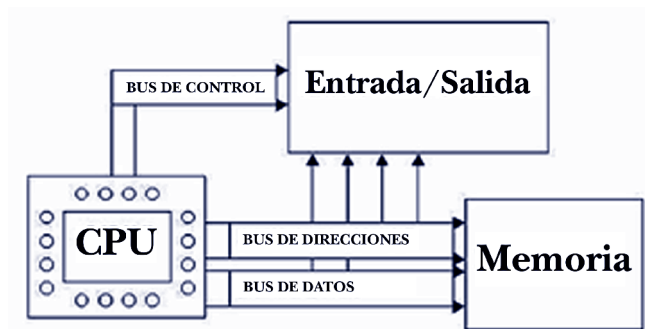


Figura Nro 26. Relación entre los buses y el Procesador

1.1.2.5 El conjunto de Instrucciones:

Se conoce como set de instrucciones al conjunto de instrucciones que es capaz de

²³ Adaptación de: <http://www.todogramacion.com.ar/images/image014.jpg>

²⁴ Hardside.com.ar(10/12/2013) Los Buses. Recuperado de: www.hardside.com.ar

entender y ejecutar un microprocesador. En función del tipo de microprocesador, concretamente si es más avanzado o no, podrá entender y ejecutar más o menos instrucciones. Una de las primeras decisiones a la hora de diseñar un procesador es decidir cuál será su juego de instrucciones. Frente a esta cuestión caben dos filosofías de diseño: CISC y RISC

1.2 LA ARQUITECTURA X86

Una primera aproximación a la arquitectura x86. Muestra que este emplea registros de propósito especial, aunque algunos de ellos se pueden también emplear para un uso general. El 8086 contiene cuatro registros de datos de 16 bits que son direccionables como registros de Bytes o como registros de 16 bits, y cuatro registros punteros e índices de 16 bits. Los registros de datos pueden utilizarse como de uso general en algunas instrucciones. Por ejemplo, una instrucción de multiplicación siempre usa el acumulador.

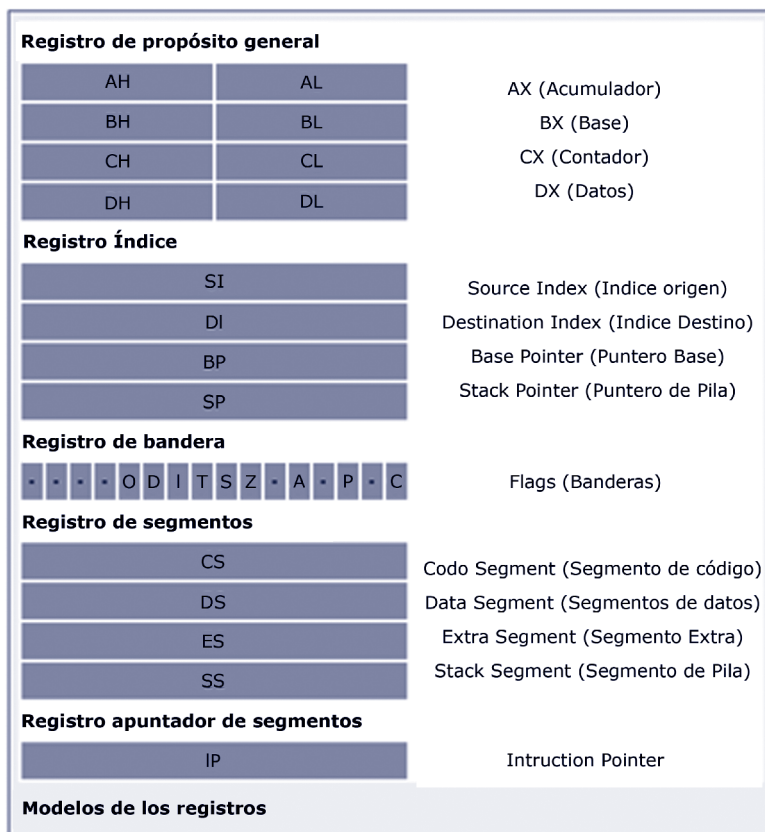


Figura Nro 27. Registros de un procesador con Arquitectura 8086

1.3 Características y parámetros de los procesadores

Dentro de los parámetros destacan: Tecnología de fabricación, número de transistores, tensión de alimentación, frecuencia de reloj, ancho de los registros de la CPU, tasa de transferencia, ancho del bus de direcciones, tamaño de las caches, velocidad de las caches, encapsulado, juegos de instrucciones avanzadas que soporta, consumo de energía y precio.

1.3.1 Generador de Reloj:

Es un dispositivo que genera un tren de pulsos en forma periódica, estos pulsos

²⁵ Chile hardware(10/12/2013). Procesador Intel 8086. Recuperado de: <http://www.chw.net/2010/09/137-procesador-intel-8086/>

digitales son estables en el tiempo hacia todos los componentes de la Tarjeta madre (Mainboard) para que estos realicen sus funciones de manera sincronizada.

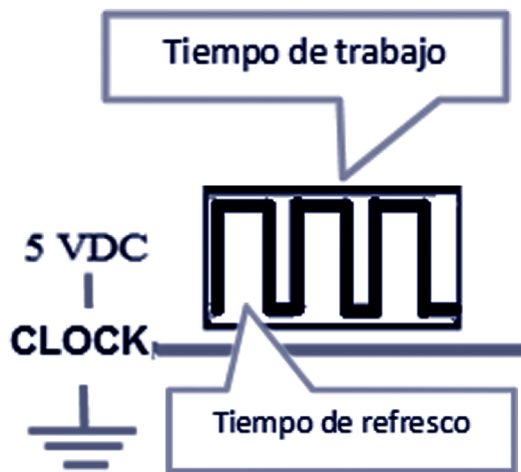


Figura Nro 28. Generador de Pulsos

1.3.2 Frecuencia:

Indica la cantidad de Ciclos que ocurren en un segundo, se mide en Hertz. Esto mismo aplicado a los procesadores: Determina el número de ciclos por segundo al que puede trabajar el núcleo de un Procesador (Ojo No es el indicativo directo y correcto para comparar dos procesadores)

UNIDADES	
Terahertz (THz)	= 10^{12} Hz
Gigahertz (GHz)	= 10^9 Hz
Megahertz (MHz)	= 10^6 Hz
Kilohertz (KHz)	= 10^3 Hz

1.3.3 Tecnología de fabricación

- Mínima distancia entre dos elementos integrados en la pastilla de silicio
- Cuanto menor es la distancia, más pequeños son los transistores y por lo tanto son más rápidos y disipan menos energía
- A medida que disminuye la distancia, aparecen problemas tecnológicos. Además, el coste de desarrollo se incrementa exponencialmente

La tecnología de Fabricación Expresa la distancia entre los transistores en el procesador, se da en Micras y en nanómetros

Un nanómetro es : 10^{-9}

Ejemplos: 1 cabello humano 100000 nm
1 germen 1000 nm

Visita: learn.genetics.utah.edu/content/begin/cells/scale

Los núcleos de las CPUs están formados por millones de transistores. Estos transistores están basados en semiconductores de metal-óxido (MOS-Metal Oxido Silicio). La tecnología MOS utiliza un hilo de óxido metálico depositado sobre un sustrato de silicio. El silicio puro no es conductor, por lo que se le dopa con impurezas dentro del sustrato formando caminos conductores. Cuando se aplica una tensión alta en la capa metálica la corriente fluye entre las capas de impurezas. Este circuito puede ser considerado como un semiconductor, ya que el circuito MOS puede comportarse como conductor o aislante con un simple cambio de campo eléctrico

La tecnología CMOS (Complementary Metal-Oxide Semi-conductor) es una actualización del estándar MOS, y es utilizada por la mayoría de dispositivos de una computadora. La tecnología CMOS funciona de forma similar al MOS, solo que con la tensión inversa (de ahí lo de Complementary). Para cerrar el circuito en un dispositivo CMOS, debe aplicarse un nivel bajo de tensión. Utilizando circuitos CMOS y MOS, una CPU puede trabajar a mayores velocidades ya que operan con



tensiones alternativas simultáneamente

La disminución en el tamaño de las puertas tiene las siguientes implicaciones:

- Aumento de la densidad de integración (más transistores en el mismo espacio)
- Menor tensión necesaria para la conmutación de dichas puertas
- Posibilidad de aumentar la frecuencia de conmutación de las puertas (Aumento de frecuencia de la señal de reloj del micro)

1.3.4 Número de transistores de la CPU

- Los transistores dentro de la CPU son como las líneas de código de un programa. Cuantos más transistores hay disponibles, más funcionalidad se puede implementar
- Está íntimamente relacionado con la tecnología de fabricación.
- Cuanto menor es la distancia de integración, más pequeños son los transistores y por lo tanto caben más en la misma superficie
- Determina la cantidad de funciones que puede incorporar un procesador: Coprocesador, FPU, Cache, Predictor, Controladora de memoria.

Por ejemplo, el Pentium 4 con núcleo Northwood emplea tecnología de fabricación de 130 nanómetros y contiene 55 millones de transistores, mientras que el Core i 7, emplea tecnología de fabricación de 45 nm o 32 nm y posee 731 millones de transistores

1.3.5 Tensión de alimentación

- Muy relacionada con el tamaño de los transistores y por lo tanto con la tecnología de fabricación
- Cuanto más pequeño son los transistores, menores son las tensiones necesarias para que desempeñen su trabajo
- En general, tensiones menores implican menor consumo de energía por transistor
- Esto ha permitido que en los últimos años el consumo de energía se haya mantenido a pesar de aumentar el número de transistores

Por ejemplo, el Pentium 4 utiliza una tensión de alimentación de 1,52 voltios. El Core i 7 emplea 1.165 Voltios.

1.3.6 Frecuencia de reloj

- Suele expresarse en megahercios (MHz) o gigahercios (GHz)
- Indica el número de ciclos por segundo del reloj de la CPU
- Muy relacionada con el rendimiento para un mismo núcleo o microarquitectura. Mayor frecuencia => mayor rendimiento, pero no es el parámetro definitivo para determinar la performance final de un computador.

Ej: un Pentium 4 con núcleo Northwood tiene 3 GHz de frecuencia de reloj interna, mientras que un Core i7 puede ir desde 2.2 GHz hasta 4.1 con Overclocking.

1.3.7 Ancho de los registros

- Indica si una CPU es de 8, 16, 32 o 64 bits
- Se refiere al tamaño de los registros de enteros en bits. Los registros de coma flotante de una CPU de 32 bits son habitualmente de tamaño mayor de 64 bits
- Coincide con el tamaño de los buses internos de la CPU
- La mayor parte de las CPUs de PC actuales son de 32 bits, compatibles IA-32 (x86) y CPUs de 64 bits para el mundo PC (hace más de 10 años que habían aparecido en servidores y estaciones de trabajo avanzadas).

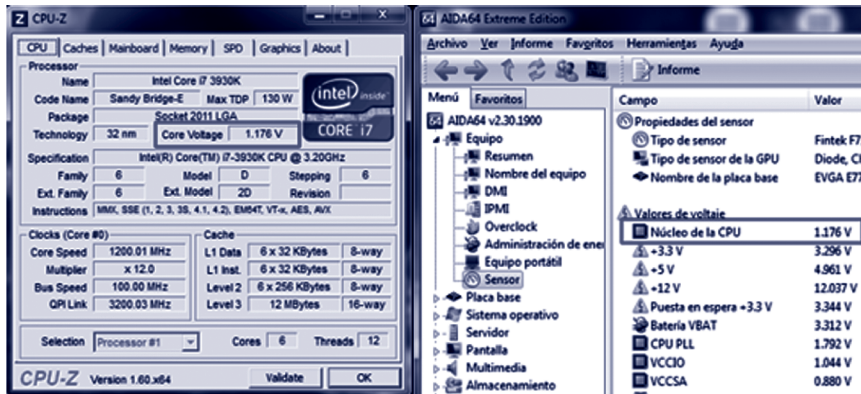


Figura Nro 29. Información del Procesador obtenida por software

1.3.8 Tasa de transferencia

- Se trata de la velocidad con la que la CPU se comunica con su entorno
- La mayor parte de las CPUs actuales se comunican a través del Front Side Bus (FSB) en el caso de los modelos a partir de Core I, mediante QuickPath Interconnect QPI, en el caso de los FSB se logran tasa de transferencias de hasta 3200MB/s y en el caso de los QPI de 6.4 G transferencias.
- Otras alternativas al FSB es el empleo de conexiones punto a punto. Este es el caso de la tecnología HyperTransport auspiciada por AMD. Esta tecnología se emplea en los modernos Athlon XP 64 y Opteron

1.3.9 Ancho del bus de direcciones

- Se trata del número de bits que componen el bus de direcciones de la CPU
- Indica la máxima cantidad de memoria física que se puede usar
- Las CPUs x86 de 32 bits actuales tienen 36 líneas de direcciones => un máximo de $2^{36} = 64$ Gbytes de memoria física
- Los Athlon XP y Opteron son de 64 bits y tienen 40 líneas de direcciones => un máximo de $2^{40} = 1$ Tbyte (terabyte) de memoria física

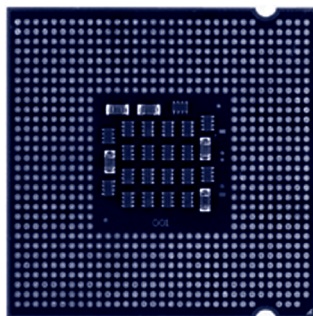
1.3.10 Tamaño de las caches

- Las caches mejoran los tiempos de acceso a memoria. En general, una mayor capacidad de cache es beneficiosa para el rendimiento
- La mayor parte de las CPUs de PC actuales incorporan dos niveles de caché dentro del mismo chip de la CPU: una cache L1 unida a la CPU y una cache L2 entre la L1 y la memoria principal
- La cache L1 está dividida en una cache de código y una cache de datos para permitir accesos simultáneos a código y datos.
- La cache L2 es unificada para código y datos.

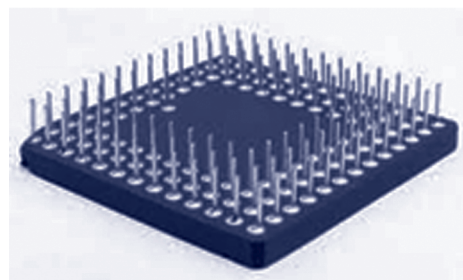
1.3.11 Encapsulado

La CPU no se suelda a la placa base sino que está unida a la placa base a través de un zócalo. ¿Por qué? El empleo de zócalos (sockets) de conexión permite conectar diferentes CPUs a la placa base, o cambiar fácilmente una CPU estropeada

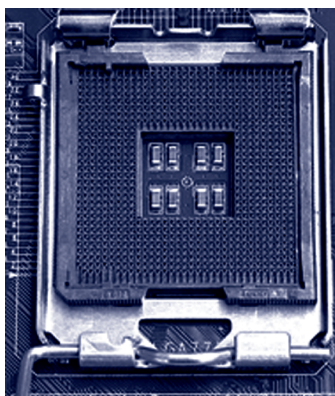
- El zócalo más empleado anteriormente fue el Pin Grid Array (PGA) o una variedad de este. Se trata de un zócalo de tipo Zero Insertion Force (ZIF)
- Por ejemplo, los Pentium 4 suelen usar el socket 478, que contiene 478 pines. El Athlon XP usa el socket A, el cual es un zócalo PGA de 462 pines.
- El zócalo más empleado en la actualidad es el LGA, Los procesadores de Intel de gama alta pasan a usar un socket LGA 775 mientras que los de AMD el socket PGA 939



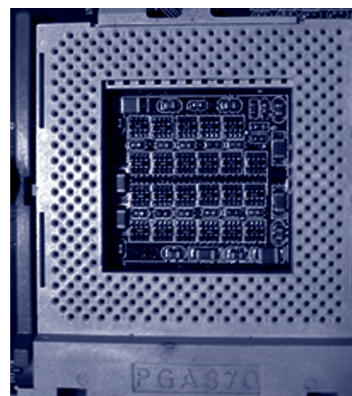
a) Empaquetado LGA



b) Empaquetado PGA



a) Socket LGA



b) Socket PGA

Figura Nro 30. Procesadores y Sockets**1.3.12 Juegos de instrucciones avanzadas:**

- Se añaden a la arquitectura IA-32 básica para mejorar el rendimiento en ciertas tareas
- Salvo las extensiones de la arquitectura AMD64, se trata de extensiones Single Instruction Multiple Data (SIMD)
- Tienen su principal aplicación en aplicaciones de procesamiento de imágenes y sonido
- Las extensiones SIMD permiten que una instrucción trabaje con vectores en lugar de escalares. Ej: una única instrucción puede sumar las componentes (Red, Green, Blue) de dos pixels.
- La primera de estas extensiones fueron las MMX, a las que siguieron SSE por parte de Intel y 3DNow! por parte de AMD.
- Un Core i 7 posee como extensiones: x86, x86-64, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2

²⁶ Diego Svalbard (Mayo 2013). Tipo de sockets para procesadores. Recuperado de: <http://coreducacion-sistemasii.blogspot.com/2012/05/socket-lga.html>

²⁷ Pin Grid Array PGA. (12/03/2012). En Wikipedia. http://es.wikipedia.org/wiki/Pin_grid_array

²⁸ LGA 775 (19/11/2013). En Wikipedia. http://es.wikipedia.org/wiki/LGA_775

²⁹ Socket PGA370.JPG (29/08/2009). En Wikipedia. http://commons.wikimedia.org/wiki/File:Socket_PGA370.JPG

1.3.13 DISIPACIÓN TÉRMICA

- Uno de los parámetros que más problemas ha ocasionado en el desarrollo de los micros ha sido el calentamiento que sufren. Fue famoso el caso de los primeros Pentium 60/66 MHz que tuvieron serios problemas de esta índole
- El aumento de la escala de integración (mas transistores en la misma superficie) en los micros hace que aumente su consumo y dificulta la disipación del calor que provocan
- Los problemas de calor son proporcionales al voltaje y a la frecuencia de trabajo y pueden provocar un comportamiento errático del sistema e incluso la destrucción del micro.

1.3.14 Resistencia térmica (Termorresistividad)

1.3.14.1 Disipador :

Va pegado a la superficie térmica del micro y su misión es disminuir la resistencia térmica del conjunto, aumentando el área de contacto del micro con el aire

- La resistencia térmica del disipador viene determinada por su superficie, el material con el que está construido, su forma, y el flujo de aire que lo atraviese

Interfaz térmica: Por contacto directo Se presiona el disipador contra la superficie térmica mediante unos clips que sujetan ambos elementos

- Ambas superficies deben ser lisas, del mismo tamaño, y la conductividad térmica del disipador igual o mejor que la de la superficie térmica

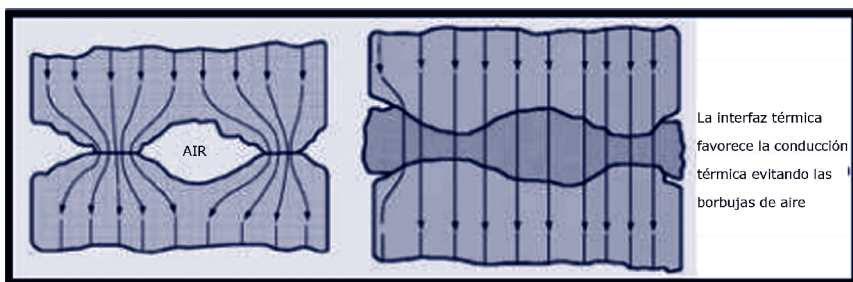


Figura Nro 31. Flujo del aire en un Disipador

Entre la superficie térmica y el disipador se puede aplicar alguno de los productos especiales existentes que poseen una conductividad mucho mayor que el aire

- Los productos “secos” como pegatinas y almohadillas son más fáciles de usar pero tienen un peor rendimiento térmico. Pueden aparecer pequeñas bolsas de aire durante su instalación.
- Los productos “húmedos” como pastas, geles, silicona termoconductora y epoxy poseen menores resistencias térmicas. Estos productos llenan las burbujas microscópicas que se forman y pueden alcanzar resistencias menores de 0,1 °C/W.

³⁰ Ignacio Moreno Velasco(2012). Mantenimiento de equipos informáticos. España: Universidad de Burgos p. 9



LECTURA SELECCIONADA N° 1

Los próximos Intel Atom (Kote @kotecinho)- 19 de noviembre de 2012 | 03:46 CET



De la misma forma que esperamos que los procesadores ARM se vayan poniendo las pilas para ser más capaces y potentes, también queremos que Intel haga avances importantes en su familia de procesadores Atom, destinados al bajo consumo y dispositivos móviles. Por primera vez tenemos constancia de la nueva generación de productos Atom, que será conocida como Intel Bay Trail-T, según aparece en una hoja de ruta filtrada en el día de hoy. En el documento podemos ver que Bay Trail-T estará en el mercado en 2014, y también nos informan de que la tecnología de fabricación se irá a los 22nm, todo ello dentro de un SoC bautizado como Valley View-T.

En los mentideros tecnológicos se cuenta que Intel quiere potenciar esta plataforma al máximo, y desde el primer día, por lo que se cuenta que ya está en contacto con fabricantes para que empiecen a trastear el hardware, con bastante más antelación de lo habitual.

Feature	Clover Trail	Bay Trail-T	Key Improvements
SOC	Cloverview SoC	Valleyview-T SOC	
CPU Core Process	32-nm	22-nm	Half the power at same performance
# cores / # threads	Dual Core / 4 threads	Quad Core / 4 Threads	50-60% performance improvement
HFM/LFM	1.5 GHz/600 MHz	2.1 GHz	
DRAM Type	CoPOP LPDDR2-800	BGA LPDDR3-1067	TM LPDDR3, BGA configuration, Higher memory capacity
# of channels	2 channel, 4 ranks	2 channel, 4 ranks	
Data width	32bit	64 bits	
Memory BW	6.4GB/s→8.5GB/s	8.5GB/s→17GB/s (LPDDR3)	
Graphics Core	SGX545 400MHz, DX*9.3	Gen7 derivative Intel Graphics, DX 11	Up to 3X+ performance improvement
Video Decoder	VXD391	Gen7 Media decode and VXD392	
Video Encoder	VXE285 (up to1080p30)	Gen7 1080p30 (H.264, H.263 @1080p30)	
Display	DSI-LVDS Bridge (1920x1080), HDMI rev1.3a	MIPI-DSI (2560x1600), eDP (2560x1600), HDMI rev1.4	

Cabe la posibilidad de que el nuevo hardware sea presentado en el CES 2013, para demostrar a la gente las mejoras importantes con respecto a la actual generación, Intel Clover Trail (32nm). Se informa sobre una mejora en rendimiento entre un 50 y un 60% entre generaciones, o bien, una reducción del consumo en un 50%

con la misma potencia. En lo que respecta a la competencia ARM, ya sabéis que tanto el nuevo Chromebook como la tablet Nexus 10, se atreven con un procesador de doble núcleo ARM Cortex-A15 con tecnología de fabricación de 28nm.

Clover Trail		Bay Trail-T
Video Battery Life	~9+hr	>11 hr
Standby Battery Life	~30 days	~20 days
Z height	~8.5mm	~8-9 mm or less
Skin Temp (Over Ambient) *	~+15 °C Plastic Skins ~+12-13 °C Metal Skins	~+15 °C Plastic Skins ~+12-13 °C Metal Skins
OS Coverage	32-bit Windows* 8	32-bit Windows* 8
Threads/Cores	4/2	4/4
Process	32nm	22nm
Platform Benefits	TTM Win 8	New Architecture, Fully Out of Order, Improved Active Battery Life, Greater CPU & Graphics Performance
Graphic Core	SGX 545	Intel Gen 7
Windows GfX Capability	DX*9.3	DX*11
Platform Features	LTE, WIDI, PC Sync, Intel® Insider	Clover Trail + Enhanced Security, Compute Continuum

La nueva plataforma aceptará procesadores de cuatro núcleos hasta 2.1GHz, soporte DirectX 11, o resoluciones de 2560 × 1600 píxeles. Hablando de otros límites en el hardware, también se adelanta la posibilidad de incorporar cámaras 3D, con capacidad de grabar vídeo 1080p a 60 imágenes por segundo.



ACTIVIDAD N° 1

Esta actividad puede consultarla en su aula virtual



TEMA N° 2: LA MEMORIA RAM Y LOS BUSES

2.1 Definición de Memoria Ram

La memoria es el espacio de trabajo del procesador de la PC. Es un área de almacenamiento provisional en donde deben residir los programas y los datos utilizados por el procesador. Es un espacio temporal. Los datos y los programas permanecen en ella en tanto la computadora tenga suministro eléctrico o no sea reiniciada, antes de ser apagada o reiniciada todo dato debe ser guardado en un dispositivo de almacenamiento permanente.

Se le conoce como RAM (Memoria de Acceso Aleatorio) por lo que es posible acceder a cualquier ubicación de ella en forma aleatoria y rápidamente, tal término se refiere a la cualidad de presentar iguales tiempos de acceso a cualquiera de sus posiciones (ya sea para lectura o para escritura). Esta característica también es conocida como “acceso directo”, esta designación es un tanto engañosa (Los discos rígidos y la memoria ROM también tienen acceso aleatorio).

De acuerdo a la arquitectura de Jhon Von Neumann, en la RAM se almacena temporalmente la información, datos y programas que la Unidad de Procesamiento (CPU) lee, procesa y ejecuta. La memoria RAM también es conocida como la Memoria principal, Central o de Trabajo” de un computador; a diferencia de las llamadas memorias auxiliares y de almacenamiento masivo (como discos duros o cintas magnéticas).

Buses: Son un conjunto de Hilos que permiten la comunicación con los distintos componentes de la computadora.

Es importante que identifiques la necesidad del Procesador de **acceder a las instrucciones** de los programas y datos que deba procesar, los cuales se encuentran **en memoria RAM**, justamente para poder realizar este proceso deben emplear una conexión a memoria la cual se realiza mediante los buses, las imágenes a continuación muestran el detalle de este trabajo:

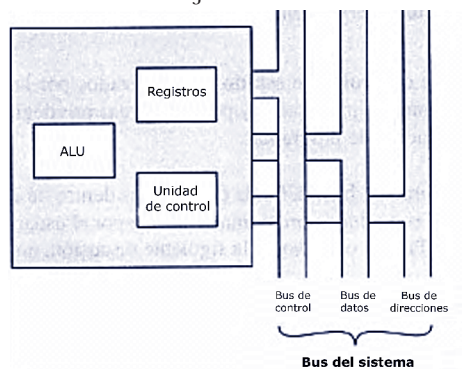


Figura Nro 32. El procesador y el bus del sistema

2.2 Elementos de Interconexión

Un bus es un camino de comunicación ente dos o más dispositivos. UNA característica clave de un bus es que se trata de un medio de transmisión compartido. AL bus se conectan varios dispositivos, y cualquier señal transmitida por uno de esos dispositivos está disponible para que los otros dispositivos conectados al bus puedan acceder a ella. SI dos dispositivos transmiten durante el mismo periodo de tiempo, sus señales pueden solaparse y distorsionarse. Consiguientemente, solo un disposi-

³¹ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education p. 440*

³² WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education p. 440*

tivo puede transmitir con éxito en un momento dado.

Usualmente, un bus está constituido por varios caminos de comunicación, o líneas. Cada línea es capaz de transmitir señales binarias representadas por 1 y por 0. En un intervalo de tiempo, se puede transmitir una secuencia de dígitos binarios a través de una única línea. Se pueden utilizar varias líneas del bus para transmitir dígitos binarios simultáneamente (en paralelo). Por ejemplo un dato de 8 bits se puede transmitir mediante ocho líneas del bus.

Las computadoras poseen diferentes tipos de buses que proporcionan comunicación entre sus componentes a distintos niveles de la jerarquía del sistema. El bus que conecta los componentes principales del computador (procesador, memoria, E/S) se denomina bus del sistema (system bus). Las estructuras de interconexión más comunes dentro de un computador están basadas en el uso de uno o más buses del sistema.

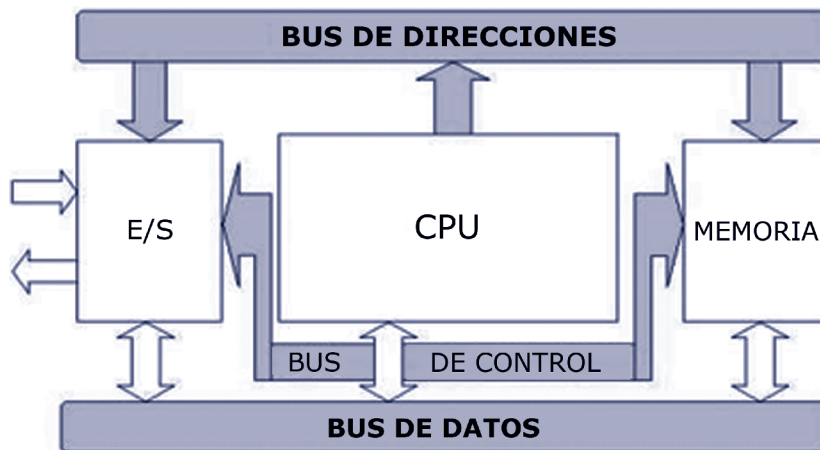


Figura Nro 33. Esquema de interconexión mediante un Bus

2.1.1 Bus de direcciones:

Transporta las direcciones de origen y/o de destino de la información transmitida sobre el bus de datos. Este bus especifica las diferentes ubicaciones de memoria generadas por el microprocesador. Es decir considerando la dirección indicada por el procesador este Bus permite apuntar (direccionar) a la ubicación requerida. Por ejemplo si el procesador requiere leer una palabra (8,16 o 32 bits), sitúa la dirección de la palabra deseada en las líneas de direcciones.

La anchura del bus de direcciones (número de bits que se empleen), determina la cantidad de direcciones de memoria que puede direccionar el CPU, es decir la capacidad máxima de memoria del sistema. Las líneas de direcciones se usan también para direccionar los puertos de E/S. Usualmente los bits de orden más alto se utilizan para seleccionar una posición de memoria o un puerto de E/S dentro de un módulo. Por ejemplo en un bus de 8 bits, la dirección 01111111 e inferiores harían referencia a posiciones dentro de un módulo de memoria (el módulo 0) con 128 palabras de memoria, y las direcciones 10000000 y superiores designarían dispositivos conectados a un módulo de E/S (módulo 1).

2.1.2 Bus de datos:

Transporta los datos a emplear desde o hacia el dispositivo direccionado. El bus de datos puede estar construido de 16, 32 o más líneas, cuyo número se conoce como anchura del bus de datos, la cual determina la cantidad de información transmitida en cada ciclo de reloj. Este es un factor clave a la hora de determinar la perfor-

³³ Angel Blas Alonso(2/12/2010). Bus. Recuperado de : <http://angeldeblasalonso.blogspot.com/2010/12/bus.html>



mance del sistema. Por ejemplo, si el bus de datos tiene una anchura de 8 bits, y las instrucciones son de 16 bits, entonces el procesador debe acceder al módulo de memoria 2 veces por cada ciclo de instrucción.

2.2.3 Bus de Control

Transporta las señales de reloj y controla la ejecución de las instrucciones de programa. El bus de control determina cuándo se debe realizar una operación de escritura o una lectura de memoria. Puesto que las líneas de datos y de direcciones son compartidas por todos los componentes, debe existir alguna forma de controlar su uso. Las señales de control transmiten tanto ordenes como información de temporización entre los módulos del sistema. Las señales de temporización indican la validez de los datos y las direcciones. Las señales de órdenes especifican las operaciones a realizar.

2.2.4 Función de los Buses

Cada uno de los buses tiene una función específica, suponiendo el inicio de la ejecución de un computador, este tiene en su PC un valor por defecto que es hacia donde apunta el bus de direcciones; se considera como una dirección de inicio igual a Cero, en ese momento el bus de control indica el tipo de operación; en este caso LECTURA, el dato lecionado se trae a través del bus de datos, hacia un registro del Procesador para que este lo interprete.

Ahora tenemos que el ciclo de captación y ejecución ahora está más detallado con respecto a la función que efectúan los demás elementos del Computador, a pesar de esto, no hemos mencionado toda una lista de detalles, esto con el fin de entenderlo más fácilmente.

2.2.5 Jerarquía de los buses

Independientemente del tipo de bus y de su cometido, todos los buses van a tener algunos problemas comunes, así si se conecta un gran número de dispositivos al bus, las prestaciones pueden disminuir. Hay dos causas principales:

En general, a más dispositivos conectados al bus, mayor es el retardo de propagación. Este retardo determina el tiempo que necesitan los dispositivos para coordinarse en el uso del bus. Si el control del bus pasa frecuentemente de un dispositivo a otro, los retardos de propagación pueden afectar sensiblemente a las prestaciones.

1. El bus puede convertirse en un cuello de botella a medida que las peticiones de transferencia acumuladas se aproximan a la capacidad del bus. Este problema se puede resolver en alguna medida incrementando la velocidad a la que el bus puede transferir los datos y utilizando buses más anchos (por ejemplo incrementando el bus de datos de 32 a 64 bits). Sin embargo, puesto que la velocidad de transferencia que necesitan los dispositivos conectados al bus (por ejemplo, controladores de gráficos y vídeo, interfaces de red) está incrementándose rápidamente, es un hecho que el bus único está destinado a dejar de utilizarse.

Por consiguiente, la mayoría de los computadores utilizan varios buses, normalmente organizados jerárquicamente. Una estructura típica se muestra en la figura 0. Hay un bus local que conecta el procesador a una memoria caché y al que pueden conectarse también uno o más dispositivos locales. El controlador de memoria caché conecta la caché no sólo al bus local sino también al bus de sistema, donde se conectan todos los módulos de memoria principal. El uso de una caché alivia la exigencia de soportar los accesos frecuentes del procesador a memoria principal.

De hecho, la memoria principal puede pasar del bus local al bus de sistema. De esta forma, las transferencias de E/S con la memoria principal a través del bus de sistema no interfieren la actividad del procesador.

Es posible conectar controladores de E/S directamente al bus de sistema. Una solución más eficiente consiste en utilizar uno o más buses de expansión. La interfaz del bus de expansión regula las transferencias de datos entre el bus de sistema y los controladores conectados al bus de expansión. Esta disposición permite conectar al sistema una amplia variedad de dispositivos de E/S y al mismo tiempo aislar el tráfico de información entre la memoria y el procesador del tráfico correspondiente a las E/S.

La figura 34 muestra algunos ejemplos típicos de dispositivos de E/S que pueden estar conectados al bus de expansión. Las conexiones a red incluyen conexiones a redes de área local (LANs, de “Local Area Networks”) tales como una red Ethernet de 10 Mbps y conexiones a redes de área amplia tales como la red de paquetes conmutados. La interfaz SCSI (de “Small Computer System Interface”) es en sí un tipo de bus utilizado para conectar controladores de disco y otros periféricos. El puerto serie puede utilizarse para conectar una impresora o un escáner.

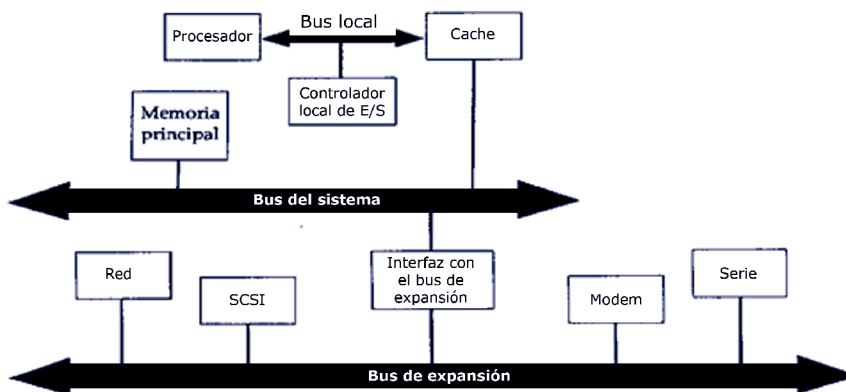


Figura Nro 34. Arquitectura de bus tradicional

Esta arquitectura de buses tradicional es razonablemente eficiente, pero muestra su debilidad a medida que los dispositivos de E/S ofrecen prestaciones cada vez mayores. La respuesta común a esta situación, por parte de la industria, ha sido proponer un bus de alta velocidad que está estrechamente integrado con el resto del sistema, y requiere sólo un adaptador (bridge) entre el bus del procesador y el bus de alta velocidad. En algunas ocasiones, esta disposición es conocida como arquitectura de entreplanta (“mezzanine architecture”).

La figura 35 muestra un ejemplo típico de esta aproximación. De nuevo, hay un bus local que conecta el procesador a un controlador de caché, que a su vez está conectado al bus de sistema que soporta a la memoria principal. El controlador de caché está integrado junto con el adaptador, o dispositivo de acoplo, que permite la conexión al bus de alta velocidad. Este bus permite la conexión de LANs de alta velocidad, tales como interfaces de fibra óptica (FDDI, de “Fiber Distributed Data Interface”) de 100 Mbps, controladores de estaciones de trabajo específicas para aplicaciones gráficas y de vídeo, y también controladores de interfaz para buses de periféricos tales como SCSI y P1394. Este último es un bus de alta velocidad diseñado específicamente para conectar dispositivos de E/S de alta capacidad. Los dispositivos de velocidad menor pueden conectarse al bus de expansión, que utiliza una interfaz para adaptar el tráfico entre el bus de expansión y el bus de alta velocidad.

La ventaja de esta organización es que el bus de alta velocidad acerca al procesador

³⁴ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education p. 81*

los dispositivos que exigen prestaciones elevadas y al mismo tiempo es independiente del procesador. Así, se pueden tolerar las diferencias de velocidad entre el procesador y el bus de altas prestaciones y las variaciones en la definición de las líneas de los buses. Los cambios en la arquitectura del procesador no afectan al bus de velocidad, y viceversa.

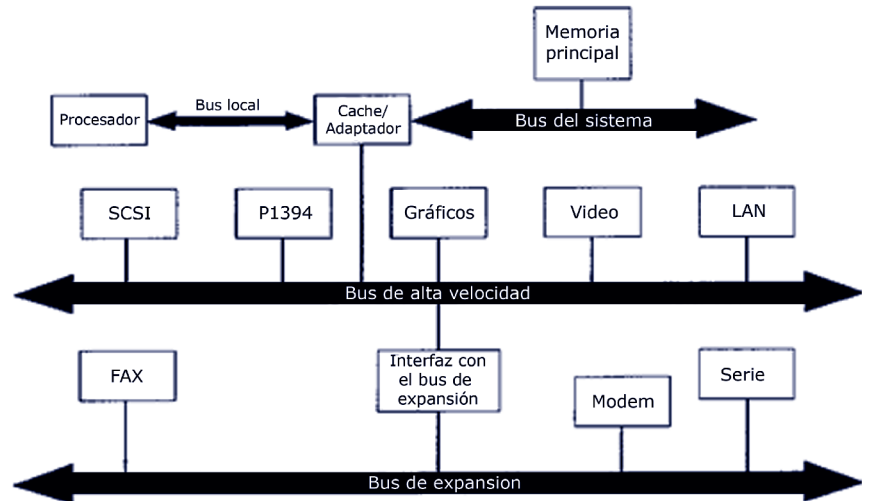


Figura Nro 35. Arquitectura de bus de altas prestaciones

2.2.6 El bus PCI (de “Peripheral Component Interconnect”

Interconexión de Componente Periférico) es un bus de ancho de banda elevado, independiente del procesador que se puede utilizar como bus de periféricos o bus para una arquitectura de entreplanta. Comparado con otras especificaciones comunes de bus, el PCI proporciona mejores prestaciones para los subsistemas de E/S de alta velocidad (por ejemplo, los adaptadores de pantalla gráfica, los controladores de interfaz de red, los controladores de disco, etc.). El estándar permite el uso de hasta 64 líneas de datos a 33 MHz, para una velocidad de transferencia de 264 Mbytes/s, o 2.112 Gbps. Pero no es precisamente su elevada velocidad la que hace atractivo al PCI. El PCI ha sido diseñado específicamente para ajustarse, económicamente, a los requisitos de E/S de los sistemas actuales; se implementa con muy pocos circuitos integrados y permite que otros buses se conecten al bus PCI.

Intel empezó a trabajar en el PCI en 1990 pensando en sus sistemas basados en el Pentium. Muy pronto Intel cedió sus patentes al dominio público y promovió la creación de una asociación industrial, la PCI SIG, para continuar el desarrollo y mantener la compatibilidad de las especificaciones del PCI. El resultado ha sido que el PCI ha sido ampliamente adoptado y se está incrementando su uso en los computadores personales, estaciones de trabajo, y servidores de sistema. La versión PCI 2.0, apareció en 1993. Puesto que las especificaciones son de dominio público y están soportadas por una amplia banda de la industria de procesadores y periféricos, los productos PCI fabricados por compañías diferentes son compatibles.

El PCI está diseñado para permitir una cierta variedad de configuraciones basadas en microprocesadores, incluyendo sistemas tanto de uno como de varios procesadores. Por consiguiente, proporciona un conjunto de funciones de uso general. Utiliza temporización síncrona y un esquema de arbitraje centralizado. La figura 37 muestra la forma usual de utilizar el bus PCI en un sistema uniprocador. Un dispositivo que integra el controlador de DRAM y el adaptador al bus PCI proporciona el acoplamiento al procesador y la posibilidad de generar datos a velocidades elevadas. El adaptador actúa como un registro de acoplo (buffer) de datos puesto que la velocidad del bus PCI puede diferir de la capacidad de E/S del procesador.

³⁵ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 81

En un sistema multiprocesador (figura 38), se pueden conectar mediante adaptadores una o varias configuraciones PCI al bus de sistema del procesador. Al bus de sistema se conectan únicamente las unidades procesador/cache, la memoria principal, y los adaptadores de PCI. De nuevo, el uso de adaptadores mantiene al PCI independiente de la velocidad del procesador y proporciona la posibilidad de recibir y enviar datos rápidamente.

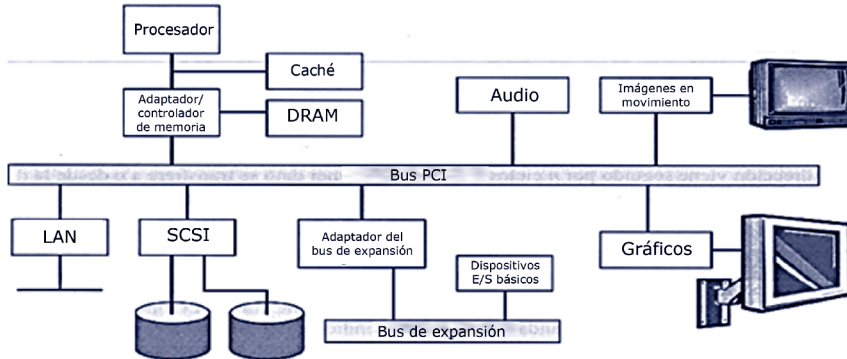


Figura Nro 36. Sistema de Escritorio Típico

³⁶ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España*: Pearson Education p. 81

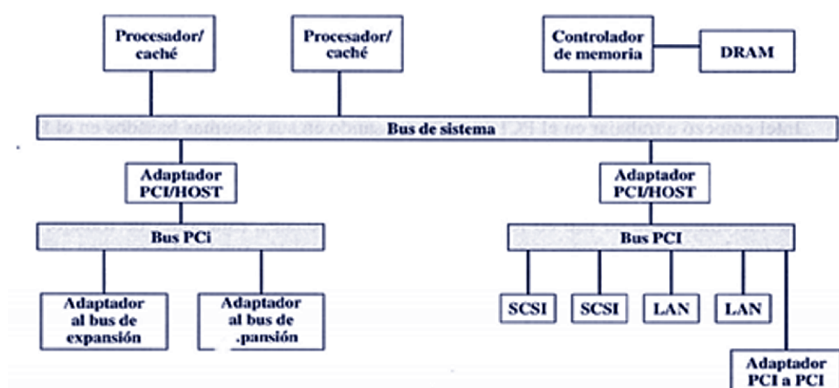


Figura Nro 37. Sistema de Servidor Típico

2.3 Jerarquía de la Memoria

Las memorias de los computadores, presentan tal vez la más alta diversidad de tipos, tecnología, estructura, prestaciones y coste, de entre todos los componentes de un computador. Ninguna tecnología es óptima para satisfacer las necesidades de memoria de un computador. En consecuencia, un computador convencional está equipado con una jerarquía de subsistemas de memoria, algunos internos (directamente accesibles por el procesador) y otros externos (accesibles por el procesador mediante módulos de entrada/salida).

2.3.1 Por qué de la jerarquía?

Las memorias que están en la parte superior de la Jerarquía son las más rápidas, pero tener un computador donde toda su memoria principal estuviera constituida por únicamente Memorias Cache sería sumamente caro, por supuesto que obtendríamos una alta velocidad de proceso del computador, solo que a un costo que bordearía los millones.

Es importante relacionar la cantidad de Memoria a poder direccionar con el ancho del bus de direcciones, para ello si el Bus tiene 4 bits entonces se podrá direccionar hasta 24 direcciones diferentes, empleamos la fórmula 2^n , donde n es el número de bits.

De igual manera el ancho del bus de datos está relacionado con la capacidad de almacenamiento de cada Registro de memoria, un bus de datos de 8 bits indica que en cada registro de memoria se almacena un Byte, un bus de datos de 32 bits indica que cada registro almacena 4 Bytes, cuanto más ancho sea el bus de datos en un mismo registro se tendrá más información por lo tanto se podrá procesar en un único ciclo de reloj más información.

Como sabemos la memoria Ram está constituida por millones de Celdas, cada una esperando por alguna operación a efectuar, pero que además debe ser constantemente refrescada con energía eléctrica.



Figura Nro 38. Jerarquía de la Memoria del Computador

2.4 Características por las cuales se clasifican las memorias

Algunas de las formas por las cuales las memorias se pueden clasificar son: su Ubicación, Interna, Externa; Capacidad, Unidades de transferencia, métodos de acceso, Prestaciones, dispositivo físico, Características físicas y Organización.

A continuación se describen algunas de ellas:

2.4.1 Organización de las memorias

El elemento básico de una memoria semiconductora es la celda de memoria. Podemos imaginar la memoria como un conjunto de casillas, cada una con una dirección que la identifica, donde se almacenan los datos y las instrucciones correspondientes a los programas. Para conocer la ubicación de cada dato estas casillas deben estar convenientemente numeradas (direccionar), es lo que se denomina dirección de memoria. En cada casilla podremos almacenar una determinada cantidad de bits según el computador, 8bits, 16 bits, 32 bits,... El número de bits que almacena un computador en cada casilla de la memoria y que puede manipular en cada ciclo se denomina longitud de palabra ("word" en inglés).

Aunque se utilizan diversas tecnologías todas comparten ciertas propiedades:

- Presentan dos estados estables (o semiestables), que pueden emplearse para representarse el 1 y el 0 binarios.
- Pueden escribirse en ellas (al menos una vez) para fijar su estado.

2.5 Pueden leerse para detectar su estado. Tipos de Memoria RAM

2.5.1 Por su construcción:

2.5.1.1 RAM Estática y RAM dinámica:

Las memorias estáticas mantienen su contenido mientras simplemente esté alimentada. En cambio la Memoria RAM dinámica requiere de Refresco, simplemente para mantener la información, si consideramos la lectura es destructiva, es decir que la información se pierde al leerla, para evitarlo hay que restaurar la información contenida en sus celdas, efectuando también la operación refresco.

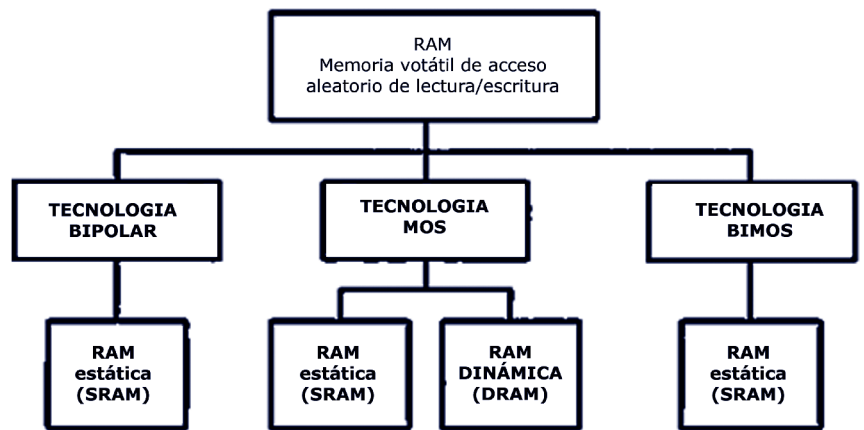


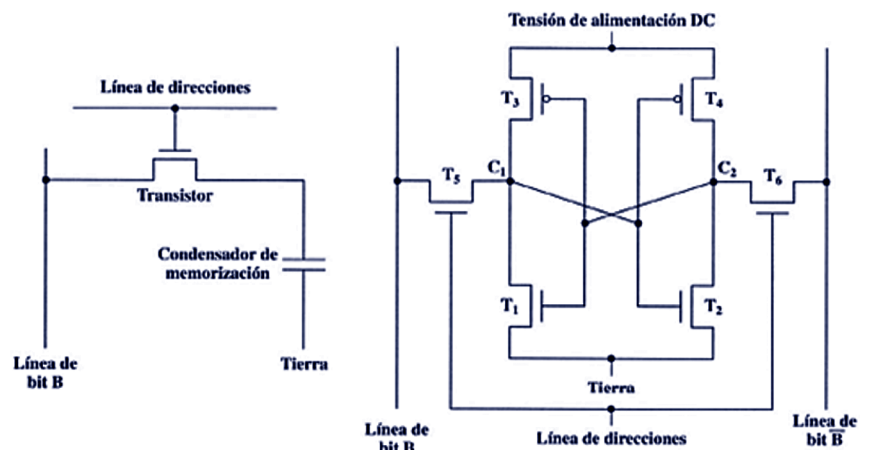
Figura Nro 39. Memorias Ram en función de la Tecnología Empleada

Las SRAM son más rápidas pero las DRAM puede almacenar muchos más datos en el mismo tamaño y precio, ya que las celdas de las DRAM son más sencillas y ocupan menos espacio.

En los sistemas de memoria es habitual utilizar técnicas de corrección de errores. Esto implica añadir bits redundantes, que se deducen a partir de los bits de datos, para formar un código de corrección de errores. Si ocurre un error en un bit el código lo detectará y normalmente lo corregirá.

SDRAM: Es rápida y relativamente económica. Está sincronizada al reloj para que los datos puedan ser enviados al CPU con cada pulso del reloj, aumentando el número de instrucciones que el procesador puede ejecutar dentro de un tiempo determinado

2.5.1.2 Circuitería de las Celdas



Celda de RAM Dinámica Celda de Memoria Estática (SRAM)

Figura Nro 40. Circuitos de memorias RAM

Algunos detalles adicionales a tomar en cuenta al comparar las memorias Estáticas y las Dinámicas son:

³⁷ Cecilio Blanco Viejo(2005). *Electrónica digital – España: Paraninfo.*

³⁸ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education p. 152*

- En lugar de los diminutos acumuladores de carga de las memorias DRAM, en las memorias SRAM se emplean transistores.
- Cada bit se almacena en un grupo de transistores:
- No necesita ciclos de refresco - mayor rapidez.
- Aumento de tamaño - problemas de integración
- Más caro de producir.
- Esto motiva que se utilice para memoria caché y no como memoria principal.

2.6 Memoria ROM(Read Only Memory)

Como sugiere su nombre, es una memoria de solo lectura, contiene un patrón de datos permanentes que no pueden ser cambiados, una ROM es no volátil, es decir no se requiere de fuente de alimentación para mantener memorizados los valores de los bits.

Es posible leer pero no escribir. Una aplicación importante de las ROM es la micro-programación. Otras aplicaciones importantes son:

- subrutinas de bibliotecas para funciones muy buscadas.
- programas de arranque de un sistema. BootStrap
- Tablas de funciones. Tabla de Códigos ASCII.
- Programa de autodiagnóstico. POST (Power On Self Test)

Cuando se requiere un tamaño modesto, la ventaja de la ROM es que el programa o los datos estarían permanentemente en memoria principal y nunca sería necesario cargarlos desde un dispositivo de almacenamiento.

Una ROM se construye como cualquier otro chip de circuito integrado, con los datos cableados en el chip durante el proceso de fabricación.

Cuando se necesitan solo unas pocas ROM con un contenido particular, una alternativa más económica es la ROM Programable.

2.6.1 Tipos de memoria rom:

a) Read Only Memory

Contiene datos permanentes que no pueden ser cambiados, i.e. no volátil. Es posible leer pero no escribir, los Datos son insertados como parte del proceso de fabricación, aplicaciones:

- librerías subrutinas para funciones muy buscadas.
- programas de arranque de un sistema.

b) Memoria tipo PROM

Programmable ROM, es No volátil, Puede escribirse solo una vez. La escritura es eléctrica y requiere de un proceso especial, proporcionan flexibilidad.

c) Memoria tipo EPROM

Erasable Programmable ROM. Antes de escribir debe borrarse el contenido, de Lectura y escritura eléctrica (como en PROM). El borrado se realiza a través luz ultravioleta. Es más cara que la PROM.

Ventaja: capacidad de actualización múltiple

d) Memoria tipo EEPROM

Electrically EPROM. Puede escribirse sin borrar el contenido anterior. Solo los bits o bytes direccionados son actualizados. Operación escritura más tardada que la de



lectura en orden de varios cientos de microsegundos por byte

e) Memoria tipo Flash Memory

Flash: velocidad en la que puede ser reprogramada. Fue introducida por primera vez a mediados de los 80's. Tienen un costo y funcionalidad: entre EPROM y EEPROM. Usa borrado eléctrico y borra en unos segundos. Borrado a nivel bloques y no a nivel byte

2.7 Memoria cache

Desde la aparición de micros con frecuencias de reloj de 25 MHz y mayores, la memoria de trabajo formada por chips DRAM con tiempos de acceso > 60 ns (16,6 MHz) se volvió más lenta que la CPU, por lo que el micro debía insertar ciclos de espera durante los accesos a memoria.

La adopción de memoria con tecnología SRAM (RAM estática), mucho más rápida, es una solución demasiado cara cuando se necesitan grandes cantidades, como sucede con la memoria de trabajo.

Una solución de compromiso podría ser disponer de una “pequeña” cantidad de memoria SRAM donde guardar los datos/instrucciones más recientes para así poder acceder a ellos más rápidamente. Puede pensarse en la caché como en la mesa de trabajo, donde guardamos los papeles que más a menudo usamos, mientras que la memoria principal sería el archivo que se encuentra en otra planta. Cuando el procesador necesita un dato/instrucción, mira primero en la caché, si no está ahí, se dirige a la memoria principal más lenta. Finalmente tendremos que la memoria Cache se puede entender como una memoria intermedia entre el Procesador y la memoria RAM. En la Imagen podemos observar un Procesador Pentium III, que lo mostramos por ser muy ilustrativo en este tema:

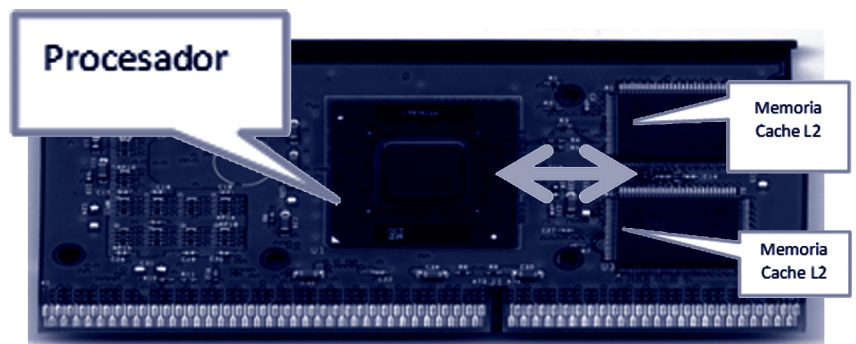


Figura Nro 41. Procesador Intel Pentium II Núcleo y Memoria Cache

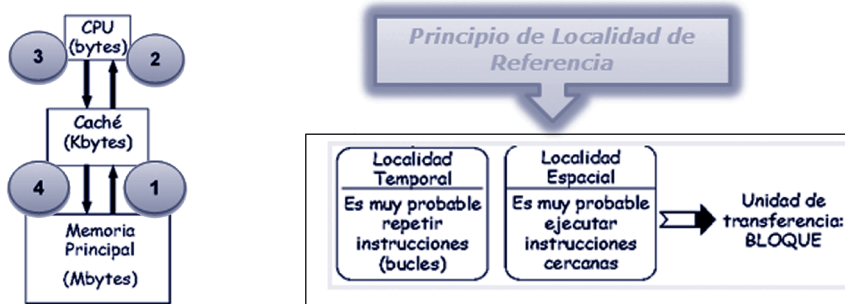
Una ventaja directa de esta implementación es la cercanía al procesador así como la frecuencia de trabajo del Bus que está conectado directamente al procesador lo cual le permite trabajar a altas frecuencias tal igual como el procesador, consecuentemente se tiene una alta tasa de transferencia.

Cuando un programa está corriendo y la CPU necesita leer datos o instrucciones desde la RAM, la CPU verifica primero para ver si los datos están en la memoria caché. Si los datos que necesita no están ahí, se produce lo que llamamos un Fallo de página, lo que le obliga a iniciar un proceso de búsqueda en la RAM, pero también carga una copia de los datos en la memoria caché. La siguiente vez que la CPU necesite los mismos datos, los encontrará en la memoria caché y ahorrará tiempo necesario para cargar los datos desde la RAM.

Conociendo el tamaño de la mayor parte de los programas y muchos archivos de datos, podrás pensar que las probabilidades de que la CPU encuentre los datos que necesita en la memoria caché son pequeñas, pero en realidad encuentra los datos que necesita con la frecuencia suficiente para mejorar el desempeño de una PC.

Las instrucciones de programas a menudo se encuentran en la memoria caché. Con frecuencia, los programas piden a la computadora que haga la misma operación de manera repetida hasta que se cumpla una condición particular; a esto le denominamos Principio de Localidad de Referencia. En lenguaje de programación de computadoras, este proceso repetitivo se llama ciclo o bucle (inglés: loop).

Finalmente recordemos que las memorias cache están construidas con tecnología SRAM, lo que les permite tiempos de respuesta muy cortos en comparación a las DRAM



a) Flujo entre la Ram y Cache b) Principio de Localidad

Figura Nro 42. Memoria Cache Flujo y Principio de localidad

Conceptualmente el funcionamiento de una memoria caché es muy simple. La circuitería de control de la memoria está diseñada para aprovechar la localidad de referencia. De acuerdo con el aspecto temporal de la localidad de referencia, siempre que un dato se necesita por primera vez, se debe traer a la memoria caché, en la que permanece, de tal forma que cuando se le vuelva a necesitar, se podrá obtener muy rápidamente.

Por su parte, la localidad espacial aconseja que en lugar de traer a la caché solamente el dato referenciado, se traiga el grupo de datos que reside en direcciones adyacentes a dicho dato. En el contexto de las cachés, se emplea el término bloque para referirse a un conjunto de datos con direcciones contiguas que se utiliza como unidad de transferencia entre la memoria principal y la caché. El espacio que tiene una caché para albergar un bloque de memoria principal se denomina línea.

Como se muestra en la figura 43, cuando se recibe una petición de lectura de la CPU, el contenido del bloque completo de memoria principal que contiene la dirección especificada se transfiere a la caché. Posteriormente, cuando el programa referencia cualquiera de las direcciones del bloque, el contenido correspondiente se lee directamente de la caché (solamente la palabra o parte de la palabra referenciada, no el bloque completo).

³⁹ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p. 111

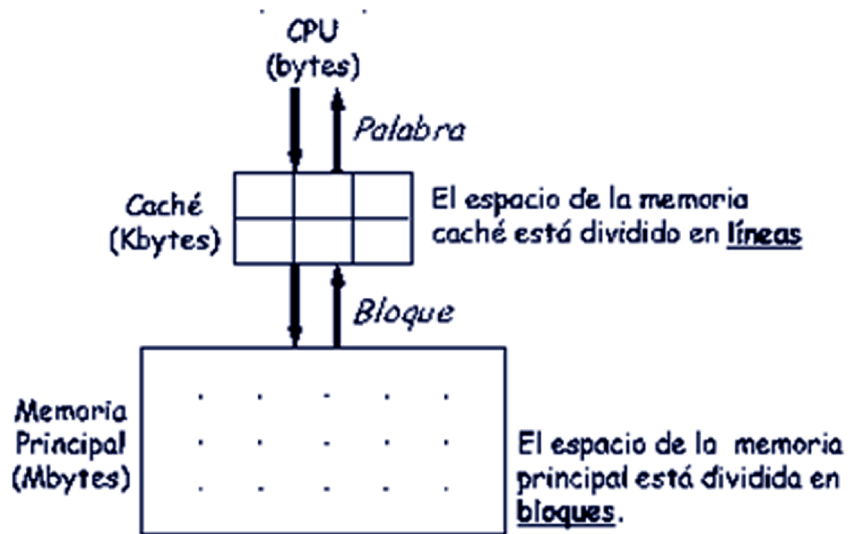


Figura Nro 43. Comparación de la estructura de la RAM y la Cache

En la figura 44, se aprecia la memoria caché L2 del CPU de AMD. También, se aprecia a la memoria caché L1 que está dividida en dos: una parte almacena las instrucciones de los programas que se están ejecutando; y la otra parte, los datos, que son resultado de la ejecución del programa. Por ejemplo, si se está escribiendo una carta, las instrucciones del programa MS-Word deberán estar en L1 (parte de instrucciones); y el contenido de la carta, en L1 (parte de datos). La mayoría de los procesadores actuales usan, también, la memoria caché L3, que es de mayor capacidad que L1 y L2.

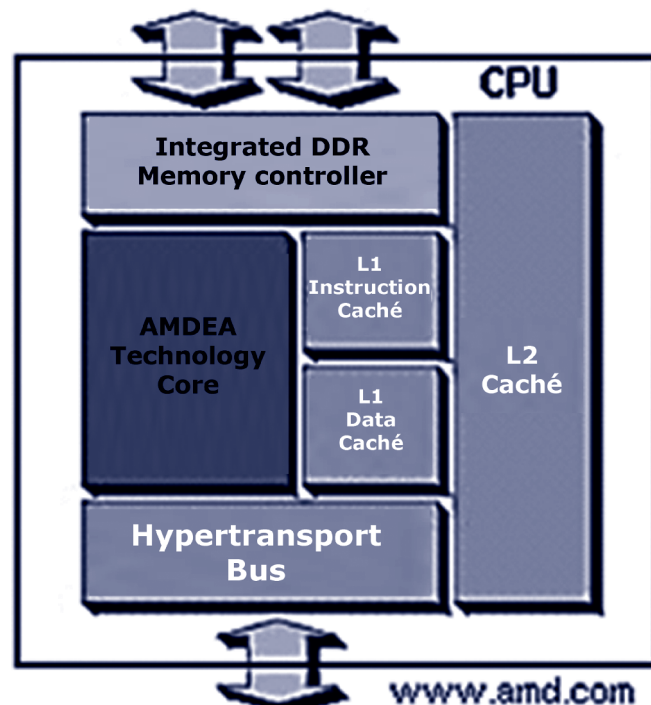


Figura Nro 44. Memoria Cache en Procesador AMD

⁴⁰ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p. 111

⁴¹ AMD. Características básicas de la arquitectura del procesador AMD Sempron Recuperado de : <http://www.amd.com/es/products/desktop/processors/sempron/Pages/AMD-sempron-processor-key-architectural-features.aspx>

2.8 Memoria Virtual:

La memoria virtual es una técnica que permite la ejecución de procesos- programas Parcialmente cargados en memoria principal. Con la característica que los programas pueden ser más grandes que la memoria RAM instalada.

Se utiliza el disco duro como almacén secundario de procesos. Debemos considerar que al tener memoria Virtual, esto no provoca que un sistema tenga mejor performance, solo permite tener más programas en estado de listo para ejecución.

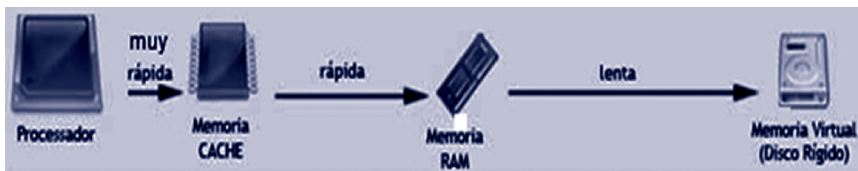


Figura Nro 45. Velocidad de acceso a Memoria Virtual

Nuevamente la memoria Virtual es posible de aplicar por el principio de Localidad de referencia, es decir podemos llevar a memoria RAM únicamente las partes del programa que se requieren, esto está básicamente respaldado por la programación estructurada y la programación modular.



TEMA N° 3: ENTRADA Y SALIDA

El procesador requiere comunicarse con otros elementos, como son: memoria RAM, memoria ROM, tarjetas de video, sonido, red y ranuras de expansión. Para ello se emplea un elemento base que es la MainBoard o simplemente Placa.

3.1 LA PLACA PRINCIPAL:

Llamada también motherboard o mainboard, maneja todas las transferencias de datos entre la CPU y los periféricos. Alberga la CPU, la caché de tercer o segundo nivel, el chipset, el BIOS, la memoria principal, los chips de entrada/salida, los controladores de discos y las tarjetas e expansión.

La primera decisión a realizar antes de comprar un motherboard es qué CPU y qué chipset se usará. Lo siguiente es elegir un fabricante.

Es un circuito impreso (PCB: Printed Circuit Board) que soporta y conecta físicamente los elementos fundamentales de un computador como son Microprocesador, memoria, chipset, tarjetas de periféricos, conectores, elementos electrónicos (condensadores, bobinas, etc.) etc.

En la imagen podemos observar una placa para Intel para un procesador Core i7.

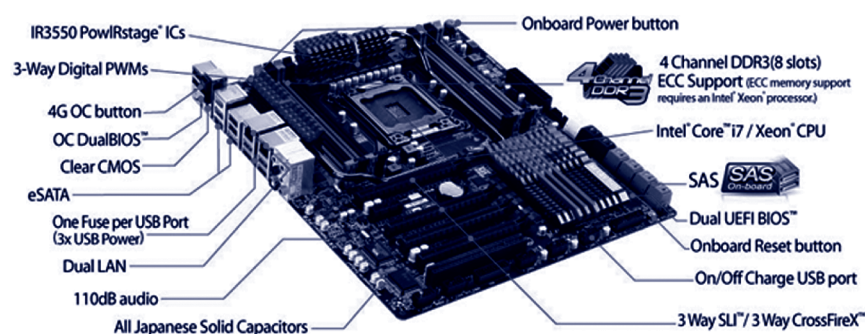


Figura Nro 46. Placa para procesador Core i7

Otro ejemplo es la siguiente tarjeta madre, donde se aprecian los componentes modernos la Desktop Board Intel® DH55PJ en factor de formato microATX ofrece un equilibrio entre las características tradicionales y las premium en un nuevo diseño con bajo consumo de energía. Veamos las características de este mainboard:

- Compatibilidad con memoria DD3-1333MHz
- Dos ranuras para hasta 8GB de memoria RAM del sistema
- Controlador de sonido Intel de alta definición (HD), integrada, con sonido envolvente 5.1
- Controlador de red Intel PRO10/100/1000 Mb/s integrada.
- Un slot o ranura PCI para las tarjetas antiguas o Dos slots PCI express 1X para tarjetas de red, sonido que permitan reemplazar a las que están integradas
- Un slot PCI express 16X para vid

⁴² WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education* p. 175-207

⁴³ Gigabyte.com(10/12/2013). Recuperado de: <http://es.gigabyte.com/products/upload/products/3429/schematic.jpg>

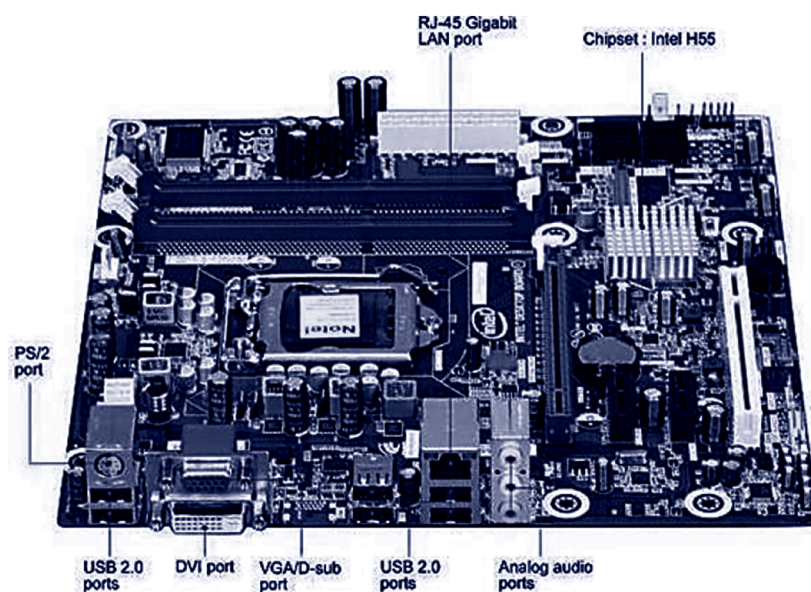


Figura Nro 47. Placa DH55-Intel

Esta placa es compatible con los procesadores Intel Core i7, Core i5, Core i3 e Intel Pentium en el paquete LGA1156. Esta mainboard cuenta con conectores VGA y DVI-D y es compatible con visualización doble independiente para procesadores con Gráficos Intel HD.

3.1.1 Componentes de la mainboard

Toda PC está formada por una placa base y una serie de placas de expansión o tarjetas controladoras, fabricadas en peltrax, un derivado de la fibra de vidrio, cada una de las cuales está destinada a cumplir una función específica, dependiendo del tipo de controlador que tenga la placa

Es en la mainboard donde se alojan los principales componentes de una PC y se conectan las tarjetas de expansión. Más allá de las diferencias originadas por las distintas generaciones, en todos los mainboard encontraremos una serie de componentes comunes. El componente en el que se conecta una tarjeta se le denomina, en forma genérica, slot o zócalo, pudiendo en algunos casos denominárselo conector; además, se encuentra los puertos, los cuales permite conectar dispositivos, así de esta manera, en toda PC tendremos un puerto para el teclado, un puerto para el monitor, etc

Los componentes básicos que se encuentran en la placa son:

3.1.1.1 Zócalo del Microprocesador:

Este es el zócalo donde se coloca el microprocesador. Las medidas y la cantidad de contactos varían según el tipo de procesador utilizado. Posee los anclajes necesarios para la colocación del cooler.

⁴⁴ Intel (10/12/2013). Recuperado de: www.intel.com

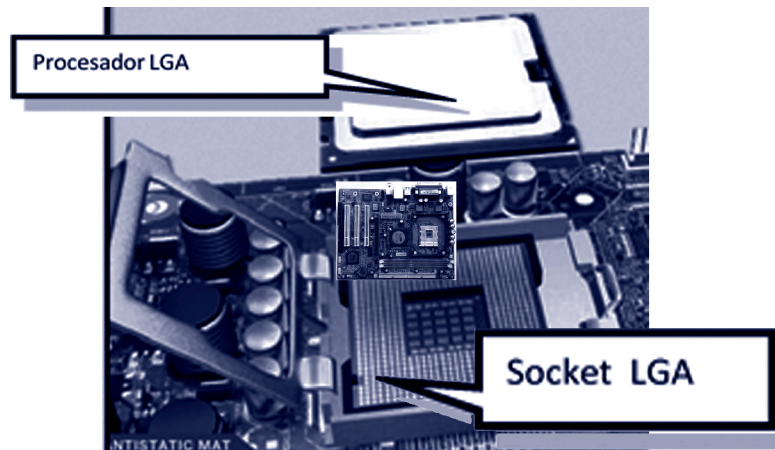


Figura Nro 48. Socket LGA

3.1.2 Chipset Northbridge.

También llamado “Puente Norte”, es el encargado de proporcionar el bus del procesador y controlar la memoria, así como de administrar el tráfico con el bus AGP o PCI Express. Su función principal es interconectar a tres elementos importantes de alta velocidad: el procesador, la RAM y la tarjeta de video. En la actualidad, en algunas modelos de PC’s este chip ha sido absorbido e integrado en el propio CPU o procesador

Como es un dispositivo de alta velocidad, genera mucho calentamiento, por lo que usa un disipador para liberar el calor, este chip es cubierto por el disipador de aluminio y por ello no se aprecia el chip.

También llamado MCH por Intel (MCH: Memory Controller Hub). Comunica a la CPU con el resto del sistema, para lo que contiene:

- Interfaz con el bus externo del micro (bus del sistema).
- Controlador de memoria (bus de memoria).
- Interfaz con el sistema gráfico (bus gráfico).
- Interfaz con el puente sur (bus de enlace).

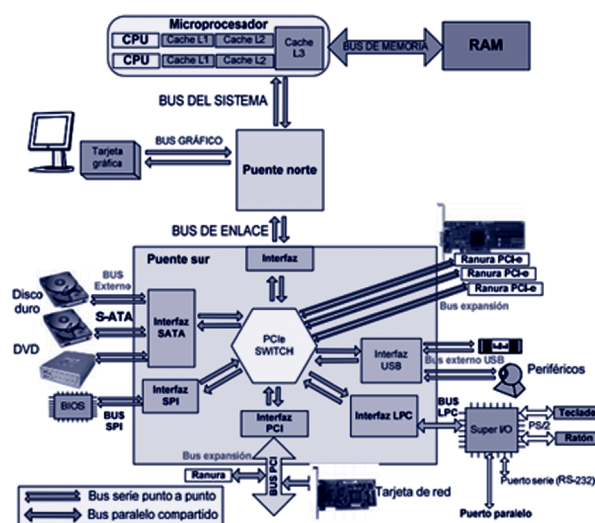


Figura Nro 49. Funciones de los Chipsets Norte y Sur

⁴⁵ Ignacio Moreno Velasco(2012). *Mantenimiento de equipos informáticos. España: Universidad de Burgos p.4*

⁴⁶ Ignacio Moreno Velasco(2012). *Mantenimiento de equipos informáticos. España: Universidad de Burgos p.4*

3.1.3 Chipset Southbridge.

También llamado “Punto Sur”, es la parte del chipset encargada de brindar conectividad entre el resto de dispositivos, sobre todo los de menor velocidad, ya que el puente norte se reserva la conexión de los tres elementos más rápidos. Lleva integrados diferentes controladores y por ello decimos que controla a los discos duros, al bus PCI, al PCI express, a los puertos USB, a los puertos del teclado, red, entre otros. También es llamado ICH por Intel (ICH: I/O Controller Hub), para lo cual contiene:

- Interfaz con el puente norte (bus de enlace).
- Interfaz con el bus de expansión: Controlador PCI, (PCI-Express).
- Dispositivos PCI integrados: controladora USB (Interfaz USB-PCI), Controladora IDE (discos duros y unidades ópticas), etc.
- Dispositivos estándar heredados (controlador DMA, controladores de interrupción 82C59, RTC y memoria CMOS).

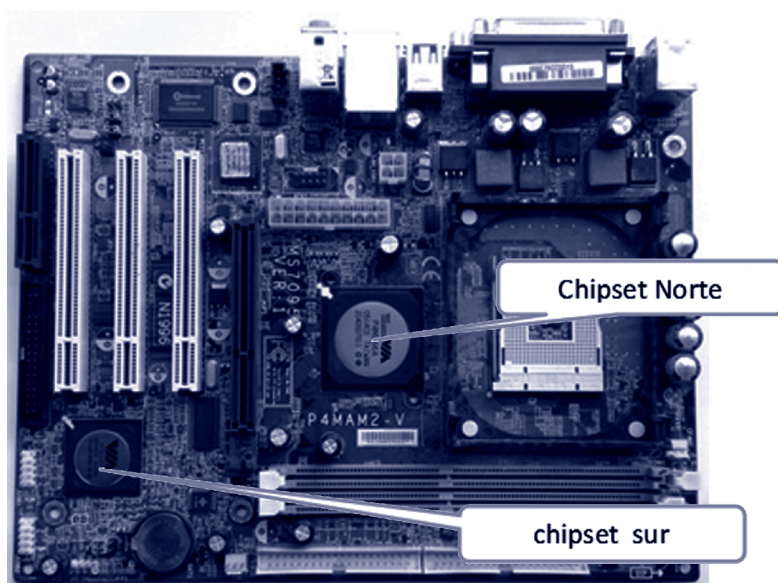


Figura Nro 50. Chipset Norte y Sur

El diagrama en bloque que a continuación se muestra, confirma lo expresado para ambos chipset

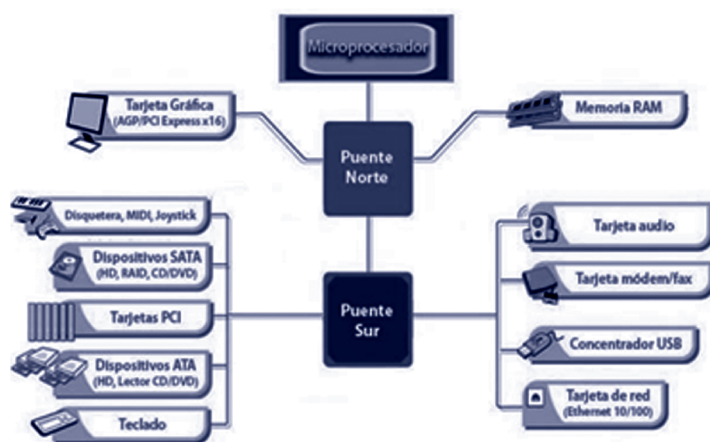


Figura Nro 51. Esquema Funcional de una placa actual

⁴⁷ VGALegacy.com. (19/09/2010) Recuperado de: <http://www.vgamuseum.info/index.php/component/content/article/805-via-p4m266>



La tendencia actual es agrupar cada vez más funciones dentro del chipset. Muchos de los circuitos integrados originales del XT y AT, como el controlador de interrupciones 8259 (PIC: Programmable Interrupt Controller), el controlador DMA (Direct Memory Access: 8237), el reloj de tiempo real RTC (Real Time Clock), etc... se encuentran integrados en el puente sur del chipset.

3.1.4 BUSES:

Los buses habituales de una placa base son actualmente:

- Bus externo del micro. Llamado Host Bus ó Bus frontal del sistema (FSB: Frontal Side Bus), en placas para Core i7, son QPI (Quick Path Interconnect).
- Bus de memoria.
- Bus del sistema gráfico (Antes AGP, ahora PCI Express).
- Buses de expansión: (Actualmente PCI, PCI Express).
- Buses externos: USB, ATA y/o Serial ATA.
- Bus de enlace (puente norte-puente sur)
- Bus de gestión del sistema SMBus (System Management Bus).
- Bus LPC (Low Pin Count) de conexión del chip super I/O (Teclado, disquete, puertos serie, etc...)

3.1.4.1 Velocidad de transferencia - Ancho de banda

Cuando se habla de buses, el término “ancho de banda” (BW = BandWidth) se usa equivocadamente para referirse a la cantidad teórica de datos que puede transportar el bus por unidad de tiempo. Sin embargo, el ancho de banda es un parámetro que debe expresarse en Hertzios (Hz) y por lo tanto, debe referirse en todo caso a un rango de frecuencias.

3.1.4.2 Bus de enlace

Denominamos así al bus que enlaza el puente norte y el sur.

Ejemplo: Intel Direct Media Interface (DMI). Direct Media Interface (DMI) es el bus de conexión entre el MCH (puente norte) y el ICH6 (puente sur) de Intel. Algunas características relevantes son:

- Conexión punto a punto de 2 GB/s (1 GB/s en cada dirección)
- Señal de reloj de 100 MHz (compartida con el enlace PCI Express para gráficos).
- Direccionamiento hacia el puente sur de 32 bits (downstream addressing)

3.1.5 Slots para tarjetas de expansión / Buses

Los buses de la PC son las principales autopistas de datos en un motherboard. Conforman el sistema nervioso del motherboard, ya que conectan la CPU con los demás componentes. En concreto, los buses son circuitos impresos en el motherboard que transmiten los datos entre los diferentes componentes.

Básicamente, se los puede dividir en dos tipos:

- El bus de sistema, que conecta la CPU con la RAM.
- Los buses de salida/entrada, que conectan dispositivos (buses I/O). A través de los años, han surgido diferentes tipos. Pasamos a analizarlos a continuación:

3.1.5.1 ISA

⁴⁸ Sites.google.com. Recuperado de: <https://sites.google.com/site/mipcyog/chipset>

ISA es el acrónimo de Industry Standard Architecture. Es la arquitectura de bus que fue introducida como un bus de 8 bits con la PC IBM original en 1981. Más tarde, fue expandido a 16 bits con la IBM PC AT en 1984. Las últimas versiones de ISA disponen de un bus de 16 bits a 8 MHz. La máxima capacidad de transferencia teórica de este bus es de 8 MB/seg. El conector o slot ISA es fácilmente identificable en un motherboard. Es de color negro y dispone de 49 contactos a ambos lados. Además, su tamaño es mayor que el de los otros slots del motherboard. Si bien en un tiempo fueron muy usados para placas de video y sonido, su escaso ancho de banda hace que tiendan a desaparecer. Hoy en día, los motherboards modernos incluyen apenas uno de ellos para la conexión de dispositivos de baja salida, como algunos módems telefónicos o tarjetas de sonido. En el futuro cercano, desaparecerán; de hecho, algunas PCs de marca ya no los incluyen.

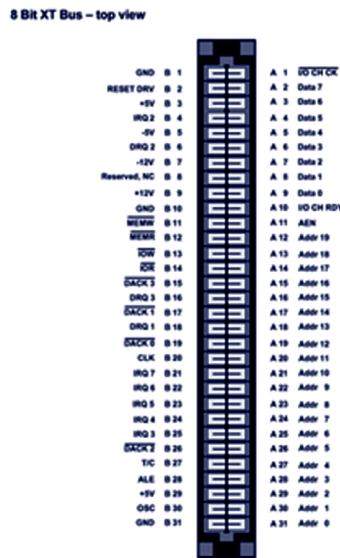


Figura Nro 52. Bus Isa

3.1.5.2 PCI

Este bus de datos funciona a 33 MHz, al mismo ancho de datos que la CPU. Es decir, 32 bits. De ser usado con una CPU de 64 bits, el bus PCI doblaría su ancho de datos. El límite teórico de transferencia a 32 bits es de 132 MB/seg. Externamente, el bus PCI dispone de slots más chicos que ISA y, en general, de color blanco (esto es así por norma). Existen tres tipos de especificaciones, determinadas según las necesidades eléctricas. La especificación de 5 voltios para PCs de escritorio; la de 3,3 para notebooks; y una universal que soporta ambos voltajes. Las tarjetas PCI disponen de una ranura que orienta al usuario respecto de la manera en que deben ser instaladas.

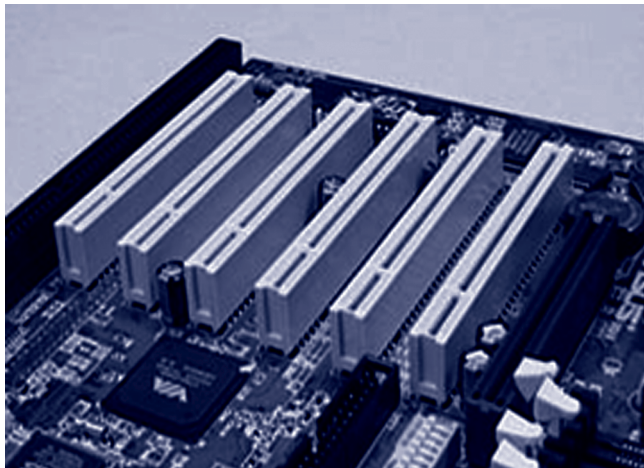


Figura Nro 53. Slots de arquitectura PCI

Fue el zócalo de expansión más usado, pero en la actualidad ha sido reemplazado por el PCI Express, ya que compañías, como IBM, 3Com, Adaptec, HP y Compaq, impulsan una versión especial de alta velocidad para servidores del bus PCI llamada PCI-X. Este nuevo estándar permite un ancho de banda de 1 GB/segundo (con un bus de 64 bits funcionando a 133 MHz).

⁴⁹ Industry Standard Architecture(25/11/2005). En Wikipedia: http://en.wikipedia.org/wiki/File:XT_Bus_pins.png

⁵⁰ Peripheral Component Interconnect(26/11/2013). En Wikipedia. Recuperado de: http://es.wikipedia.org/wiki/Peripheral_Component_Interconnect

3.1.5.3 Slot de Expansión PCI Express 16X.

En el gráfico se muestra dos slots PCI express 16X, de color rojo y azul, acompañado de dos slots PCI y dos slots PCI express 1X, siendo estos los más pequeños



Figura Nro 54. Slots Pci Express(Rojo y Azul) PCI(Blanco)

3.1.6 Conectores Externos – Puertos Hardware.

Muchas veces también denominados puertos simplemente, permiten la relación del computador con el exterior, se encargan de facilitar el intercambio de información entre el computador y la periferia. Su función consiste en posibilitar la transmisión de datos entre dos sistemas distintos.

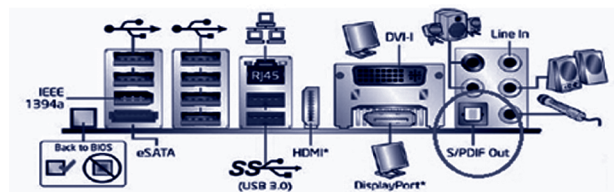


Figura Nro 55. Conectores del Panel posterior

Los puertos los podemos tener ya integrados en la placa base como ya vienen actualmente, pero en las placas anteriores a Pentium, no venían integrados en la placa base.

3.1.7 Puerto serie

Fue diseñado para permitir una comunicación de dos vías entre distintos dispositivos y la PC. Es usado para conectar, por ejemplo, el mouse. Su interface externa se caracteriza por tener nueve pines. Los motherboards incluían dos, denominados COM1 y COM2. El corazón de cualquier puerto serie es el chip UART (Universal Asynchronous Receiver/Transmitter). Este chip se encarga de controlar el proceso de envío y recepción de datos. En la actualidad se siguen empleando con adaptadores para USB sobre todo para comunicaciones Seriales por ejemplo con el Puerto de Consola de Routers y Sw



Figura Nro 56. Interfaces de Puertos serial- DB9 Macho

⁵¹ Tomshardware(10/12/2013). Recuperado de: www.tomshardware.com

⁵² Intel (10 de octubre de 2011). Desktop Boards, S/PDIF information. <http://www.intel.com/support/sp/motherboards/desktop/sb/cs-032871.htm>

⁵³ Rodrigo Pacheco(22/10/2009). Recuperado de: <http://culturacion.com/2009/10/interface-serie-y-paralela/>

3.1.8 Puerto paralelo

Los puertos paralelos son normalmente usados para conectar la impresora a la PC. Aun cuando ésa fue su intención original de diseño, esta tecnología ha evolucionado hacia múltiples funciones y, ahora, sirve como interface entre varios dispositivos. Originariamente, los puertos eran unidireccionales; los puertos paralelos modernos, bidireccionales, pueden enviar y recibir datos. Los puertos paralelos llevan este nombre porque tienen ocho líneas para enviar todos los bits que componen un byte de datos en forma simultánea a través de ocho cables. Esta interface es más rápida que el serial y ha sido usada como una opción para transmitir datos entre sistemas. Con el tiempo, han proliferado distintas variantes de puertos. Todas cumplen con la norma original IEEE-1284, pero incorporan distintas mejoras. Así es que, en la actualidad, los motherboards incluyen puertos paralelos multimodo. Desde el BIOS, se pueden configurar como SPP, EPP o ECP.



Figura Nro 57. Interface de Puerto Paralelo-DB25 Hembra

3.1.9 USB

El Universal Serial Bus (USB) es un estándar que permite conectar dispositivos Plug & Play externos a la PC. USB elimina la necesidad de instalar tarjetas, que ocupan valiosos slots, y reconfigurar el sistema, lo que ahorra importantes recursos, como las interrupciones (IRQs). Las computadoras equipadas con USB permiten la configuración inmediata de los dispositivos que utilizan esta interface no bien son enchufados, sin necesidad de apagar o reiniciar el sistema completo. A diferencia de los cables paralelos o seriales, los conectores USB son pequeños y no necesitan ser sujetados con tornillos. USB 1 constituye una interface de 12 Mbit/seg a través de una conexión de cuatro hilos dentro de un cable..



Conectores de USB 3.0 y USB 2.0

⁵⁴ Iden(53)

⁵⁵ Thomas Soderstrom(14-04-2010), Portable Performance: Four USB 3.0 Enclosures For Your 2.5" Drive()Recuperado de: <http://www.tomshardware.com/reviews/usb-3.0-superspeed-external-enclosure,2597-2.html>

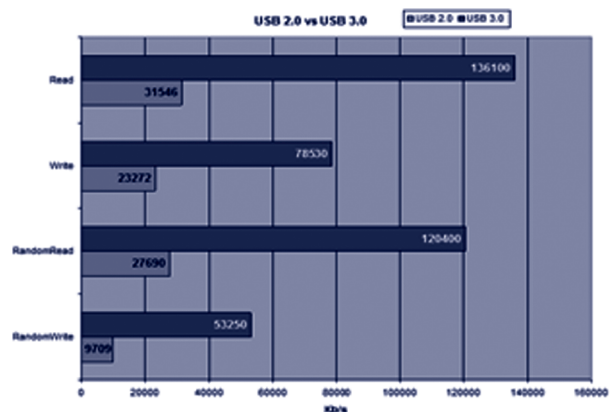


Figura Nro 58. Comparativa de Rendimiento USB 3.0 Vs 2.0

El bus soporta hasta 127 dispositivos gracias a que se pueden formar redes de dispositivos a través de hubs de expansión que pueden estar en la PC o en otros dispositivos USB, como teclados y monitores; incluso, hay hubs USB dedicados. Si bien esta capacidad es, en principio, impresionante, hay que considerar que todos los dispositivos tendrán que compartir el mismo ancho de banda de 12 Mbits/seg. O sea, cada vez que se agrega un dispositivo a la cadena, la performance se puede ver afectada. En la práctica, muy pocos usuarios querrán tener más de cinco dispositivos conectados a la vez. Una de las ventajas más grandes de una interface como USB es que sólo necesita una interrupción de la PC, aunque estén conectados múltiples dispositivos.

En la actualidad podemos encontrar hasta 3 versiones de USB, USB1, USB 2 y USB3

3.1.10 IEEE-1394

IEEE-1394 es una tecnología de bus relativamente nueva, es el resultado de las grandes demandas de transferencia de datos de los dispositivos multimedia de la actualidad. Es extremadamente rápido, con tasas de transferencia de hasta 400 Mbits/seg (50 MB/seg), y velocidades aún más rápidas están en desarrollo. 1394 también es conocido por otros dos nombres comunes: i.Link y FireWire. i.Link es la designación adoptada por Sony, en un esfuerzo por colocar un nombre más vendedor a la tecnología. La mayoría de los fabricantes de dispositivos 1394 para PCs han apoyado este nombre. En cuanto a FireWire, es una marca registrada de Apple y, en general, se aplica a los dispositivos para Mac.

Tipo	Imagen de puerto	Imagen de conector
6 pines		
4 pines		

Figura Nro 59. Conectores de IEEE 1394-iLink

3.1.11 Controladores de disco

Las nuevas generaciones de discos rígidos usan distintos modos de transferencias de datos para aumentar su velocidad. Para que funcionen al máximo, el

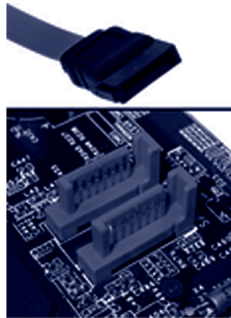
⁵⁶ Iden(53)

⁵⁷ www.tiendacables.com

motherboard debe ser compatible con las especificaciones. Los controladores de disco, integrados en los motherboards, son desarrollos de IDE, conocidos como ATA o UDMA. La norma ATA 33 ofrece una tasa de transferencia máxima de 33 MB/seg; ATA 66 de 66 MB/seg; ATA 100 de 100 MB/seg y la más reciente ATA 133 de 133 MB/seg. Las nuevas interfaces ATA son compatibles con las más viejas. Podríamos conectar un disco ATA 66 a un motherboard ATA 33, pero aquél estaría limitado a 33 MB/seg. Además, la norma ATA 66 exige cintas con mayor cantidad de cables para disminuir las interferencias.



a) Conector IDE



b) Conector SATA



Figura Nro 60. Cables IDE y SATA

3.2 Función de la entrada /salida

Un computador no puede estar formado sólo por la CPU y la memoria. Para darle alguna utilidad debe de comunicarse con el mundo exterior a través del subsistema de entrada/salida (I/O input/output). La misión principal del subsistema de E/S es adaptar los dispositivos externos antes de conectarlos al bus del sistema.

¿Por qué no se conectan directamente al bus del sistema?

1. La velocidad de transmisión de datos de los periféricos es siempre menor que la de la memoria y la CPU.
2. Debido a la gran diversidad de periféricos no es posible incorporar toda la lógica necesaria en el computador para manejar cada uno de éstos.
3. Los formatos de datos de los periféricos son diferentes a los del resto del computador

Hay dos formas de implementar la E/S en un computador:



1. Construir a medida el subsistema utilizando elementos básicos
2. O bien incorporar y programar dispositivos estándares más complejos y genéricos.

Ejemplo primer caso: circuito para que al escribir un uno en una dirección de memoria se ilumine un LED.

3.2.1 Funciones del sistema de E/S:

Las Funciones del bus de sistema son:

1. **DIRECCIONAMIENTO:** selección del dispositivo correspondiente de entre los dispositivos disponibles en el sistema.

2. **SINCRONIZACIÓN:** ha de posibilitar que la CPU y la memoria (alta velocidad transferencia de datos) se puedan comunicar con los dispositivos de E/S (baja velocidad) sincronizando los envíos de datos entre ambos.

3. **TRANSFERENCIA:** el sistema E/S debe de tener toda la circuitería y señales de comunicación apropiadas para llevar a cabo la comunicación con cada uno de los dispositivos del sistema.

3.3 PERIFÉRICOS:

Es un dispositivo electrónico mecánico mediante el cual se implementa la Entrada/salida.

Elemento que permiten la transferencia de información entre la CPU y el mundo exterior. Interfaz que traduce la información asíncrona y analógica del mundo exterior a la información síncrona y codificada del computador.

Está compuesto por dos partes: módulo de E/S y dispositivo (externo)

3.3.1 MÓDULO DE E/S:

Coordina el correcto flujo de información entre uno o varios dispositivos externos (impresora, monitor, ...) e internos (memoria, procesador).

3.3.1.1 FUNCIONES DEL MÓDULO DE E/S :

1. Reconocer la dirección de la CPU que identifica al dispositivo externo.
2. Transferencia de datos entre la CPU y el dispositivo externo.
3. Recepción de mandatos (comandos) desde la CPU.
4. Mantener información del estado del periférico y mantener el protocolo de comunicaciones con el periférico.

Un módulo de E/S puede controlar varios dispositivos externos. El módulo de E/S también almacena datos temporalmente debido a las diferencias de velocidades entre los periféricos y la CPU o la memoria. Dispone de un mecanismo de detección de errores tales como el uso del bit de paridad. Los términos “controlador”, “procesador de E/S” y “módulo de E/S” son equivalentes. La diferencia radica en su complejidad.

3.4 Métodos de programación de Entrada/Salida.

Para lograr comunicarse con los periféricos existen varias técnicas, entre ellas tenemos:

3.4.1 E/S Programada (Encuesta o pooling).

La entrada salida programada se caracteriza por:

- La CPU tiene el control absoluto de la operación de E/S: inicia y lleva a cabo la transferencia.

- La CPU está dedicándose por completo a realizar la operación de E/S: realiza tanto la comprobación de estado como la transferencia y la inicialización: poco eficiente.
- Hardware mínimo.

3.4.2 E/S POR Interrupciones

La E/S le indica a la CPU cuando está preparada para transferir datos (genera una interrupción a la CPU), activando una línea especial conectada a la CPU (línea de interrupción).

3.4.2.1 Funcionamiento:

- 1) El procesador ejecuta instrucciones de un programa. Al finalizar cada instrucción comprueba si se ha producido una interrupción.
- 2) En caso afirmativo se salva el estado actual del programa (contador del programa y registros) y se salta a ejecutar la rutina de servicio correspondiente
- 3) La rutina de servicio efectúa las operaciones apropiadas en la E/S para realizar la transferencia de datos solicitada
- 4) Al finalizar la rutina de servicio se recupera el estado de la CPU y se continúa ejecutando el programa que se estaba ejecutando antes de la interrupción

Las interrupciones pueden ser:

- Enmascarables (se pueden dejar de atender por software)
- NO enmascarables (siempre atendidas).

Generalmente existen varios periféricos (y no uno sólo) conectados que pueden realizar interrupciones, esto obliga a establecer prioridades y decidir cómo se conectan a la CPU. También hay que determinar para cada periférico su vector de interrupciones.

3.4.2.2 SOLUCIONES MAS EXTENDIDAS EN INTERRUPCIONES

a) Una sola línea de interrupción

Todos los periféricos interrumpen por la misma línea, con lo cual el vector de interrupción es fijo y común a todos los periféricos. Se lleva a cabo mediante encuesta (polling) la CPU identifica el periférico y desactiva la interrupción. La prioridad viene determinada por el orden de la encuesta.

b) Varias líneas de interrupción

Cada periférico tiene su línea de interrupción.

Cada línea tiene su propio vector de interrupción asociado y la CPU determina la prioridad.

c) Líneas de interrupción y aceptación:

Una línea de entrada para aceptar interrupciones y otra para dar el reconocimiento de la interrupción al periférico. En el procesador 8086, Hay varias variantes:

- Daisy-chain o encadenamiento.
- Interrupciones vectorizadas.
- Gestión centralizada por un controlador de interrupciones (PIC, programmable interrupt controller)

3.4.3 Entrada salida por DMA (Direct Memory Access).

El DMA es un procesador/controlador especializado en transferencias “muy grandes” desde periféricos a memoria y viceversa. Es programable. La CPU no realiza ninguna tarea (salvo programar el DMA) ya que la inicialización y transferencia son gobernadas por el periférico. Para programar el DMA hay que enviarle al menos los siguientes datos:



- Dirección/puerto periférico E/S.
- Posición/dirección en memoria principal.
- Tamaño (número de bytes a transferir).
- Tipo transferencia: lectura o escritura.

Al finalizar el DMA avisa mediante una interrupción. Esta interrupción al igual que el resto de interrupciones son normalmente atendidas al final de cada instrucción. La rutina de servicio asociada comprobará el estado del DMA para ver si se han producido errores al ejecutar la transferencia que se le ha encomendado.

A diferencia del mecanismo por interrupción convencional, una orden DMA puede transferir muchísimos datos de una sola vez. Por lo tanto, el número de interrupciones por byte transferido es mucho menor que con las interrupciones convencionales: se gana en rapidez.

El procesador no se encarga de la transferencia de datos, significando que puede dedicarse a desarrollar otras tareas inclusive si se está dando una operación de E/S con uso intensivo de memoria.

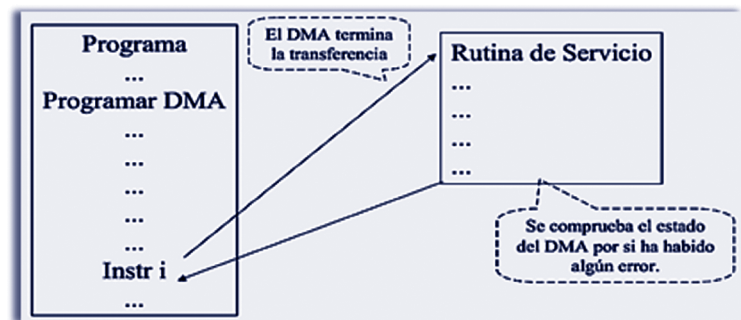
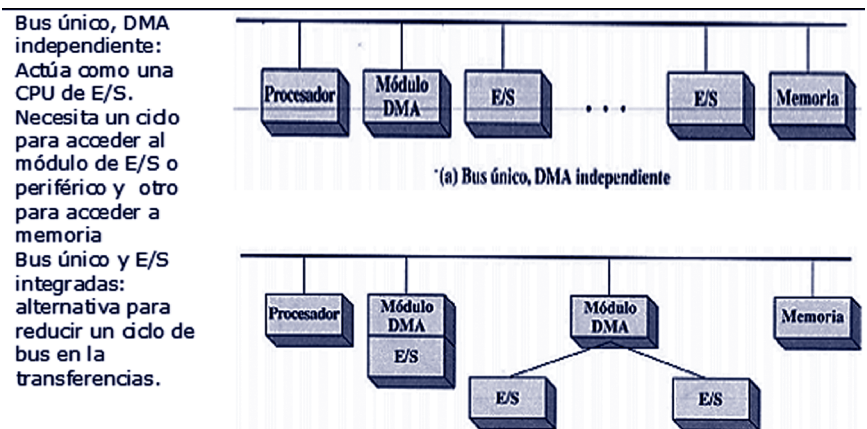


Figura Nro 61. Esquema de DMA

3.4.4 Distintas formas de conectar el DMA al sistema

Los sistemas actuales poseen diversas formas de implementar el DMA, cada una posee ventajas y desventajas propias de la implementación.



⁵⁹ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores* –España: Pearson Education p. 229

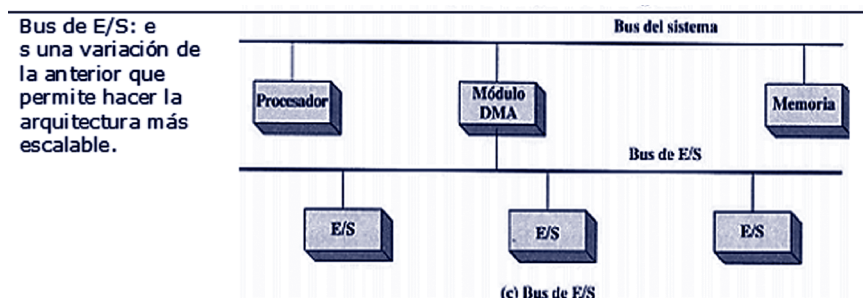


Figura Nro 62. Formas de Conectar DMA al sistema

Existen dos formas básicas de realizar el acceso directo a memoria:

1. MEMORIA MULTIPUERTA:

La memoria permite realizar transferencias simultáneas por parte de la CPU y otros dispositivos. La memoria tiene varias “puertas” que permiten acceso concurrente a un mismo bloque de memoria por lo que hay que establecer un sistema de arbitraje.

2. ACCESO A MEMORIA POR ROBO DE CICLO:

Solución más económica. El DMA cada vez que quiere tomar el control del bus del sistema para realizar la transferencia de un dato, lo solicita a la CPU mediante la señal HOLD. La CPU concede el control del bus mediante la señal HLDA. Después de realizar la transferencia el controlador de DMA lo comunica a la CPU mediante las señales de control correspondientes.

3.4.5 Puertos de Entrada/salida

Cualquier dispositivo (distinto de la memoria RAM) que intercambie datos con el sistema lo hace a través de un “puerto”, por esto se denominan también puertos de E/S (“I/O ports”). Desde el punto de vista del software, un puerto es una interfaz con ciertas características; se trata por tanto de una abstracción (no nos referimos al conector con el que se conecta físicamente un dispositivo al sistema), aunque desde el punto de vista del hardware, esta abstracción se corresponde con un dispositivo físico capaz de intercambiar información (E/S) con el bus.

Como los dispositivos E/S tienen algún tipo de conexión con el bus externo, deben tener una dirección (o conjunto de ellas) que los identifique. Los hemos comparado con un teléfono, un dispositivo con el que se puede enviar/recibir información mediante una dirección. Hablar de un puerto E/S es casi siempre sinónimo de una dirección, y es costumbre informática referirse a estas direcciones en hexadecimal

Podríamos resumir entonces que un Puerto es una zona de memoria RAM que se identifica mediante una dirección en donde los dispositivos dejan información o recogen información.

3.4.5.1 Direcciones de puertos

El modelo de direccionamiento del PC ha adoptado dos modelos, según la naturaleza del dispositivo a conectar:

⁶⁰ WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education p. 239



- **Direccionamiento dedicado** para dispositivos de media/baja velocidad. Por ejemplo, puertos serie y paralelo (impresora), teclado, Etc.
- **Direccionamiento mapeado** en memoria para dispositivos rápidos (típicamente los adaptadores gráficos).

Respecto a los dispositivos de direccionamiento dedicado, desde la introducción del IBM-PC, es costumbre que determinados dispositivos muy comunes se conecten mediante puertos de direcciones determinadas. En la tabla adjunta se muestran algunas de estas direcciones típicas para dispositivos de E/S. Las referencias a los chips se refieren a los utilizados en los primeros PCs. Actualmente han sido sustituidos por uno o dos integrados (Chipset) que realizan todas las funciones.

Puede observarse que para un mismo periférico se tienen varias direcciones (un “rango”). La razón es que estos dispositivos tienen varios registros y/o son programables, para lo que disponen de registros especiales (cada registro de un Byte tiene una dirección, y estas son correlativas). Por ejemplo, el controlador de puerto serie de los primeros PCs era un circuito integrado 8250 de National Semiconductor; que dispone de 8 registros, aunque los datos se reciben y transmiten por uno de ellos, el resto sirve para programar y controlar el funcionamiento del chip.

Respecto a las asignaciones “Estándar” de algunos puertos, es pertinente hacer algunas observaciones:

En realidad IBM definió de forma distinta las direcciones de puerto de las interfaces de impresora en los PC monocromo y en los de color. Las asignaciones fueron las siguientes:

Puerto	Monocromo	Color
LPT1	3BC-3BFh	378-37Fh
LPT2	378-37Fh	278-27Fh
LPT3	278-27Fh	---

En lo referente a los puertos serie, durante la secuencia inicial de identificación del hardware, la BIOS realiza una búsqueda de dispositivos de E/S por direcciones en un orden determinado, y les asigna un nombre lógico según el orden de aparición. En el caso de los puertos serie el orden suele ser el siguiente:

Puerto	Dirección
COM1	3F8-3FFh
COM2	2F8-2FFh
COM3	3E8-3EFh
COM4	2E8-2EFh

Puedes comprobar la asignación de puertos del sistema mediante la utilidad msinfo32.exe (las direcciones de memoria están expresadas en hexadecimal).

Las direcciones mapeadas en memoria se eligieron para algunos dispositivos como los adaptadores gráficos, que requerían mover grandes cantidades de datos muy rápidamente.

3.4.5.2 Comunicar con los puertos

Nos detendremos brevemente en analizar los aspectos involucrados en las opera-

ciones E/S con los puertos desde una doble vertiente: Lógica y física; esta última relativa a los mecanismos involucrados en el intercambio.

Cuadro N° 1. Direcciones de puertos

DISPOSITIVO	DIRECCIÓN
Controlador DMAC#1 primario 8237A	000-00F
Controlador de interrupciones PIC#1 8259A	020-021
Temporizador programable PIT 8253 y 8254	040-043
Controlador periféricos PPI 8255 (XT)	060-063
Controlador 8742 (AT)	060-064
Altavoz del sistema	061-061
CMOS RAM & NMI Mask Reg (AT)	070-071
Registros de página DMA	081-08F
Controlador de interrupciones PIC#2 (AT)	0A0-0A1
NMI Mask register (XT)	0A2-0AF
Controlador DMAC#2 secundario (AT)	0C0-0DF
Clear / Reset del coprocesador matemático	0F0-0F1
Coprocesador matemático 80287	0F8-0FF
Controlador de disco duro (AT)	1F0-1FF
Puerto de juegos	200-20F
Unidad de expansión (XT)	210-21F
Ratón (conectado al bus)	238-23B
Ratón de bus alternativo	23C-23F
Puerto paralelo (LPT2)	278-27F
Puerto serie 8250 (COM4)	2E8-2EF
Puerto serie 8250 (COM2)	2F8-2FF
Controlador disco duro (XT)	320-32F
Tarjeta de red	360-363
Tarjeta de red	368-36B
Puerto paralelo (LPT1)	378-37F
MDA (Monochrome Display Adapter)	3B0-3BB
Puerto paralelo (LPT3)	3BC-3BF
EGA	3C0-3CF
CGA	3D0-3DF
Puerto serie 8250 (COM3)	3E8-3EF
Controlador de disquete	3F0-3F7
Puerto serie 8250 (COM1)	3F8-3FF





LECTURA SELECCIONADA N° 2

“NUEVAS Y REVOLUCIONARIAS MEMORIAS”

Cuatro años después de un descubrimiento en el campo de la física, ya se está trabajando en un modo de aprovecharlo para crear memorias de ordenador con varias ventajas clave sobre las convencionales de hoy en día. Y la labor está bastante avanzada.

Unos investigadores en la Universidad Rice en Houston, Texas, están diseñando memorias de ordenador en láminas flexibles, transparentes y con arquitecturas en vertical, no sólo en horizontal como es lo normal en los chips convencionales.

Pero lo más llamativo no es ese conjunto de características sino el modo de almacenar la información, que está basado en el descubrimiento hecho en 2008 de que el óxido de silicio tiene propiedades de conmutación.

El equipo del químico James Tour y el físico Douglas Natelson está fabricando dispositivos transparentes (en un 95 por ciento) de memoria resistiva no volátil, basados en el citado hallazgo hecho en la universidad de que el óxido de silicio de por sí puede ser un conmutador. Un voltaje aplicado a una lámina delgada de óxido de silicio hace que sean expulsados los átomos de oxígeno en un canal de 5 nanómetros de ancho, lo cual hace que este canal pase a ser un conductor hecho de silicio. Aplicando adecuadamente los voltajes, se puede inducir en ese canal la citada conversión y luego revertirlo a su estado inicial. Este ciclo se puede repetir a voluntad miles de veces.

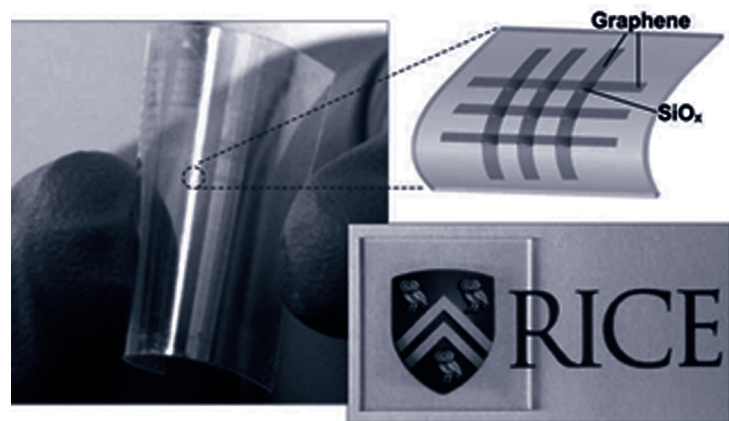


Figura Nro 63. Memorias de ordenador en láminas flexibles.

Ese canal puede, por tanto, ser leído como un “1” o un “0”, actuando así como el conmutador que es la unidad básica de las memorias de los ordenadores. Como es de sólo 5 nanómetros, podría mantener vigente la Ley de Moore, la cual predice que los circuitos de ordenador duplicarán su potencia cada dos años. Los componentes electrónicos de la actualidad son fabricados con circuitos de 22 nanómetros. En el trabajo de investigación y desarrollo también han intervenido Jian Lin así como Jun Yao (ahora en la Universidad de Harvard, en Cambridge, Massachusetts, Estados Unidos).

Tomado de:

http://noticiasdelaciencia.com/not/5535/nuevas_y_revolucionarias_memorias_electroquimicas_deordenador/

⁶¹ Foto: Jun Yao/Rice University. Recuperado de http://noticiasdelaciencia.com/not/5535/nuevas_y_revolucionarias_memorias_electroquimicas_deordenador/

**ACTIVIDAD N° 2**

Esta actividad puede consultarla en su aula virtual

**TAREA ACADÉMICA N°1**

Esta actividad puede consultarla en su aula virtual.

**GLOSARIO DE LA UNIDAD**

APU: Un Accelerated processing unit en español Unidad de Procesamiento Acelerado, Unidad de Cómputo Acelerado o simplemente APU, dichos microprocesadores combinan un CPU multinúcleo, una GPU y un bus de interconexión de alta velocidad que permite transferir información a mayores velocidades, debido a que se encuentran en el mismo chip el CPU la GPU y controlador de memoria entre otros, esto es posible gracias a la miniaturización de los chips actuales procesos de fabricación de entre 40 y 32 nm.

El término unidad de procesamiento acelerado o APU fue utilizado por primera vez en un contexto público en el año 2006.

ALU: Unidad Aritmético Lógica. Es la encargada de las operaciones aritméticas y lógicas: suma, resta, multiplicación, división, mayor, menor, diferente, igual, retornando como resultado de su operación falso o verdadero etc.

BIOS: Basic Input Output System. Es un programa informático inscrito en componentes electrónicos de memoria Flash (ejemplo clásico de FIRMWARE) existentes en la placa base. Este programa controla el funcionamiento de la placa base y de dichos componentes. Se encarga de realizar las funciones básicas de manejo y configuración del computador.

BIT: Unidad elemental de información que equivale a 0 o 1. Acrónimo de Binary Digit

BYTE: Conjunto de 8 bits. Se relaciona directamente con un carácter de la tabla de códigos ASCII.

CHIPSET: Es un conjunto de componentes electrónicos contruidos en un circuito integrado que gestiona el flujo de datos entre el procesador, la memoria y los dispositivos periféricos. Por lo general se encuentra en la placa base de un ordenador, están diseñados para trabajar con una familia específica de procesador.

CODIFICACIÓN: Transformación de la representación de la información.

CONEXIÓN ATX: Es la conexión que se hace de la fuente de poder a la Placa. Compuesta de 20 pines o 24 pines.

CPU: Unidad Central de Procesamiento. Su nombre proviene del hecho que se





encarga de procesar datos de acuerdo a las instrucciones que se le indican en memoria, finalmente presenta los resultados en alguna forma desalida.

DATOS: Símbolos para representar un número, objeto, letra etc.

DDR SDRAM : DDR permite a ciertos módulos de memoria RAM compuestos por memorias síncronas (SDRAM), disponibles en encapsulado DIMM, la capacidad de transferir simultáneamente datos por dos canales distintos en un mismo ciclo de reloj. Los módulos DDR soportan una capacidad máxima de 1 GiB (1 073 741 824 bytes).

DVI: Digital Visual Interface. Es una interfaz de vídeo diseñada para obtener la máxima calidad de visualización posible en pantallas digitales.

ESTACIONES DE TRABAJO: Computador de altas prestaciones destinado para trabajo técnico o científico. En una red de computadoras, es una computadora que facilita a los usuarios el acceso a los servidores y periféricos de la red.

HARDWARE: Elementos electrónicos y mecánicos del computador

IDE: Tecnología para la conexión de dispositivos de almacenamiento como el disco duro y la unidad de CD mediante un bus de datos de 40 hilos, relacionada con el estándar PATA(Parallel ATA).

INSTRUCCIONES: Pasos a realizar por la computadora para la ejecución de un programa.

LENGUAJE DE PROGRAMACIÓN: Instrucciones que conforman sentencias las cuales siguen unas reglas precisas.

LENGUAJES DE ALTO NIVEL: Instrucciones de entendimiento para los humanos fácilmente.

MACROCOMPUTADORAS (MAINFRAMES): Computadores de uso industrial, tienen gran procesamiento y al igual que las supercomputadoras puede tener cientos de usuarios conectados al mismo mainframe (multiusuario).

MEGAHERTZ: Medida de la frecuencia de Reloj, equivalente a 1024 KHz.

MEMORIA AUXILIAR: Dispositivos de almacenamiento permanente y externos como discos duros, USB, DVD, CD etc.

MEMORIA RAM: Random Access Memory. Memoria en la que cada posición direccionable tiene un único mecanismo de direccionamiento. El tiempo de acceso a una posición dada es independiente de la secuencia de acceso previa, es empleada por el procesador, desde la cual el procesador va tomando instrucciones y datos según la dirección que indique el Program Counter(PC).

MEMORIA ROM: Read Only Memory. Normalmente empleada como FIRMWARE, para almacenar programas que no requieran ser compilados.

NANOCOMPUTADORA: Computadora con una circuitería tan pequeña que sólo puede verse a través de un microscopio. Las nanocomputadoras pueden ser electrónicas (donde la nanolitografía se usa para crear los circuitos microscópicos), bioquímica u orgánica. Las nanocomputadoras se componen de materiales a nivel molecular y son la promesa de crear computadoras cada vez más pequeñas y rápidas, un concepto muy importante en el mundo de la computación

ORDENADOR: Término empleado en España para designar a las Computadoras.

PERIFÉRICOS: Son las unidades de entrada y salida.

SATA: Serial Advanced Technology Attachment. Emplea una conexión serial, por ello emplea una sola interfaz para cada dispositivo.

SISTEMA OPERATIVO: Programa principal para controlar la ejecución de todos los programas a su vez hace la labor de interfaz entre el usuario y el Hardware.

SLOTS: Ranuras donde se insertan tarjetas de expansión: Sonido, Video.

SOCKET: Elemento físico diseñado para insertar el procesador.

SOFTWARE: Son los programas.

UC: Unidad de control. Elemento de la CPU encargada de interpretar las instrucciones que se toman desde la RAM y de acuerdo a este resultado determina el flujo siguiente a efectuar.

UNIDADES DE ENTRADA: Dispositivos externos del computador que reciben la información del usuario: Mouse, Teclado, Scanner.

UNIDADES DE SALIDA: Dispositivos externos del computador que envían la información al usuario: Monitor, Parlantes, Impresora.



BIBLIOGRAFIA DE LA UNIDAD I

- Josu Bilbao Eguia (2008). Arquitectura de ordenadores. España: Editorial Independiente.
- WILLIAM STALLINGS (2006). Organización y arquitectura de computadores –España: Pearson Education.
- Ignacio Moreno Velasco (2012). Mantenimiento de equipos informáticos. España: Universidad de Burgos.

Enlaces de interés:

- www.tomshardware.com
- www.todalapc.com
- www.culturacion.com
- <http://tabasco.torreingenieria.unam.mx/GCH/ensamblador/Documentos/8086/8086.htm>
- <http://eduardrosario.blogspot.com/2009/10/empaquetado.html>
- http://es.wikipedia.org/wiki/LGA_775



AUTOEVALUACIÓN DE LA UNIDAD I





1. Familia de Procesadores de Intel que se coloca en un zócalo LGA 775
 - a. Pentium IV
 - b. Pentium M
 - c. Pentium III
2. Se denomina memoria de acceso aleatorio por su característica de:
 - a. La lectura es directa y no se debe esperar
 - b. Se desconoce con anticipación de donde se obtendrán los datos
 - c. Se generan números al azar para realizar la grabación de los datos.
3. La memoria se usa específicamente como:
 - a. Zona de almacenamiento permanente
 - b. Espacio temporal de trabajo
 - c. Zona de aparcamiento de los cabezales
4. Con respecto a las velocidades de acceso se puede decir que:
 - a. La ROM es más rápida que la RAM
 - b. La RAM tiene la misma velocidad de acceso
 - c. La ram es más veloz que la ROM
5. No es una función de la unidad de Control
 - a. Ejecutar instrucciones
 - b. Almacenar variables para el procesador
 - c. Tomar instrucciones desde la Memoria
6. Procesador de la familia Intel que se caracteriza por qué se puede colocar en un slot LGA 775 y además posee FSB de 666 Mhz o superior
 - a. Pentium I
 - b. Pentium IV
 - c. Core Duo
7. En la arquitectura x86 el chipset Sur tiene como misión controlar
 - a. La RAM
 - b. El procesador
 - c. Periféricos
8. Normalmente el FSB es con respecto a la frecuencia de reloj interna
 - a. Menor
 - b. Mayor
 - c. Igual
9. La memoria Cache mejora la ejecución de los programas por:
 - i. Emplea un medio de almacenamiento magnético.
 - ii. Su cercanía al procesador.
 - iii. El efecto “Localidad de los Programas”.
 - a. solo iii es cierta
 - b. sólo i es cierta
 - c. son ciertas ii y iii
10. Con un Bus de direcciones de 12 bits se pueden direccionar hasta:

³⁵ <http://es.wikipedia.org/wiki/Percepci%C3%B3n>

- a. 2 MB
- b. 4 KB
- c. 12 MB

11. Procesador de Intel, que se caracteriza por que sus arquitectura era una combinación del Pentium MMX y el Pentium Pro.

- a. Pentium I
- b. Pentium II
- c. Celeron







UNIDAD III: LENGUAJE ENSAMBLADOR



DIAGRAMA DE PRESENTACIÓN DE LA UNIDAD III



ORGANIZACIÓN DE LOS APRENDIZAJES

CONOCIMIENTOS	PROCEDIMIENTOS	ACTITUDES
Tema 1: Lenguaje Ensamblador y la Organización del Computador 1.1 El CPU de la familia 8086 1.2 Registro de 16 bits y 32 bits Lectura seleccionada 1 Tema 2 : Lenguaje Ensamblador Básico 2.1 Instrucciones básicas y estructura de un programa 2.2 Directivas de datos 2.3 Interrupciones 2.4 Interrupciones más comunes 2.5 Empleando el Emulador 8086 Tema 3 : Lenguaje Ensamblador y arquitecturas avanzadas 3.1 Instrucciones de control de flujo 3.2 Salto incondicional 3.3 Salto condicional 3.4 Subprogramas 3.5 Instrucciones Aritméticas Lectura seleccionada 2 Ventajas y desventajas del Lenguaje Ensamblador (6/02/2012), Cómputo integrado Autoevaluación de la unidad III	1. Entiende el funcionamiento del Lenguaje ensamblador en relación a los elementos de un computador 2. Elabora programas básicos empleando el Lenguaje ensamblador Actividad N° 1 Empleando el Emulador 8086 efectúa el seguimiento a un programa básico en assembler Actividad N° 2 Elabora un programa para mostrar un carácter en una posición específica de pantalla dada por el usuario Control de lectura N° 2 Responde un cuestionario de los temas 1 y 2	Muestra interés por entender y explicar el funcionamiento de un computador



TEMA N° 1: LENGUAJE ENSAMBLADOR Y LA ORGANIZACIÓN DEL COMPUTADOR

1.1. El CPU de la familia 8086.⁶²

Antes de comenzar a describir lo que es el lenguaje ensamblador en 80x86, vamos a recordar cómo se representan los números en una CPU. Una CPU es la Unidad Central de Proceso dentro de un microprocesador; tal que la información en una CPU se representa en binario, esto es, usando base 2. Un BIT es el elemento que representa el elemento básico unidad. A partir de ahí tenemos:

1 NIBBLE: 0000 (4 BITS). Es base del hexadecimal, e.j. 1111 = Fh.

1 BYTE: 00000000 (8 BITS) ó 2 NIBBLES

1 WORD: 0000000000000000 (16 BITS) ó 2 BYTES ó 4 NIBBLES. Es el tamaño usado para un registro de 16-bits.

1.1.1 Organización de Registros

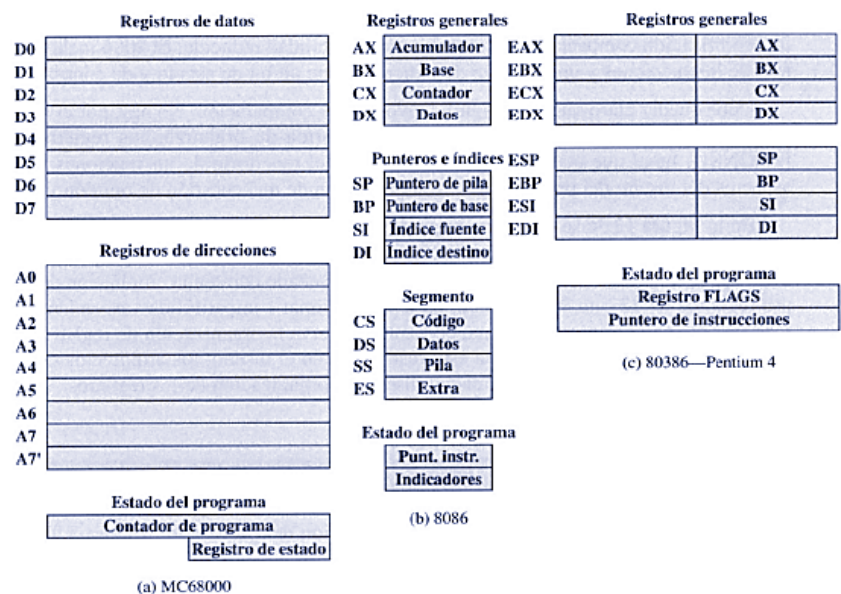


Figura N° 64. Ejemplos de Organizaciones de Registros⁶⁴

1.2 Registros de 16 y 32 bits

Los registros son los elementos principales de almacenamiento de la CPU. Pueden almacenar y mantener la información. En la arquitectura 8086 de INTEL, existen tres tamaños de registros: de 8-bit, de 16-bit y de 32-bit (para versiones superiores al 386). Además, se especifican 4 tipos de registros básicos y registros de control:

- Registros de propósito general,
- Registros de pila,
- Registros de índice y
- Registros de segmento

⁶² Sadot Alexandres Fernández(2004). *Introducción al ensamblador 80x86*. Madrid: Departamento de Electrónica y Automática. Escuela Técnica Superior de Ingeniería Universidad Pontificia Comillas de Madrid

⁶³ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education p. 445*

15	8	7	0	Bits	H=High L=Low
AH		AL		AX	Acumulador
BH		BL		BX	base
CH		CL		CX	contador
DH		DL		DX	datos
				SP	puntero de pila
				BP	puntero de base
				SI	índice a origen
				DI	índice a destino
REGISTROS DE CONTROL					
				IP	(puntero de instrucciones)
				FLAGS	(registro de estado)

Cuadro N° 2. Registros de propósito general, de pila e índice

1.2.1 Registros de propósito general

Son cuatro registros de 16-bits etiquetados como AX, BX, CX y DX. Éstos se dividen en 2 registros de 8-bits etiquetados como AH, y AL que contiene el BYTE BAJO; Ah contiene el Byte Alto. A partir del 386 hay registros de 32 bits, que conservan al mismo nombre y una E que le precede, esto es: EAX. Estos registros se pueden usar indistintamente como: AL, AH, AX y EAX. Por tanto si, AX contiene el número 24689 en decimal, tenemos que:

AH	AL	AX
01100000	01110001	6071 (hexadecimal)

1.2.2 Registros de índices

Estos registros de 16 bits también son llamados “registros punteros”. Su utilidad es para el uso de instrucciones que operan con cadenas de caracteres. Lo forman el registro de índice SI (origen) y el registro DI (destino). Al igual que los registros de propósito general, en las versiones del 386 y superiores de Intel, existen los registros ESI y EDI.⁶⁴

1.2.3 Registros de Pila

Los registros de control de pila, BP y SP, se usan cuando manejamos una pila (stack). Estos registros se ven más adelante en el párrafo que corresponde al STACK.

1.2.4 Registros de Segmentos y offset

En los orígenes del 80x86 se pensó que nadie iba a usar más de 1 MByte de memoria, tal que los primeros microprocesadores (el 8088) no permiten direccionar más allá de este número. Es decir, se utilizan 20 bits de direcciones, tal que si queremos más direccionamiento se podrían usar 2 registros de 16 bits. Esto nos da un número binario de 32 bits. Sin embargo, ¡esto es mucho!, pensaron los ingenieros de diseño. Aparecen entonces para resolver este dilema, los segmentos y el desplazamiento (offset). Este modo de operación es el denominado modo

⁶⁴ Sadot Alexandres Fernández(2004). *Introducción al ensamblador 80x86*. Madrid: Universidad Pontificia Comillas de Madrid. p. 6



real y se generó de la siguiente manera:

Dos registros de 16 bits, uno que contiene el segmento y otro el offset, generan una dirección física de 20 bits. Para ello, se colocan en posición tal que, el registro segmento se mueve 4 posiciones a la izquierda y el de offset a la derecha, también 4 posiciones, dejando 4 bits a 0 respectivamente. Si se unen (se suman) los 2 registros, se obtiene una dirección de 20 bits, que nos da la dirección física en memoria.⁶⁵

SEGMENTO	0010010000010000 —	Dirección base del segmento + 4 bits a 0
OFFSET	—0100100000100010	4 bits a 0 + Offset
DIRECCION FISICA	00101000100100100010	20 bits total para dirección física a memoria

Por ejemplo, el registro DS guarda el Segmento y el registro SI guarda el offset, ambos son de 16 bits, por tanto un SEGMENTO tiene un tamaño de 64KByte. Sin embargo ambos registros se usan en conjunto para generar una dirección de 20 bits. Se usa para ello la notación DS:SI, es decir SEGMENTO:OFFSET.

Los registros para segmentos son: CS, SS, DS, ES y los registros que se usan para offset son: DX, BX, DI, SI, BP, SP

15	0		
		CS	(segmento de código)
		SS	(segmento de pila)
		DS	(segmento de datos)
		ES	(segmento extra de datos)

Registros de segmentos

1.2.5 Otros registros

Existen además otros registros de control. Los más importantes son, el registro IP (puntero de instrucción) y el registro de estado (FLAGS), que contiene la información del estado de la CPU.

1.2.6 La pila (stack)

En general, en la arquitectura de Intel, seis registros de propósito general, son insuficientes para almacenar todos los valores requeridos para las operaciones en ensamblador. Para el caso de mantener y recuperar los valores de estos registros, existe un área de memoria llamada PILA (stack). El nombre de PILA no es por otra cosa sino por su forma de uso. Es decir, el último que entra es el primero que sale, comúnmente también llamado “Last In First Out (LIFO)”.

Si entran más datos en la pila de los que puede almacenar, ésta desborda. Obsérvese que la PILA tiene asociada una dirección base alta y crece en sentido decreciente (en dirección). Es labor del usuario controlar el número de datos que entran en la PILA para no desbordar.

⁶⁵ Sadot Alexandres Fernández(2004). *Introducción al ensamblador 80x86*. Madrid: Universidad Pontificia Comillas de Madrid. Pag 3.

⁶⁶ Sadot Alexandres Fernández. *Introducción al ensamblador 80x86*. Departamento de Electrónica y Automática. Escuela Técnica Superior de Ingeniería Universidad Pontificia Comillas de Madrid Febrero de 2004. Pag 5.

**LECTURA SELECCIONADA N° 1****LOS MICROCONTROLADORES Y MICROPROCESADORES****MICROCONTROLADOR**

En 1980 aproximadamente, los fabricantes de circuitos integrados iniciaron la difusión de un nuevo circuito para control, medición e instrumentación al que llamaron microcomputador en un sólo chip o de manera más exacta **MICRO-CONTROLADOR**.

Un microcontrolador es un circuito integrado que contiene toda la estructura (arquitectura) de un microcomputador, o sea CPU, RAM, ROM y circuitos de entrada y salida. Los resultados de tipo práctico, que pueden lograrse a partir de estos elementos, son sorprendentes.

Algunos microcontroladores más especializados poseen además convertidores análogos digital, temporizadores, contadores y un sistema para permitir la comunicación en serie y en paralelo.

Se pueden crear muchas aplicaciones con los microcontroladores. Estas aplicaciones de los microcontroladores son ilimitadas (el límite es la imaginación) entre ellas podemos mencionar: sistemas de alarmas, juego de luces, paneles publicitarios, etc. Controles automáticos para la Industria en general. Entre ellos control de motores DC/AC y motores de paso a paso, control de máquinas, control de temperatura, control de tiempo, adquisición de datos mediante sensores, etc.

Un controlador es un dispositivo electrónico encargado de, valga la redundancia, controlar uno o más procesos.

Por ejemplo, el controlador del aire acondicionado, recogerá la información de los sensores de temperatura, la procesará y actuará en consecuencia.

Al principio, los controladores estaban formados exclusivamente por componentes discretos. Más tarde, se emplearon procesadores rodeados de memorias, circuitos de E/S,... sobre una placa de circuito impreso (PCB).

Actualmente, los controladores integran todos los dispositivos antes mencionados en un pequeño chip. Esto es lo que hoy conocemos con el nombre de microcontrolador.

Diferencia entre microcontrolador y microprocesador

Es muy habitual confundir los términos de microcontrolador y microprocesador, cayendo así en un error de cierta magnitud. Un microcontrolador es, como ya se ha comentado previamente, un sistema completo, con unas prestaciones limitadas que no pueden modificarse y que puede llevar a cabo las tareas para las que ha sido programado de forma autónoma. Un microprocesador, en cambio, es simplemente un componente que conforma el microcontrolador, que lleva a cabo ciertas tareas que analizaremos más adelante y que, en conjunto con otros componentes, forman un microcontrolador.

Debe quedar clara por tanto la diferencia entre microcontrolador y microprocesador: a modo de resumen, el primero es un sistema autónomo e independiente, mientras que el segundo es una parte, cabe decir que esencial, que forma parte de un sistema mayor.

Arquitectura interna de un microcontrolador

Como ya hemos visto, un microcontrolador es un dispositivo complejo, formado por otros más sencillos. A continuación se analizan los más importantes.

Procesador

Es la parte encargada del procesamiento de las instrucciones. Debido a la necesidad de conseguir elevados rendimientos en este proceso, se ha desembocado en el empleo generalizado de procesadores de arquitectura Harvard frente a los tradicionales que seguían la arquitectura de von Neumann.





Esta última se caracterizaba porque la CPU se conectaba con una memoria única, donde coexistían datos e instrucciones, a través de un sistema de buses.

En la arquitectura Harvard son independientes la memoria de instrucciones y la memoria de datos y cada una dispone de su propio sistema de buses para el acceso. Esta dualidad, además de propiciar el paralelismo, permite la adecuación del tamaño de las palabras y los buses a los requerimientos específicos de las instrucciones y de los datos.

El procesador de los modernos microcontroladores responde a la arquitectura RISC (Computadores de Juego de Instrucciones Reducido), que se identifica por poseer un repertorio de instrucciones máquina pequeño y simple, de forma que la mayor parte de las instrucciones se ejecutan en un ciclo de instrucción.

Otra aportación frecuente que aumenta el rendimiento del computador es el fomento del paralelismo implícito, que consiste en la segmentación del procesador (pipe-line), descomponiéndolo en etapas para poder procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

Memoria de programa

El microcontrolador está diseñado para que en su memoria de programa se almacenen todas las instrucciones del programa de control. Como éste siempre es el mismo, debe estar grabado de forma permanente.

Existen algunos tipos de memoria adecuados para soportar estas funciones, de las cuales se citan las siguientes:

- **ROM** con máscara: se graba mediante el uso de máscaras. Sólo es recomendable para series muy grandes debido a su elevado coste.

- **EPROM**: se graba eléctricamente con un programador controlador por un PC. Disponen de una ventana en la parte superior para someterla a luz ultravioleta, lo que permite su borrado. Puede usarse en fase de diseño, aunque su coste unitario es elevado.

- **OTP**: su proceso de grabación es similar al anterior, pero éstas no pueden borrarse. Su bajo coste las hacen idóneas para productos finales.

- **EEPROM**: también se graba eléctricamente, pero su borrado es mucho más sencillo, ya que también es eléctrico. No se pueden conseguir grandes capacidades y su tiempo de escritura y su consumo es elevado.

- **FLASH**: se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar en circuito al igual que las EEPROM, pero que suelen disponer de mayor capacidad que estas últimas. Son recomendables aplicaciones en las que es necesario modificar el programa a lo largo de la vida del producto. Por sus mejores prestaciones, está sustituyendo a la memoria EEPROM para contener instrucciones. De esta forma Microchip comercializa dos microcontroladores prácticamente iguales que sólo se diferencian en que la memoria de programa de uno de ellos es tipo EEPROM y la del otro tipo Flash. Se trata del PIC16C84 y el PIC16F84, respectivamente.

Memoria de datos

Los datos que manejan los programas varían continuamente, y esto exige que la memoria que los contiene debe ser de lectura y escritura, por lo que la memoria RAM estática (SRAM) es la más adecuada, aunque sea volátil.

Hay microcontroladores que disponen como memoria de datos una de lectura y escritura no volátil, del tipo EEPROM. De esta forma, un corte en el suministro de la alimentación no ocasiona la pérdida de la información, que está disponible al reiniciarse el programa. El PIC16F84 dispone de 64 bytes de memoria EEPROM para contener datos.

Líneas de E/S

A excepción de dos patitas destinadas a recibir la alimentación, otras dos para el cristal de cuarzo, que regula la frecuencia de trabajo, y una más para provocar el Reset, las restantes patitas de un microcontrolador sirven para soportar su comunicación con los periféricos externos que controla.

Las líneas de E/S que se adaptan con los periféricos manejan información en paralelo y se agrupan en conjuntos de ocho, que reciben el nombre de Puertas. Hay modelos con líneas que soportan la comunicación en serie; otros disponen

de conjuntos de líneas que implementan puertas de comunicación para diversos protocolos, como el I2C, el USB, etc.

Recursos auxiliares

Según las aplicaciones a las que orienta el fabricante cada modelo de microcontrolador, incorpora una diversidad de complementos que refuerzan la potencia y la flexibilidad del dispositivo. Entre los recursos más comunes se citan los siguientes:

- Circuito de reloj: se encarga de generar los impulsos que sincronizan el funcionamiento de todo el sistema.
- Temporizadores, orientados a controlar tiempos.
- Perro Guardián o WatchDog: se emplea para provocar una reinicialización cuando el programa queda bloqueado.
- Conversores AD y DA, para poder recibir y enviar señales analógicas.
- Sistema de protección ante fallos de alimentación
- Estados de reposos, gracias a los cuales el sistema queda congelado y el consumo de energía se reduce al mínimo.

Programación de microcontroladores

La utilización de los lenguajes más cercanos a la máquina (de bajo nivel) representan un considerable ahorro de código en la confección de los programas, lo que es muy importante dada la estricta limitación de la capacidad de la memoria de instrucciones. Los programas bien realizados en lenguaje Ensamblador optimizan el tamaño de la memoria que ocupan y su ejecución es muy rápida.

Los lenguajes de alto nivel más empleados con microcontroladores son el C y el BASIC de los que existen varias empresas que comercializan versiones de compiladores e intérpretes para diversas familias de microcontroladores. En el caso de los PIC es muy competitivo e interesante el compilador de C PCM de la empresa CCS y el PBASIC de microLab Engineerign.

Hay versiones de intérpretes de BASIC que permiten la ejecución del programa línea a línea, y en ocasiones, residen en la memoria del propio microcontrolador. Con ellos se puede escribir una parte del código, ejecutarlo y comprobar el resultado antes de proseguir.

Tomado de: <http://ingenio-upp.blogdiario.com/>



ACTIVIDAD N° 1

Esta actividad puede consultarla en su aula virtual.

TEMA N° 2: LENGUAJE ENSAMBLADOR BÁSICO

2.1 Instrucciones básicas de Ensamblador y estructura de un programa

El número de instrucciones en ensamblador es muy grande, pero hay unas 20 instrucciones que son las usadas más frecuentemente. La mayoría de estas instrucciones se indican con tres letras en donde se especifica el tipo de instrucción y lleva asociada una serie de operandos separados por coma. Por ejemplo, para escribir en un registro usamos la instrucción MOV (de move en inglés MOVER).

mov ax,10 ; escribe 10 (decimal) en el registro ax

mov bx,20 ; escribe 20 (decimal) en el registro bx

mov cx,30 ; escribe 30 (decimal) en el registro cx

mov dx,40 ; escribe 40 (decimal) en el registro dx

Observa que en ensamblador el punto y coma (;) indica que se trata de comentarios.





2.1.1 Tipos de Operandos

Hay tres tipos de operandos en ensamblador:

- Tipo inmediato
- Tipo registro
- Tipo memoria

El operando inmediato es un número conocido en compilación, es decir una constante. El operando tipo registro es cualquier registro de propósito general o registro de índice. Un operando tipo memoria es una variable almacenada en memoria.

2.1.2 Instrucciones básicas

Hay una serie de instrucciones importantes y necesarias para cualquier programa en ensamblador. La más importante es la instrucción MOV.

MOV: mueve un valor de un lugar a otro.

MOV destino, origen

Ejemplos:

mov ax,10 ; mueve un valor inmediato al registro ax

mov bx,cx ; mueve en valor registro desde cx a bx

mov dx,Number ; mueve el valor memoria definido como Number a dx

2.1.3 Instrucciones Push y Pop. Uso del Stack

Conocido el Stack o pila, veremos cómo escribir datos y recuperarlos. Hay dos instrucciones para estas operaciones, que son:

PUSH: Pone el dato de un registro en la pila.

POP: Recupera el dato de la pila y lo escribe en un registro.

push cx ; escribe el contenido del registro cx en la pila

push ax ; escribe el contenido del registro ax en la pila

pop cx; recupera el valor de la pila en el registro cx

pop ax; recupera el valor de la pila en el registro ax

Nótese que los valores de CX y AX se intercambian en el código anterior. Existe una instrucción especial para intercambiar valores de dos registros: XCHG, la cual produce el mismo resultado con una única instrucción:

xchg ax,cx ; intercambia el valor de ax por el de cx

2.1.4 El Ensamblador

Para la compilación de un programa en ensamblador y su enlazado, hay que seguir una serie de pasos. Usaremos el ensamblador llamado TurboAssembler (TASM). Este ensamblador lo que hace es obtener un programa “ejecutable” para un ordenador que tenga integrado un microprocesador de la familia 80x86. La figura siguiente de talla los cuatro pasos que lleva a cabo el compilador.⁶⁸

⁶⁷ Sadot Alexandres Fernández(2004). *Introducción al ensamblador 80x86*. Madrid: Universidad Pontificia Comillas de Madrid. Pag 7

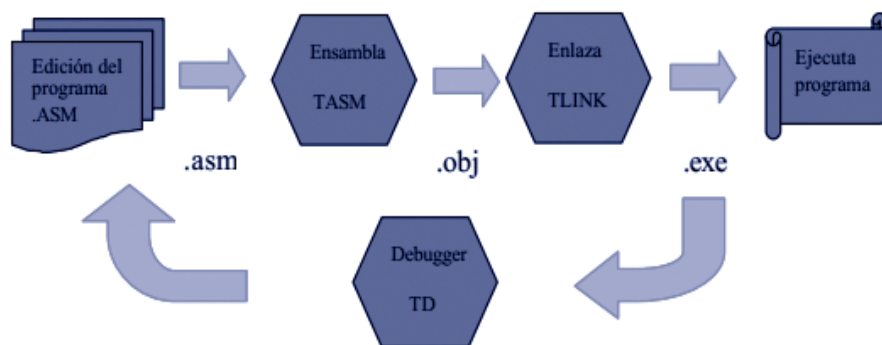


Figura Nro 65. Pasos para obtener un archivo ejecutable

- El fichero.asm contiene el programa en texto. Es el fichero origen para el compilador.
- El fichero.obj contiene un fichero intermedio. Este o varios ficheros son generados por el compilador.
- El fichero.exe es el fichero ejecutable desde el DOS. Es el fichero ejecutable generado por el enlazador o “linker”.

2.1.5 FORMATO DE UN PROGRAMA

El código fuente de un programa en lenguaje ensamblador se constituye por las siguientes partes.

- Área de comentarios
- Definición del modelo de memoria
- Área de datos
- Cuerpo del programa

El área de comentarios sirve para incluir comentarios acerca del programa que se está elaborando, comienza con la directiva `.COMMENT` y el comentario es colocado entre dos caracteres ‘*’

La definición del modelo de memoria es la parte donde se indica que tipo de código se va generar (16 o 32 bits). En este trabajo sólo se escribirán programas ejecutables `.COM`, por lo que siempre se usa la directiva `.MODEL TINY`

El área de datos es el lugar donde deben ser declaradas las constantes y variables del programa. Las variables son declaradas después de la directiva `.DATA` y las constantes después de `.CONST`

En el cuerpo del programa es donde se colocan las instrucciones en lenguaje ensamblador que se encargarán de realizar las tareas deseadas. El cuerpo del programa comienza con la directiva `.CODE` y termina con la directiva `END`. Esta parte corresponde al `Begin` y `End` de un programa en lenguaje Pascal

Adicionalmente se debe indicar un punto de entrada al programa. El punto de entrada se indica por medio de una etiqueta antes de la primer instrucción real del programa. En el ejemplo anterior el punto de entrada es `INICIO`; y el punto final de las instrucciones se indica por medio de la instrucción `END INICIO`

Cuando se requiere comentar las instrucciones de un programa, se debe colocar un punto y coma (;) y así el ensamblador interpreta todo lo que sigue como un comentario de una sola línea. Si requiere comentarios de más de una línea puede usar la directiva `COMMENT`.

⁶⁸ Sadot Alexandres Fernández(2004). *Introducción al ensamblador 80x86*. Madrid: Universidad Pontificia Comillas de Madrid. Pag 11

⁶⁹ Iden 85 Pag 11



2.1.6 Ejemplo de un Programa:

```
.COMMENT
*****

PRIMERO.ASM Ejemplo de un programa en lenguaje ensamblador.
*****

.MODEL TINY; Modelo de memoria para el programa
.DATA; Declaración de variables
    Mensaje db 'Mi primer programa',10,13,'$'
.CODE; Cuerpo del programa
INICIO; Punto de entrada al programa
    mov dx, OFFSET Mensaje ; Dirección de la cadena de texto
    mov ah,9 ; Función para imprimir cadenas
    int 21h ; Llamada al sistema operativo
    mov ah,4ch ; Función para terminar un programa
    int 21h ; Llamada al sistema operativo
END INICIO; Fin del bloque principal del programa
END
```

2.2 Directivas de datos ⁷⁰

2.2.1 Directivas simplificadas de segmentos

El ensamblador de Microsoft proporciona algunas formas abreviadas para definir segmentos. Para usar estas abreviaturas, inicialice el modelo de memoria antes de definir algún segmento. El formato general es:

```
.MODEL modelo de memoria
```

El modelo de memoria puede ser TINY, SMALL, MEDIUM, COMPACT o LARGE. Los requisitos para cada modelo son:

MODELO	NUMERO DE SEGMENTOS DE CODIGO	NUMERO DE SEGMENTOS DE DATOS
Tiny	*	*
Small	1	1
Medium	Más de 1	1
Compact	1	Más de 1
Large	Más de 1	Más de 1

2.2.2 DECLARACIÓN DE VARIABLES

La declaración de variables se hace en el segmento de datos. Para esto se debe incluir la directiva .DATA en el código fuente en el lugar donde se declaran las variables

Para declarar un variable se recomienda especificar el nombre de la variable, el tamaño de la variable y el valor inicial que esta tendrá. La tabla siguiente muestra algunas de las directivas utilizadas para especificar el tamaño de una variable

⁷⁰Peter Abel. (1996). *Lenguaje Ensamblador Programación para IBM PC y Compatibles*. México: Prentice Hall Iberoamericana.

DIRECTIVA	TAMAÑO	EJEMPLO
DB	Byte	Valorx db ? Mensa db hola
DW	Palabra	Contador dw 345Fh
DD	Doble palabra (32 bits)	Dirección dd 123FA4C1h

Si no se inicializa una variable se utiliza el símbolo? entonces la variable tendrá un valor indeterminado.

2.2.3 IDENTIFICADORES

Un identificador es un nombre que se aplica a elementos en el programa. El nombre, que se refiere a la dirección de un elemento de dato, y etiqueta, que se refiere a la dirección de una instrucción. Las mismas reglas se aplican tanto para los nombres como para las etiquetas.

Un identificador puede utilizar los siguientes caracteres:

- **Letras del alfabeto:** desde la A hasta la Z
- **Dígitos:** desde el 0 hasta 9 (no puede ser el primer carácter)
- **Caracteres especiales:** signo de interrogación (?) Subrayado (_) Signo de pesos (\$) Arroba (@) Punto (.) (no puede ser el primer carácter)

El primer carácter de un identificador debe ser una letra o un carácter especial, excepto el punto. El ensamblador trata a las letras mayúsculas y minúsculas como iguales. La longitud máxima de un identificador es de 31 caracteres (247 desde la versión 6.0)

2.2.4 INSTRUCCIÓN XCHG

La instrucción XCHG realiza otro tipo de transferencia de datos, pero en lugar de copiar los datos de una localidad a otra, XCHG intercambia los datos. El formato general para XCHG es: [etiqueta:]XCHG{registro/memoria} , {registro/inmediato}

Opciones validas con XCHG implican intercambio de datos entre dos registros y entre un registro y la memoria

2.2.5 INSTRUCCIÓN LEA

La instrucción LEA es útil para inicializar un registro con una dirección de desplazamiento. De hecho, un nombre más descriptivo para esta instrucción sería "Load Offset Address, carga una dirección de desplazamiento". El formato general para LEA es: [etiqueta:]LEA{registro/memoria}

El uso común para LEA es para inicializar un desplazamiento en el registro BX, DI o SI para indexar una dirección de memoria

2.2.6 INSTRUCCIONES INC Y DEC

INC y DEC son instrucciones adecuadas para aumentar y disminuir en 1 los contenidos de registros y localidades de memoria. El formato general para INC y DEC es:

[etiqueta:]INC / DEC{registro/memoria}

Note que estas instrucciones sólo necesitan de un operando. Dependiendo del resultado, la operación apaga o prende las banderas OF, SF y ZF, a las que las instrucciones de salto condicional puede verificar para menos, cero o más

2.3 Interrupciones

Todos los periféricos y otras funciones se pueden leer o escribir desde programa a través de una solicitud de servicio llamada interrupción. Una interrupción es una operación que invoca la ejecución de una rutina específica que suspende la ejecución del programa que la llamó, de tal manera que el sistema toma control del computador colocando en el stack el contenido de los registros CS e IP. El programa suspendido vuelve a activarse cuando termina la ejecución de la inte





rrupción y son restablecidos los registros salvados.

Existen dos razones para ejecutar una interrupción: (1) intencionalmente como petición para la entrada o salida de datos de un dispositivo, y (2) un error serio y no intencional, como un resultado de desbordamiento de la ALU o de división por cero.

2.3.1 Llamada a una interrupción

En un PC el control de estos servicios se hace a través del BIOS (Basic Input-Output System) o del sistema operativo MS-DOS (MicroSoft Disk Operating System), que le llamaremos DOS. Existen interrupciones software e interrupciones hardware, los ejemplos siguientes muestran cómo llamar al DOS y a la BIOS.

INT: llama a una función del DOS o BIOS a través de una subrutina para realizar un servicio especial, por ejemplo manipular el vídeo, la pantalla, abrir un fichero, escribir en un puerto, etc. En este texto usaremos una de ellas, la salida a pantalla, que se corresponde con la 21h. Las demás se pueden consultar en la bibliografía.

2.3.2 INT interrupt number

Ejemplos:

`int 21h` ; Llama a la interrupción número 21 (hexadecimal)

`int 10h` ; Llama a la interrupción del Video en la BIOS

Casi todas las interrupciones tienen asociados más de un servicio, por ello es necesario pasar el número de servicio deseado. Se utiliza para ello el registro AH. El ejemplo más usado es escribir algo en la pantalla del monitor:

`mov ah,9` ; el servicio número 9 (decimal) se pasa al registro AH

`int 21h` ; llamada a la interrupción 21

El servicio, o función, número 9 se corresponde a la impresión de caracteres en pantalla. Para ello, antes es necesario especificar qué vamos a escribir en pantalla. Se hace uso de DS:DX que será el puntero de la dirección física en memoria indicando dónde se encuentra la cadena de caracteres a imprimir en pantalla. La cadena de caracteres se termina con el símbolo (\$). Sería sencillo si pudiésemos escribir directamente en DS, como no es posible, tenemos que usar el registro AX para generar la dirección física. El ejemplo es el siguiente:

`mov dx,OFFSET Message` ; DX contiene el offset de message

`mov ax,SEG Message` ; AX contiene el segmento del message

`mov ds,ax` ; DS:DX apunta a la dirección física del mensaje

`mov ah,9` ; carga el servicio 9 – caracteres a pantalla

`int 21h` ; llamada a la interrupción de DOS

Las directivas OFFSET y SEG indican al compilador que el segmento y el offset de Message se use para obtener la dirección física en memoria la cadena de datos, para posteriormente escribir a pantalla. Hecho esto, podremos enviar el mensaje a la pantalla, escribiendo previamente en el segmento de datos el mensaje con la etiqueta y la directiva DB de la siguiente forma:

Message DB “Bienvenidos a Arquitectura de Computadores \$”

Nótese que la cadena de caracteres termina con el símbolo (\$). DB es una directiva para asignar memoria a variables del tipo byte y el contenido del mensaje en un array de bytes (ASCII). Además de formato bytes (DB) se puede usar formatos tipo word (DW) y doble word (DD).

Number1 db ?

Number2 dw ?

El símbolo (?) al final de la línea indica que no hay inicialización.

```
Number1 db 0  
Number2 dw 1
```

En este caso Number1 es igual a 0 y Number2 es igual a 1. Hay que tener en cuenta el tamaño de los datos, por ejemplo:

```
mov al,Number1 ; ok  
mov ax,Number1 ; error  
mov bx,Number2 ; ok  
mov bl,Number2 ; error
```

Lo que indica que, únicamente se pueden usar bytes en registros de 8 bits y palabras en registros de 16 bits

2.3.3 EJECUCION DE UNA INTERRUPCION

Para que una interrupción se ejecute correctamente es necesario almacenar el número de función en el registro acumulador AH (y demás registros de propósito general, si así se requiere), y posteriormente mandar a llamar el tipo de interrupción requerido.

A continuación se mostrara un fragmento de un programa para ejemplificar la ejecución de una interrupción.

CODIGO PARA BORRAR PANTALLA

```
mov cx, 0000  
mov dx, 2479h  
mov bh, 07  
mov ah, 06  
mov al, 00  
int 10h  
ret  
end
```

Como puede verse en este ejemplo, para ejecutar la interrupción 10H, primero fue necesario cargar el valor de la función en el registro acumulador y los demás atributos en los demás registros de propósito general.

El funcionamiento de una interrupción existe cuando el microprocesador concluye al ejecutar la instrucción en curso.

2.4 Interrupciones más comunes

2.4.1 INT 1AH

La interrupción de BIOS, 1AH, permite leer y escribir la fecha y hora actuales desde y al reloj del sistema. La tabla siguiente es un listado de las opciones disponibles con la interrupción del BIOS tipo 1AH.





Valor AH	Función	Entrada	Salida
AH=0	Lee inicialización reloj actual		CX--Bytes superiores del reloj DX--Bytes inferiores del reloj AL--0, si temporizador no ha pasado 24 horas
AH=1	Inicializa reloj actual	CX= DX=	Bytes superiores del reloj Bytes inferiores del reloj
AH=2	Lee hora--reloj de tiempo real		CH--BCD horas CL--BCD minutos DH--BCD segundos
AH=3	Inicializa hora--reloj de tiempo real	CH= CL= DH= DL=	BCD horas BCD minutos BCD segundos 1--hora del día 0--hora estándar
AH=4	Lee fecha-- reloj de tiempo real.		CH--BCD siglo CL--BCD año DH--BCD mes DL--BCD día
AH=5	Inicializa Fecha--reloj de tiempo real	CH= CL= DH= DL=	BCD siglo BCD año BCD mes BCD día
AH=6	Inicializa alarma (hasta 23:59:59)	CH= CL= DH=	BCD horas BCD minutos BCD segundos
AH=7	Reinicializa alarma.		

Cuadro N° 3. Interrupción 1Ah⁷¹

2.4.2 INT 10H

Muchas veces cuando se programa, la entrada o salida de la computadora necesita ser mostrada en la pantalla del monitor. Con frecuencia esto requerirá una simple operación de borrado de pantalla; en otras ocasiones se podrá desear cambiar los colores o atributos de la pantalla. El control de la pantalla puede conseguirse fácilmente con las interrupciones del BIOS tipo 10H. La tabla siguiente es un listado de las posibles variaciones programables con las interrupciones tipo 10H.

Valor AH	Función	Salida	Entrada
Control de interfaz del CRT			
AH=0	Inicializa el modo de visualizar	AL=0	40x25 B/W
		AL=1	40x25 COLOR
		AL=2	80x25 B/W
		AL=3	80x25 COLOR
		AL=4	320x200 color gráficas
		AL=5	322x200 B/W gráficas
		AL=6	640x200 B/W gráficas
AH=1	Inicializa tipo de cursor	AL=10	640x350 E.G.A. Gráficas
		CH=	Bits 4-0 principio de línea para el cursor
		CL=	Bits 4-0 fin de línea para cursor

⁷¹ Peter Abel. (1996). *Lenguaje Ensamblador Programación para IBM PC y Compatibles*. México. Prentice Hall Iberoamericana.

AH=2	Inicializa posición de cursor	DH= DL= BH=	Fila Columna Número página de pantalla
AH=3	Lee posición de cursor (valores hasta ejecución)	DH= DL= CH= CL= BH=	Fila Columna Modo cursor Modo cursor Número página de pantalla
AH=4	Obtiene posición lápiz luminoso (valores hasta ejecución)	AH=0 AH=1 DH= DL= CH= BX=	Conmutación no abajo/ disparo Respuestas validas; sigue: Fila Columna Grafo línea (0-199) Columna Grafica
AH=5	Inicializa página pantalla activa	AL=	Valor nueva página Modos 0 y 1 (0-7) Modos 2 y 3 (0-3)
AH=6	Repagina activa hacia arriba	AL= CH= CL= DH= DL= BH=	Número de líneas; 0 para pantalla entera Fila, esquina superior izquierda Columna, esquina superior, izquierda Fila, esquina inferior derecha Columna, esquina inferior derecha Atributo a ser usado
AH=7	Enrolla página activa hacia abajo	AL= CH= CL= DH= DL= BH=	Número de líneas; 0 para pantalla entera Fila, esquina superior izquierda Columna, esquina superior izquierda Fila, esquina inferior derecha Columna, esquina inferior derecha Atributo a ser usado
	Manipulación de caracteres		
AH=8	Lee atributo/ carácter en posición cursor	BH= AL= AH=	Página pantalla Carácter leído Atributo de carácter
AH=9	Escribe atributo/ carácter en posición cursor	BH= CX= AL= BL=	Página pantalla Cuenta de caracteres a escribir Carácter a escribir Atributo de carácter



⁷² 21/10/2012. Recuperado de: <http://www.abreojosensamblador.net/Productos/AOE/html/Pags/Cap11.html>



AH=10	Escribe carácter en posición cursor	BH = CX = AL =	Visualiza página Cuenta de caracteres a escribir Carácter a escribir
	Interfaz de gráficos		
AH=11	Selecciona paleta colores	BH = BL =	Paleta ID (0-127) Color para paleta ID 0-Fondo (0-15) 1-paleta 0-Verde (1), rojo (2), amarillo (3) 1- Cyan (1), magenta (2),
AH=12	Dibuja punto en pantalla	DX= CX= AL=	Fila (0-199) Columna (0-319/639) Color del punto
AH=13	Lee información punto	DX= CX= AL=	Fila (0-199) Columna (0-319/639) Valor del punto
	Salida teletipo ASCII		
AH=14	Escribe a página activa	AL= BL=	Carácter a escribir Color primer plano
AH=15	Devuelto a estado video	AL= AH= BH=	Modo actual Número de columnas de pantalla Página pantalla actual
AH=16	Reservado		
AH=17	Reservado		
AH=18	Reservado		
AH=19	Escribe cadena	ES:BP= CX= DX= BH= AL=0 AL=1 AL=2 AL=3	Señala a cadena Longitud de cadena Posición de cursor para comienzo Número de página BL= atributo(car,car,car...car) Cursor no transferido BL= atributo (car,car,car...car) Cursor es transferido (car,atr,car...atr) Curso no transferido (car,atr,car...atr) Cursor es transferido

Cuadro N° 4. INTERRUPCIÓN 10H⁷²

2.4.3 INT 21H

La interrupción 21H del DOS es una interrupción de llamada a función. Las especificaciones y requerimientos para utilizar esta interrupción se muestran en la siguiente tabla:

Valor AH	Función	Salida	Entrada
AH=1	Espera y visualiza carácter de teclado con comprobación CTRL-BREAK		AL - Carácter introducido
AH=2	Visualiza carácter de teclado con comprobación CTRL-BREAK	DL =	Carácter a visualizar
AH = 3	Entrada asíncrona de carácter		AL - Carácter introducido
AH = 4	Salida asíncrona de carácter	DL =	Carácter a enviar
AH = 5	Carácter a escribir	DL =	Carácter a escribir
AH = 6	Entrada carácter de teclado	DL = 0FFH	Carácter introducido
AH = 7	Espera para carácter de teclado (no visualiza)		AL - Carácter introducido
AH = 8	Espera para carácter de teclado (no visualiza-comprueba CTRL-BREAK)		AL - Carácter introducido
AH = 9	Visualiza cadena	DS:DX=	Dirección de cadena. Debe finalizar con centinela \$
AH = A	Cadena teclado a buffer	DS:DX=	Dirección de buffer. Primer byte=tamaño, segundo byte=número de caracteres leídos
AH = B	Status de entrada de teclado		AL-No. carácter=0FFH Carácter = 0
AH = C	Borra buffer teclado y llama función	AL =	1,6,7,8,0A - Número función
AH = D	Unidad implícita disco (reset)	Ninguna	Ninguna
AH = E	Unidad implícita disco (seleccionada)	DL =	AL - Núm. unidades 0 - Unidad A 1 - Unidad B, etc.
AH = 19	Código unidad (implícito)		AL - Núm. unidades 0 - Unidad A 1 - Unidad B, etc
AH = 25	Interrupción (inicializa)	DS:DX =	Dirección del vector de interrupción
AH = 2A	Fecha (lee)	AL=	Número de interrupción CX-Año(80 a 99) DH-Mes DL-Día(1 a 31)
AH = 2B	Fecha (inicializa)	CX:DX	Igual que antes AL-0 si valido OFF si no valido
AH = 2C	Hora (lee)		CH-Horas (0-23) CL-Minutos(0-59)
AH = 2D	Hora (inicializa)	CX:DX	Igual que antes AL-0 si valido OFF-si no valido





AH=2E	Verifica estado (inicializa)	DL=	0 AL= 0 0;verifica off 1;verifica on
AH=35	Dirección interrupción (lee)	AL=	Número de interrupción ES:BX- apunta a dirección vector
AH=36	Selecciona paleta colores	DL =	Unidad (0--implicita,1- A,2-B, etc.) AX-sectores/ apiñamiento(FFFF si invalido) BX-Número apiñamiento li- bres CX-Bytes por sector DX- Número total de apiñamientos
AH=39	Hace directorio	DS:DX	Dirección de cadena para directorio
AH=3A	Eliminar directorio	DS:DX	Dirección de cadena para directorio
AH=3B	Cambia directorio	DS:DX	Dirección de cadena para nuevo directorio
AH=3C	Archivo (crea)	DS:DX CX=	Dirección de cadena para archivo AX--Retorno manipula archivo Atributo archivo
AH=3D	Archivo (abre)	DS:DX AL=	Dirección de cadena para archivo 0--Abre para leer 1--Abre para escribir 2--Abre para ambos AX--Retorno manipula archivo
AH=3E	Manipula archivo(cierra)	BX=	Manipula archivo
AH=3F	Archivo o dispositivo(lee)	BX= CX= DS:DX	Manipula archivo Número de bytes a leer Dirección de buffer AX--Número de bytes a leer
AH=40	Archivo o dispositivo(escribe)	BX= CX= DS:DX	Manipula archivo Número de bytes a escribir Dirección de datos a escribir AX--Número de bytes a escribir
AH=41	Archivo (suprime)	DS:DX	Dirección de archivos cadena
AH=42	Archivo (pone atributo)	CX:DX AL=0 AL=1 AL=2	Desplazamiento en bytes Puntero desplazado (CX:DX) bytes desde el comienzo del archivo Puntero a posición actual más desplazamiento Puntero a EOF más desplazamiento
AH=43	Archivo (Inicializa atributo)	DS:DX	Dirección de archivo cadena SI AL=0, atributo devuelto a CX SI AL=1, inicializa archivo con atributo de CX
AH=47	Dirección actual	DL= DS:DI=	Número Unidad (0--implicita,1- unidad A,2--unidad B) Dirección buffer DS:DI--Devuelve dirección de cadena
AH=54	Verifica estado	Ninguna	AL--0 si verifica off 1 si verifica on
AH=56	Archivo(renombra)	DS:DX ES:DI	Dirección de cadena para informa- ción antigua Dirección de cadena para informa- ción nueva

Cuadro N° 5. INTERRUPCION 21H⁷⁵

2.5 Empleando el Emulador 8086

Una vez ya tenido instalado el programa para comenzar a trabajar nos dirigimos a: Inicio -> Todos los programas->emu8086 ->emu8086; se nos abrirá dos ventanas como se muestra en el gráfico

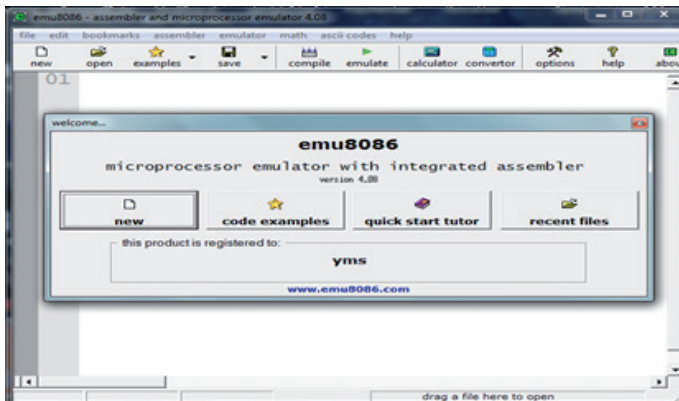


Figura Nro 66. Pantalla de bienvenida del emulador EMU8086

2.5.1 Pantalla de bienvenida del emulador EMU8086.

Se presentan cuatro diferentes opciones para elegir:

- **New:** Le permite escribir un nuevo código en lenguaje ensamblador (al que llamaremos "Código Fuente" y tendrá extensión .ASM)
- **Code examples:** Le permite acceder a una serie de programas ejemplos que pueden ayudarle a comprender funciones más complejas.
- **Quick star tutor:** activa un conjunto de documentos de ayuda, se recomienda revisarlos frecuentemente en caso de dudas.
- **Recent file:** Le muestra los últimos archivos que se trabajaron en la máquina.

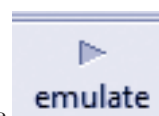
Para continuar este primer contacto con el emulador, seleccione New

Ahora ya tenemos preparado nuestro ambiente de trabajo.

En el espacio de trabajo que se muestra ingresa el siguiente código

```
01 .MODEL Small
02 .Stack 100h
03
04 .DATA
05 Mensaje DB 'Arquitectura del Computador',13,10,'$'
06 ;declarando la variable de cadena mensaje
07
08 .CODE
09 Inicio: mov ax,@data
10 mov ds,ax ; DS ahora apunta al segmento de datos.
11 mov ah,9 ; Función del DOS para impresión de cadenas.
12 mov dx,OFFSET Mensaje ; Apuntador al mensaje 'Hola Mundo'.
13 int 21h
14 mov ah,4ch ; Transferencia del control al DOS.
15 int 21h
16 END Inicio
```

Figura Nro 67. Ejemplo de codificación en EMU8086



Para ejecutar el Programa efectúa clic sobre el botón Emulate

Inmediatamente verás una ventana indicando que se está ejecutando la compilación.

Luego se te mostrará las siguientes pantallas:

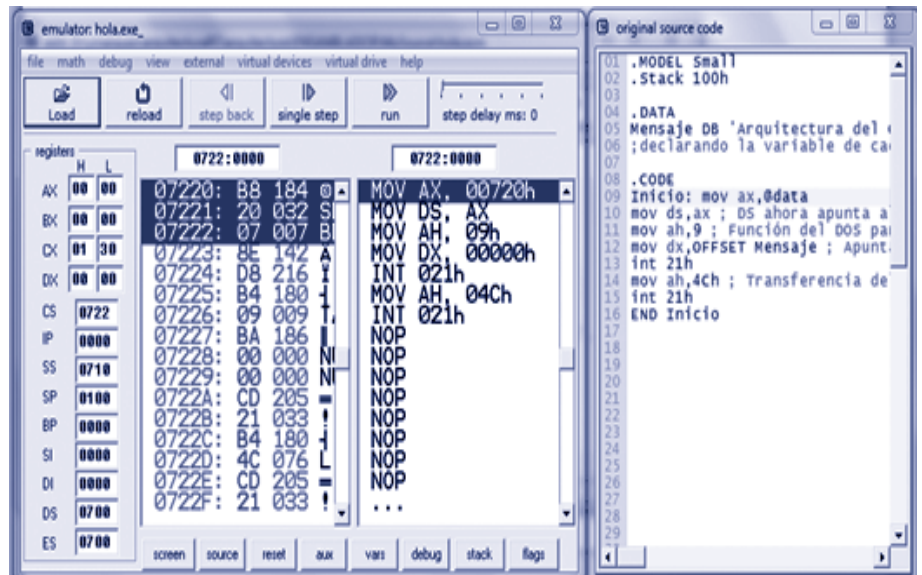


Figura Nro 68.Elementos del EMU8086

Para poder ver la ejecución del Programa Clic al Botón run.



Figura Nro 69.Programa en ejecución en EMU8086



LECTURA SELECCIONADA N° 2

Ventajas y desventajas del Lenguaje Ensamblador

Las computadoras son máquinas diseñadas para ejecutar las instrucciones que se le indican, de manera de resolver problemas o hacer algún otro tipo de cosas. Dichas instrucciones básicas no suelen ser más complicadas que realizar una suma o resta, una comparación, leer un dato o escribir un resultado en la memoria. Además, lo normal es que cada instrucción esté almacenada mediante un código en la memoria, de forma que la computadora “ve” esos números y los entiende como instrucciones. También, estos códigos son específicos de cada tipo de CPU. A este conjunto de instrucciones codificadas se le llama lenguaje de máquina y es el único lenguaje que una computadora entiende. Ahora bien, como programador, no resulta reconfortante pensar y escribir programas como una secuencia de números, de la manera en que la computadora los quiere ver, sino que es natural verlos como una secuencia de instrucciones. Para ello, se define un lenguaje, en principio, que es un mapeo directo de esos códigos a instrucciones comprensibles por un humano. A este lenguaje se lo denomina *assembly* o lenguaje de ensamble y al programa encargado de tomar programas escritos en *assembly* y generar los códigos que la computadora entenderá se lo denomina *assembler* o ensamblador. Como veremos, no todas las instrucciones de *assembly* se corresponden directamente con instrucciones de lenguaje de máquina, sino que varias son instrucciones para el ensamblador mismo.

Ventajas y desventajas del Lenguaje Ensamblador

El proceso de evolución trajo consigo algunas desventajas, que ahora veremos cómo las ventajas de usar el Lenguaje Ensamblador, respecto a un lenguaje de alto nivel:

1. Velocidad
2. Eficiencia de tamaño
3. Flexibilidad

Un programa escrito en el lenguaje ensamblador requiere considerablemente menos memoria y tiempo de ejecución que un programa escrito en los conocidos lenguajes de alto nivel, como C, C++, java, etc. El lenguaje ensamblador da a un programador la capacidad de realizar tareas muy técnicas que serían difíciles, si no es que imposibles de realizar en un lenguaje de alto nivel.

El conocimiento del lenguaje ensamblador permite una comprensión de la arquitectura de la máquina que ningún lenguaje de alto nivel puede ofrecer. Aunque la mayoría de los especialistas en Software desarrolla aplicaciones en lenguajes de alto nivel, que son más fáciles de escribir y de dar mantenimiento, una práctica común es recodificar en lenguaje ensamblador aquellas rutinas que han causado cuellos de botella en el procesamiento.

Los programas residentes y rutinas de servicio de interrupción casi siempre son desarrollados en el lenguaje ensamblador.

Por otro lado, al ser un lenguaje más primitivo, el Ensamblador tiene ciertas desventajas respecto a los lenguajes de alto nivel:

1. Tiempo de programación.
2. Programa fuente grande.
3. Peligro de afectar recursos inesperadamente.
4. Falta de portabilidad.



⁷⁴ <http://computo-integrado.blogspot.com/2012/02/lenjuage-assembly.html>



Como para y donde que implementar un tipo de ensamblador

El lenguaje ensamblador se puede implementar directamente desde la descripción de los estados en un block de notas, para realizar operaciones de bajo nivel que se son necesarias en un programa para facilitar todo esto

Para que se utilice el lenguaje ensamblador pero lo que investigue es que él se utiliza para realizar operaciones de memoria y facilitan los cosas porque un lenguaje de alto nivel tiene que decodifica las operaciones

Para qué se usa Porque hay aplicaciones o programas que deben tratar directamente con los registros de la máquina, la memoria, dispositivos de E/S, etc. Un programa ensamblador “bien hecho” produce un ejecutable más rápido y corto.

El proceso de traducción se realiza en dos pasos:

Primero: se recorre el programa fuente; por cada instrucción implementa el contador según el código de la instrucción. Comprueba si tiene o no etiqueta, y si la tiene coloca su símbolo y su dirección en la tabla de símbolos. Después compara el símbolo del código de operación con una tabla de símbolos posibles; si es válido sustituye el código real y si no emite un mensaje real; a continuación comprueba la sintaxis

Segundo: recorre las instrucciones del módulo fuente reemplazando los símbolos por sus direcciones reales tomadas de la tabla.



ACTIVIDAD N° 2

Esta actividad puede consultarla en su aula virtual.



CONTROL DE LECTURA N° 1

Esta actividad puede consultarla en su aula virtual.



GLOSARIO DE LA UNIDAD III

- **BCD:** Número decimal codificado en binario (Binary Coded Decimal). Se usan estas siglas para referirse a una forma especial de codificar números decimales en binario, usando 4 bits para cada dígito decimal, de modo que en un byte se pueden almacenar dos números decimales
- **Código Máquina:** lenguaje en el computador es capaz de reconocer directamente. Los programas que se creen en otros lenguajes deberán traducirse previamente a Código Máquina (compilarse) o el computador no podrá entenderlos.
- **Coma (fija y flotante):** La expresión “coma fija” (en inglés fixed point) se usa para referirse a números con una cantidad de cifras decimales constante. Estos números son más sencillos de manipular (pero menos útiles en la práctica) que 7 los números en “coma flotante” (en inglés, Floating Point), con un número variable de cifras decimales
- **Ensamblador:** Lenguaje de programación bajo nivel, muy cercano al código máquina. Su sintaxis depende por completo del tipo de computador que se esté usando

- **EXE:** Es la extensión que corresponde a un tipo de fichero ejecutable bajo MsDos, OS/2 y/o Windows
- **Interrupción:** Cuando un procesador está esperando recibir información de un cierto dispositivo externo, tiene dos formas básicas de hacerlo: mirando continuamente si hay información disponible (modo de espera o modo “polling”), o dejando la posibilidad de que sea el dispositivo el que avise cuando la tenga preparada (modo interrupción). Suele haber varios (pocos) canales de interrupción, así como distintos niveles de prioridades.
- **OS/2:** Sistema operativo multitarea de IBM creado para PCs, hoy en día en desuso
- **OS/400:** Sistema operativo multitarea y multiusuario creado por IBM para sus sistemas AS/400
- **Tiny:** Modelo de Memoria, con Código, Datos y Stack contenidos en un segmento de 64K. Llamada a subrutinas y transferencias de datos son cercanas (near). Utilizado solo para programas *.COM
- **Small:** Modelo de Memoria, con Código y Datos en segmentos separados de 64K. Llamada a subrutinas y transferencias de datos son cercanas (near). Utilizada en programas .EXE de tamaño small a medium. La mejor opción para programas exclusivamente en lenguaje ensamblador
- **Médium:** Modelo de Memoria, con Tamaño de código ilimitado. Datos limitado a un segmento de 64K. Las llamadas a subrutinas son de tipo far (lejanas); referencias a datos son de tipo near (cercanas).
- **Compact:** Modelo de Memoria, con Tamaño de código limitado a un segmento de 64k. Datos ilimitado. Las llamadas a subrutinas son de tipo near (cercanas); referencias a datos son de tipo far (cercanas). Utilizada por programas de tamaño medium a large con muchas variables o muy largas.
- **Large:** Modelo de Memoria, con Tamaños de Código y Datos ilimitados. Las llamadas a subrutinas y referencias de datos son de tipo far. Utilizadas para requerimientos de almacenaje de programas y datos grandes. Tan grandes que las variables no excedan 64K.
- **Huge:** Modelo de Memoria, con Tamaños de Código y Datos ilimitados. Las llamadas a subrutinas y referencias de datos son de tipo far. Utilizadas por programas muy grandes donde una o más variables exceden los 64K.



BIBLIOGRAFÍA DE LA UNIDAD III

Miquel Albert Orega, *Gerard Enrique Manonellas*(2011). Estructura de computadores. España: Material Docente de la UOC..

WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores* –España: Pearson Education.

Ignacio Moreno Velasco(2012). *Mantenimiento de equipos informáticos*. España: Universidad de Burgos.

Sadot Alexandres Fernández(2004). *Introducción al ensamblador 80x86*. Madrid: Universidad Pontificia Comillas de Madrid.

ENLACES DE INTERES

- <http://ingenio-upp.blogdiario.com>
- “Borland Turbo Assembler Quick Referente” .Borland. <http://www.borland.com>
- “Intel Pentium Family User’s Manual”: Volumen 3. <http://www.intel.com>
- <http://computo-integrado.blogspot.com/2012/02/lenjuage-assembly.html>





AUTOEVALUACIÓN DE LA UNIDAD III

Elige la alternativa que corresponda con la afirmación:

- Interrupción empleada para acceder a las funciones de la pantalla:
 - Int 21h
 - Int 10h
 - Int 16h
 - Int 12h
- En la siguiente instrucción: `mov ax, @data`
`mov ds, ax`, la segunda línea se establece para:
 - Almacenar el valor del registro ax en el procesador
 - Almacenar cada una de las variables del programa en el registro de desplazamiento.
 - Almacenar la dirección límite del segmento de datos en el registro de desplazamiento.
 - Almacenar la dirección base del segmento de datos en el registro de desplazamiento.
- Cuál es el motivo para incluir el carácter \$ en la asignación de variables que contienen texto?
 - Para poder determinar el final de la cadena de texto.
 - El símbolo \$ indica que el que se trata de una cadena de texto.
 - Se emplea para insertar un salto de nueva línea dentro de una cadena de texto.
 - En realidad no es necesario incluir el símbolo \$, es únicamente una formalidad.
- En un programa de assembler se observa en la primera línea la siguiente instrucción: `model small`, cual es el significado?
 - Es un Modelo de Memoria, con Código y Datos en segmentos separados de 64K.
 - Es un Modelo de Memoria, con Código y Datos de tamaño ilimitado
 - Es un Modelo de Memoria, con Código y Datos en segmentos separados de 32K.
 - Es un Modelo de Memoria, con Código de 64K y datos de tamaño ilimitado.
- Se acaba de guardar el archivo `prog1.asm` de un programa en assembler x86, los pasos que se deben seguir para llegar a ejecutarlo son:
 - Invocar a `prog1.exe` y luego ingresar `tasm prog1.asm`, finalmente `link prog1`.
 - Invocar a `tasm prog1`, luego a `link prog1.asm`, finalmente `prog1.exe`.
 - Invocar a `tasm prog1.asm`, luego a `link prog1`, finalmente `prog1.exe`.
 - Invocar a `link prog1`, luego a `tasm prog1.asm` finalmente `prog1.exe`.
- Proceso por el cual un programa invoca las librerías que empleará:
 - Linkeado.
 - Traducción.
 - Carga.
 - Compilado.
 - Debug.

7. Ordena las siguientes instrucciones para que se muestre un mensaje y luego se capture el ingreso de un carácter por teclado

- i. `int 21h`
- ii. `mov ah,9`
- iii. `mov ah,1`
- iv. `mov dx,OFFSET mensa1`

- a. ii,iii,i,iv,i
- b. iii,i,ii,iv,i
- c. iv,ii,i,iii,i
- d. iv,ii,i,iii,ii

8. El lenguaje assembler puede emplear muchas interrupciones, por ejemplo la _____ que es una interrupción del sistema operativo mientras que la _____ que es una interrupción BIOS.

- A. Int 21h, int 10h
- B) Int 16h, int 17h
- C) Int 10h, int 21h
- D) Int 17h, int 10h

9. Al efectuar la captura de un carácter por ejemplo el dígito 5 mediante la instrucción `int 21h` sucede lo siguiente:

- a. El carácter se lleva al registro `al`, donde lo que se almacena es su valor posicional en la tabla ASCII.
- b. El carácter se lleva al registro `al`, donde lo que se almacena es su valor numérico
- c. El carácter se lleva al registro `ah`, donde lo que se almacena es su valor ASCII.
- d. El carácter se lleva al registro `ax`, donde lo que se almacena es su valor numérico

10. Al encontrar la instrucción `model small` en un programa de assembler x86 se esta indicando:

- i. Que se empleara una estructura predeterminada
- ii. El programa no excederá de 128 KB
- iii. La zona de datos se define por `.data` y la de código por `.code`

- a. Es cierta solo i
- b. Son ciertas i y iii
- c. Es cierta solo iii
- d. Son ciertas ii y iii







UNIDAD IV: LENGUAJE ENSAMBLADOR Y ARQUITECTURAS AVANZADAS



DIAGRAMA DE PRESENTACIÓN DE LA UNIDAD IV



ORGANIZACIÓN DE LOS APRENDIZAJES

CONOCIMIENTOS	PROCEDIMIENTOS	ACTITUDES
Tema N° 1: Estructuras de control y subprogramas 1.1 Instrucciones de control de flujo 1.2 Salto incondicional 1.3 Salto condicional 1.4 Subprogramas 1.5 Instrucciones Aritméticas. 1.6 Instrucciones Lógicas	1.Elabora programas en lenguaje ensamblador que le permitan resolver problemas simples empleando instrucciones repetitivas 2. Lista las aplicaciones reales de las arquitecturas escalares y paralelas Actividad N° 1 Actividad N° 2 TAREA ACADÉMICA N° 2 Desarrolla un programa que emplee leds y modifique alternadamente los bits en el puerto paralelo	Muestra interés por entender y explicar el funcionamiento de un computador
Lectura Seleccionada 1 Diez lenguajes experimentales que podrían impactar en un futuro http://www.genbetadev.com/autor/jorge-rubira		
Tema N° 2: Arquitecturas avanzadas 2.1 Procesamiento Paralelo. 2.2 Arquitectura Superescalar		
Lectura Seleccionada 2 ¿Por qué máquinas vectoriales? JOHN PAUL SHEN y MIKKO H. LIPASTI (2005-2006) Arquitectura de Computadores: Fundamentos de los procesadores superescalares . México		
Autoevaluación de la unidad IV		



TEMA N° 1: ESTRUCTURAS DE CONTROL Y SUBPROGRAMAS

El assembler 8086 implementa las estructuras de control condicional y repetitivo mediante la unión de dos o más instrucciones de control de flujo como veremos a continuación.

1.1 Instrucciones de control de Flujo

Las instrucciones de control de flujo, o instrucciones de salto, son las que permiten a los programas cambiar la dirección del siguiente código máquina a ser ejecutado.

Sin instrucciones de control de flujo, un programa simplemente podría iniciar desde el comienzo del programa y correr sin parar hasta el término del código, sin paros, ciclos, o caminos alternos a través de su vía. Con control de flujo, los programas pueden realizar decisiones, inspeccionan banderas, y toman acciones basadas en operaciones previas, prueba de bits, comparaciones lógicas, y aritméticas. También, las instrucciones de control de flujo proporcionan a los programas la habilidad de repetir instrucciones basadas en ciertas condiciones, conservando memoria para ciclar continuamente a través de las mismas secciones de código.

INSTRUCCIONES LÓGICAS MNEMÓNICO/OPERANDO	DESCRIPCIÓN
call destino	Llama un procedimiento
jmp destino	Salta incondicionalmente
ret valor	Retorna de un procedimiento
retn valor	Retorna de un procedimiento cercano
retf valor	Retorna de un procedimiento lejano
Instrucciones de transferencia condicional	
ja/jnbe destino-corto	Salta si mayor/no menor o igual
jae/jnb destino-corto	Salta si mayor o igual/no menor
jb/jnbe destino-corto	Salta si menor/no mayor o igual
jbe/jnba destino-corto	Salta si menor o igual/no mayor
jc destino-corto	Salta si acarreo
je/jz destino-corto	Salta si igual/0
jg/jnle destino-corto	Salta si mayor/no menor o igual
jge/jnl destino-corto	Salta si mayor o igual/no menor
jl/jnge destino-corto	Salta si menor/no mayor o igual
jle/jng destino-corto	Salta si menor o igual/no mayor
jnc destino-corto	Salta si no acarreo
jne destino-corto	Salta si no igual/0
jno destino-corto	Salta si no hay sobreflujo
jnp/jpo destino-corto	Salta si NO paridad/paridad impar
jns destino-corto	Salta si no hay sobreflujo
jo destino-corto	Salta si sobreflujo
jp/jpe destino-corto	Salta si paridad / paridad par
js destino-corto	Salta si signo
Instrucciones de ciclos	
jcxz destino-corto	Salta si cx es igual a 0
loop destino-corto	Ciclar mientras $cx \neq 0$
loope/loopz destino	Ciclar mientras igual/0
loopne/loopnz destino	Ciclar mientras no igual/0
Instrucciones de Control de Interrupciones	

⁷⁵ MSI. Genaro Méndez L. (2010). *Material Didáctico y Referencia del LENGUAJE ENSAMBLADOR*. México: Instituto Tecnológico de Veracruz

⁷⁶ Iden (75).

int tipo interrupción	Interrupción
Into	Interrupción en sobreflujo
iret	Retorno de interrupción

Cuadro N° 6. Instrucciones de transferencia ⁷⁶

1.2 Salto Incondicional

El 8086 tiene más de una docena de instrucciones de saltos diferentes. Uno de estos, `jmp`, es un salto incondicional; todos los demás son saltos condicionales. La diferencia entre los dos tipos de saltos es muy importante:

- Un salto incondicional siempre causa que el programa inicie corriendo en una nueva dirección.
- Un salto condicional causa que el programa inicie corriendo a una nueva dirección solo si ciertas condiciones son satisfechas. De otra manera, el programa continúa como si la instrucción de salto condicional no existiese.

El `jmp` incondicional trabaja en forma idéntica a `call`, excepto que la dirección de retorno no es colocada en el stack. La instrucción `jmp` toma un solo parámetro: la etiqueta de la localidad donde el programa transferirá el control.

jmp Salida

1.3 Salto Condicional

La tabla siguiente lista las 18 instrucciones de salto condicional del 8086, muchas de las cuales tienen dos mnemónicos representando la misma instrucción, por ejemplo, `je/jz` y `jg/jnle` haciendo un total de 30 mnemónicos. Esto puede parecer un gran número de saltos condicionales que aprender, pero, como las conjugaciones de los verbos, las diferentes formas son más fáciles de aprender si se separa la raíz (siempre `j` para saltar), de las terminaciones (`a`, `nbe`, `e`, `z`, etc.) Cada una de estas terminaciones representan una condición única. Una vez que se memorizan sus significados, se tendrán menos problemas diferenciando los muchos tipos de saltos condicionales. En la tabla, las terminaciones a la derecha son negaciones de las terminaciones a la izquierda. (Dos mnemónicos de salto condicional, `jpe` y `jpo` no tienen contrapartes negativas).

Todos los saltos condicionales requieren de una dirección destino - una etiqueta marcando la localización donde se desea el programa siga corriendo si la condición especificada se cumple.

Por ejemplo, siguiendo a la comparación de dos registros con `cmp`, se puede utilizar `je` (salta si igual) para transferir el control a una localización si los valores de los registros son iguales.

⁷⁷ Iden (75)



Tér m	Significado	Traducción	Tér m	Significado	Traducción
a	Above	superior	na	not above	no superior
ae	above or equal	superior o igual	nae	not above or equal	no superior o igual
b	Below	inferior	nb	not below	no inferior
be	below or equal	inferior o igual	nbe	not below or equal	no inferior o igual
c	Carry	acarreo	nc	not carry	no acarreo
e	Equal	igual	ne	not equal	no igual
g	Greater	mayor	ng	not greater	no mayor
ge	grater or equal	mayor o igual	nge	not grater or equal	no mayor o igual
l	Less	menor	nl	not less	no menor
le	less or equal	menor o igual	lne	not less or equal	no menor o igual
o	overflow	sobreflujo	no	not overflow	no sobreflujo
p	parity	paridad	np	not parity	no paridad
pe	parity even	paridad par			
po	parity odd	paridad impar			
s	Sign	signo	ns	not sign	no signo
z	Zero	Cero	nz	not zero	no cero

Cuadro N° 7. Instrucciones de Comparación

1.3.1 Restricciones

Todos los saltos condicionales tienen una gran restricción: pueden transferir control solo hacia una distancia muy corta - exactamente 128 bytes hacia atrás (hacia direcciones bajas) o 127 bytes hacia adelante (hacia direcciones altas) desde el primer byte de la siguiente instrucción inmediata al salto. No hay que preocuparse, el Ensamblador informará si se intenta realizar un salto que sobrepase este rango.

1.4 Subprogramas

1.4.1 Procedimientos

En ensamblador un procedimiento es equivalente a una función en C o Pascal. Es también un mecanismo sencillo de encapsular partes de código. Para definir un procedimiento se siguen los siguientes pasos:

```

PROC AProcedure
.
.           ; código
.
ret
ENDP AProcedure

```

La llamada para ejecutar este procedimiento es:

```
call Aprocedure
```

Ejemplo de un procedimiento que envía a pantalla Hola

```

.model tiny
.stack
.code

```

Start:

```
call Display_Hola      ; Llamada al procedimiento Hola
mov ax,4C00h          ; vuelta al DOS
int 21h               ; fin de programa
```

```
Display_Hola PROC
mov dx,OFFSET HolaMuyBuenas
mov ah,9
int 21h
ret
Display_Hola ENDP
```

```
.data
HolaMuyBuenas DB 13,10 "Que pasa amigo $" ; mensaje
end Start
```

1.4.2 Paso de parámetros a un procedimiento

La utilidad de un procedimiento es la modificación de valores desde el procedimiento. Para ello es necesario el paso de parámetros. Existen tres formas de hacer esto: por registro, por memoria o por pila (stack).

1.4.2.1 Por registro

Es el más sencillo y el más rápido. Se necesita mover los parámetros a los registros y después hacer la llamada al procedimiento.

```
; imprime en pantalla un cuadro usando registros
; como paso de parámetros
.model tiny
.code
.stack
Start:
```

```
mov dh,4          ; posición de la fila en pantalla
mov dl,5          ; posición columna en pantalla
mov al,254        ; ASCII del carácter a pantalla
mov bl,4          ; color
```

```
call PrintChar    ; a pantalla
mov ax,4C00h      ; fin del programa
int 21h
```

```
PrintChar PROC
```

```
push bx           ; salvaguarda registros
push cx
xor bh,bh         ; limpia registro bh – página video 0
mov ah,2          ; función 2 - mover cursor
int 10h           ; fila x columnas ya en dx
pop bx            ; recupera bx
xor bh,bh         ; limpia registro bh – página video 0
mov ah,9          ; función 09h escribe carácter
mov cx,1          ; cx se usa como contador, se pone un 1
int 10h           ; llamada a servicio
```





```

pop cx      ; recupera registro
ret         ; return
PrintChar ENDP
end Start

```

1.4.2.2 Por memoria

También es sencillo, pero el programa resultante es más grande y más lento que el anterior. Para pasar parámetros a través de memoria es necesario copiar estos en variables almacenados en memoria.

```

; lo mismo pero por paso de parámetros
.model tiny
.code
.stack

```

Start:

```

mov Row,4      ; fila
mov Col,5      ; columna
mov Char,254   ; ASCII
mov Colour,4   ; color
call PrintChar ; llamada a función
mov ax,4C00h   ; termina programa
int 21h
PrintChar PROC

```

```

push ax cx bx  ; guarda registros
xor bh,bh      ; limpia bh – página de video 0
mov ah,2       ; funcion 2 - mover cursor
mov dh,Row
mov dl,Col
int 10h        ; llamada a la Bios
mov al,Char
mov bl,Colour
xor bh,bh      ;
mov ah,9       ; funcion 09h escribe
mov cx,1       ;
int 10h        ; llamada a servicio de la Bios
pop bx cx ax   ; recupera registros
ret           ; return
PrintChar ENDP
; variables para almacenar datos
Row db ?
Col db ?
Colour db ?
Char db ?
end Start

```

1.4.2.3 por pila (stack)

Debido a sus características, es el más flexible. No por ello más complicado.

```

; lo mismo usando el Stack

```

```

.model tiny
.code
.stack
Start:
mov dh,4           ; fila
mov dl,5           ; columna
mov al,254         ; ascii
mov bl,4           ; color
push dx ax bx      ; guarda parametros en stack
call PrintString   ; funcion a pantalla
pop bx ax dx       ; recupera registros
mov ax,4C00h       ; termina programa
int 21h

PrintString PROC
push bp            ; guarda bp
mov bp,sp          ; mueve sp a bp
push cx            ; guarda registro (se va a usar cx)
xor bh,bh          ; limpia bh
mov ah,2           ; funcion 2 - mover cursor
mov dx,[bp+8]      ; recupera dx
int 10h            ; llamada a Bios
mov ax,[bp+6]      ; caracter
mov bx,[bp+4]      ; atributo
xor bh,bh          ; pantalla en página 0
mov ah,9           ; funcion 09h escribe
mov cx,1
int 10h            ; llamada de servicio a bios
pop cx             ; recupera de stack
pop bp
ret                ; return
PrintString ENDP
end Start

```

1.5 Instrucciones Aritméticas

La mayoría de computadores son grandiosos para las matemáticas, esto viene a sorprender que el lenguaje ensamblador solo tiene unos operadores matemáticos relativamente primitivos. No hay símbolos de exponenciación, no hay punto flotante, no hay raíz cuadradas, y no existen funciones SENO y COSENO dentro del grupo de instrucciones del 8086. Las instrucciones Matemáticas en lenguaje ensamblador están restringidas a sumar, multiplicar, dividir y restar valores enteros con signo o sin signo.



INSTRUCCIONES GENERALES MNEMÓNICO/OPERANDO	DESCRIPCIÓN
Instrucciones de Adición	
aa	Ajuste ASCII para adición
adc destino, fuente	Suma con acarreo
add destino, fuente	Suma bytes o palabras
daa	Ajuste decimal para adición
inc destino	Incremento
Instrucciones de Substracción	
aas	Ajuste ASCII para substracción
cmp destino, fuente	Compara
Das	Ajuste decimal para substracción
dec destino	Decrementa byte o palabra
neg destino	Negar (complemento a dos)
sbb destino, fuente	Subtrae
sub destino, fuente	Subtrae
Instrucciones de Multiplicación	
Aam	Ajuste ASCII para multiplicación
imul fuente	Multiplicación con enteros
mul fuente	Multiplicar
Instrucciones de División	
Aad	Ajuste ASCII para división
Cbw	Convierte bytes a palabras
Cwd	Convierte palabras a dobles palabras
div fuente	Divide
idiv fuente	División de Enteros

Cuadro N° 8. Instrucciones Aritméticas

Ejemplo:

%TITLE "Demostración de ADD,SUB,INC,DEC"

IDEAL

DOSSEG

MODEL small

STACK 256

DATASEG

codSalida DB 0

contador DW 1

CODESEG

Inicio:

```

mov ax, @data      ; Inicializa la dirección
mov ds, ax         ; del segmento de datos
mov ax, 4
mov bx, 2
add ax, bx          ; ax<-ax+bx
mov cx, 8
add cx, [contador] ; cx<-cx+[contador]
add [contador], cx  ; [contador]<-[contador]+cx
inc [contador]      ; [contador]<-[contador]+1
dec [contador]      ; [contador]<-[contador]-1
inc ax              ; ax<-ax+1
dec cx              ; cx<-cx-1

```

Fin:

```
mov    ah, 04Ch; Función de DOS: Salir del programa
mov    al, [codSalida]; Retorno el Código de salida
int     21h          ; Llama a DOS. Terminar programa
END     Inicio        ; Fin del programa
```

1.6 Instrucciones Lógicas

Las instrucciones lógicas están agrupadas en dos subdivisiones: lógicas e instrucciones de rotación/corrimiento. Las instrucciones lógicas combinan bytes y palabras con AND, OR, y otras operaciones lógicas. Las Instrucciones de rotación/corrimiento, realizan corrimientos y rotaciones de bits en bytes y palabras.

Mnemónico/Operando	Descripción
Instrucciones Lógicas	
and destino, fuente	AND lógico
not destino	NOT lógico, complemento a 1
or destino, fuente	OR lógico
test destino, fuente	Prueba bits
xor destino, fuente	OR Exclusivo lógico
Instrucciones de Corrimiento/Rotación	
rcl destino, contador	Rota hacia la izquierda con acarreo
rcr destino, contador	Rota hacia la derecha con acarreo
rol destino, contador	Rota hacia la izquierda
ror destino, contador	Rota hacia la derecha
sar destino, contador	Corrimiento aritmético a la derecha
shl/sal destino, contador	Corrimiento a la izquierda/aritmético
shr destino, contador	Corrimiento a la derecha

Cuadro N° 9. Instrucciones Lógicas

La instrucción lógica más simple not, cambia los bits en un byte o palabra de ceros a unos y los unos a ceros. Esto es llamado complemento a uno, (sumar uno a este resultado forma el complemento a dos, aunque es más fácil usar neg para este propósito). Una forma de utilizar not es para cambiar valores de verdadero y falso. Si un valor cero representa falso y un valor no cero representa verdadero. La siguiente instrucción cambia el registro dh de verdadero a falso y lo regresa a verdadero.

```
mov    dh,1          ; Inicializa dh a verdadero
not     dh            ; cambia dh a "no verdadero", falso
not     dh            ; cambia dh a "no falso", verdadero
```

⁷⁹ Iden (75)





LECTURA SELECCIONADA N° 1

DIEZ LENGUAJES EXPERIMENTALES QUE PODRÍAN IMPACTAR EN UN FUTURO.

En la actualidad disponemos de gran variedad de lenguajes. Un programador medio difícilmente puede llegar a aprender el 5% de los estos lenguajes. Cada lenguaje tienen una finalidad, unas ventajas y unos inconvenientes. También tienen una época en la que surgen para cubrir una necesidad inexistente y pasan a estar deprecated cuando un lenguaje más avanzado puede reemplazarlo.

He escuchado alguna vez comentarios afirmando que en la programación estaba ya todo inventado. Por supuesto está inventado todo lo que uno utiliza y se puede llegar a pensar que no es necesario más. Pero creo que esto no ha hecho nada más que empezar. Muchos lenguajes cubren gran parte de las necesidades que existen actualmente. Sin embargo, siempre surgen iniciativas para crear o modificar la perspectiva de nuestros desarrollos evolucionando a otros lenguajes.

Vía Infoworld he encontrado un interesante artículo acerca de 10 lenguajes experimentales que en la actualidad no son conocidos y puede que en un futuro tengan un gran impacto asentándose en nuestras vidas. Aunque estudiarlos detalladamente es costoso, pues siempre existe alta probabilidad de que los paradigmas cambien, es interesante conocer al menos porqué y para qué se han creado. Una lista para los más intrépidos y vanguardistas.

Dart

Lenguaje de reciente aparición desarrollado por Google diseñado para mejorar la capa web cliente. El objetivo no es sustituir JavaScript ya que en ocasiones puede venir bien para aplicaciones web sencillas. Sin embargo, si pretende convertirse en un referente para aplicaciones web donde en el cliente existe gran carga de líneas de código.

Su sintaxis es similar a C y permite definir clases e interfaces como en C++ o Java. De momento, se encuentra en una fase alfa donde aún no se pueden realizar desarrollos serios. Sin embargo, ha sido muy acogida la noticia al respecto por parte de la comunidad.

Ceylon

Pretende ser la evolución de Java. Gavin King, desarrollador en Red Hat con este lenguaje, niega que se pretenda matar a Java. Sin embargo, afirma que hay mucho que mejorar. Java tiene una sintaxis excesivamente detallada, faltan funciones fundamentales y hay poco apoyo a la meta-programación.

Por otra parte Gavin se siente frustrado por el abuso en la actualidad de la programación declarativa en el que toda la aplicación se encadena mediante XML y dejando de lado la programación imperativa. Sin embargo, no pretenden reinventar la rueda. Ceylon compila a bytecodes de Java. Pero si pretenden reemplazar el SDK en el que según Gavin afirma: “Está hinchado y es torpe”.

Go

Go es un lenguaje iniciado en el 2009. Este es otro lenguaje experimental de Google de propósito general en el que se busca facilitar la sintaxis al puro estilo C, pero con características modernas como la recolección de basura, reflexión en tiempo de ejecución y apoyo a la concurrencia. Go sigue siendo un trabajo en proceso en el que la especificación del lenguaje podría cambiar.

F#

F# es un lenguaje diseñado por Microsoft que pretende aportar mejoras a aquellos desarrolladores que les gusta la programación funcional. Una de las quejas que existen en la actualidad es que utilizando lenguajes de programación funcional tipo Haskell o Lisp no es posible utilizar bibliotecas o características de otros lenguajes CLR (C# y Visual Basic). Ahora con F# ya no es un problema ya que se está incluido en Visual Studio 2010 permitiendo esta compatibilidad y

utilizar programación funcional.

Opa

Este lenguaje pretende aplicar un nuevo paradigma al desarrollo web. Para realizar una aplicación web es necesario el conocimiento de lenguajes en el cliente (HTML y JavaScript) y en el servidor (PHP o Java). Del mismo modo, SQL también lleva su propio lenguaje. Opa pretende eliminar todas estas dependencias o conocimientos requeridos en diversos lenguajes y centrarse en un único lenguaje llamado precisamente Opa.

Opa decidirá si una rutina debe ejecutarse en el cliente, en el servidor o en ambos así como determinar por donde se realizará la salida realizando la transmisión de datos mediante Ajax en el caso de ser en el cliente. Este sistema requiere algo de Back-End ya que se dispone de su propio servidor web y gestión de datos y no puede ser reemplazado con alternativas Stand-Alone.

Fantom

Fantom ha sido diseñado para desarrollar aplicaciones portables entre plataformas. La compilación permite crear ficheros bytecodes tanto para la JVM como para la CLINet. También permite el acceso al conjunto de APIs disponibles de Java y .Net creando una capa adicional.

Existen planes de extender la portabilidad a más plataformas. Por ejemplo, ya hay disponible un compilador Fantom-a-JavaScript. Las expectativas son que permita en un futuro exportar a Parrot VM y Objective-C.

Zimbu

Lenguaje creado por Bram Moolenaar, creador del editor de texto Vim. Zimbu pretende que el lenguaje que se escriba sea sencillo, conciso, fácil de leer y rápido de escribir y que no sean necesarias tantas herramientas gráficas. La programación cuanto más breve, mejor.

Es un lenguaje compilado en el que los binarios se construirán en un compilador C nativo. Por otra parte, es muy rico en funciones aunque no están desarrolladas todas las especificadas ya que, en la actualidad, este proyecto se encuentra en una fase Alfa.

X10

Unas de las cualidades que muchos lenguajes no cubren en la actualidad es la ejecución concurrente de procesos para el caso de tener múltiples núcleos. X10 intenta abordar esta necesidad. El código se divide en unidades que son ejecutadas por núcleos o procesadores diferentes permitiendo de esta manera escalar en rendimiento añadiendo más núcleos.

La sintaxis de X10 es parecida a Java y permite crear ficheros fuente C++ y Java. Es un lenguaje bastante maduro aunque siguen realizándose modificaciones. Para facilitar el desarrollo de aplicaciones existe un IDE basado en Eclipse y un depurador.

haXe

hAxe pretende ser un lenguaje multiplataforma que permita desarrollar para diferentes entornos tanto de manera nativa como utilizando máquinas virtuales. A partir de código fuente haXe es posible compilarlos devolviendo OBJs, JavaScript, PHP, Flash/ActionScript, bytecode NekoVM, C# o Java. Dispone de una biblioteca estándar que funciona en todos los anteriores tipos de compilación.

La sintaxis es similar a C y, aunque se encuentra en desarrollo, haXe es utilizado comercialmente por su creador. Lo que se comúnmente se conoce como “comer tu propia comida de perros” para demostrar su valor.

Chapel

Chapel pretende ser una ambiciosa iniciativa financiada por EE.UU. a través de la sociedad DARPA (Agencia de Investigación de Proyectos Avanzados de Defensa). Chapel tiene como objetivo crear un lenguaje de alto rendimiento y la





ejecución de algoritmos en paralelo para la supercomputación.

La sintaxis tiene influencias de C, C++ y Java y toma conceptos de programación científica como Fortran y Matlab. La influencia del procesamiento en paralelo viene dado por ZPL y Fortran. En la actualidad se utiliza en supercomputadores y se dispone de una versión portátil para sistemas tipo Unix con licencia BSD.

Conclusión

Hasta aquí los diez lenguajes experimentales que pueden ser muy interesantes. En mi opinión me llaman la atención Dart por su acogida en la comunidad sin estar acabado, Ceylon por su retorno a la programación imperativa y no pretender abusar de la programación XML y haXe por tener la intención de unificar todas las diferentes librerías en el que cada lenguaje utiliza nombres diferentes para lo mismo, en una única librería estándar.



ACTIVIDAD N° 2

Esta actividad puede consultarla en su aula virtual.

TEMA N° 2: ARQUITECTURAS AVANZADAS⁸⁰

Hoy en día, en nuestra vida cotidiana, tenemos la necesidad de realizar un sinnúmero de actividades en el menor tiempo posible. En la mayoría de las ocasiones no nos percatamos cómo es que estas actividades poco a poco resultan ser más rápidas y sencillas para el ser humano

Pues bien, lo que ocurre internamente en los procesadores de los sistemas, lo cual para el usuario final es totalmente transparente, tiene que ver en gran parte con el tema de esta unidad, el procesamiento paralelo. Esto no es otra cosa que realizar pequeñas divisiones de procesos, los cuales están compuestos por un fragmento de instrucciones, que son ejecutadas de forma simultánea en diferentes procesadores obteniendo así un tiempo de ejecución más corto. De hecho, ésta es una de las principales metas que tiene el procesamiento paralelo. Trabajar y poder procesar grandes volúmenes de información con el fin de obtener resultados en el menor tiempo posible es otra de sus ventajas, sin embargo también se enfrenta a algunas dificultades como son la falta de coordinación al tratar de ejecutar los procesos al mismo tiempo, así como su control y monitoreo. Sin duda alguna, este tema es muy importante por lo que se pretende que el alumno comprenda el funcionamiento, la estructura y las aplicaciones del paralelismo en los sistemas⁸¹

⁸⁰ IRMA FLORES RIOS(2012), *Arquitectura De Computadoras Ii.. España: Red Tercer Milenio* Pag 9

⁸¹ *Iden(80)*

⁸² *Iden(80)*



Figura Nro 70. Mapa conceptual Procesamiento Paralelo

2.1 Procesamiento Paralelo

2.1.1 Paralelismo en sistemas Monoprocesador

Un sistema monoprocesador es aquel que cuenta sólo con un procesador para ejecutar todos los procesos a realizar, bajo este concepto podríamos afirmar que en este tipo de sistemas no existe el paralelismo. En la siguiente figura se muestra un ejemplo de la diferencia que existe entre los sistemas monoprocesadores y los multiprocesadores, estos últimos siendo los que cuentan con más de un procesador para ejecutar los diferentes procesos de forma simultánea.

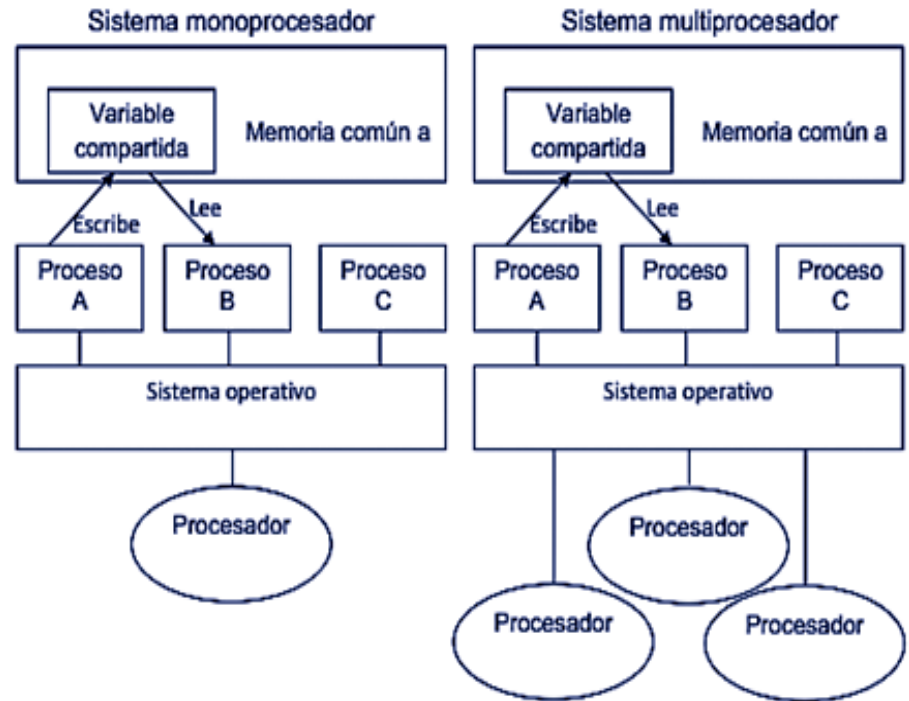


Figura Nro 71. Paralelismo en sistemas Monoprocesador

Una manera tradicional de incrementar las prestaciones de un sistema consiste en utilizar varios procesadores que puedan ejecutar en paralelo una carga de trabajo dada. Las dos organizaciones de múltiples procesadores más comunes son los multiprocesadores simétricos (SMP) y los Clusters. Recientemente, los sistemas de acceso no uniforme a memoria (NUMA) han aparecido comercialmente.

Un SMP es un computador constituido por varios procesadores similares, interconectados mediante un bus o algún tipo de estructura de conmutación. El problema más crítico a resolver en un SMP es la coherencia de caché. Cada procesador tiene su propia caché, y es posible que una línea de datos esté presente en más de una caché. Si esa línea se altera en una caché, entonces tanto la memoria principal como las otras cachés tienen versiones no válidas de dicha línea.

Cuando en un único chip se implementan varios procesadores, se habla de multiprocesador monochip. Un diseño relacionado consiste en repetir algunos componentes de un procesador para que este pueda ejecutar varias hebras concurrentemente. Es lo que se conoce como procesador multihebra.

Un cluster es un grupo de computadores completos interconectados y trabajando juntos como un solo recurso de cómputo, proporcionando la ilusión de ser una única máquina. El término computador completo significa que puede funcionar autónomamente, fuera del cluster.

Un sistema NUMA es un multiprocesador de memoria compartida en el que el tiempo de acceso de un procesador a una palabra de memoria varía con la ubicación de la palabra en memoria.

Tradicionalmente el computador se ha visto como una máquina secuencial. La mayoría de lenguajes de programación requieren que el programador especifique los algoritmos mediante una secuencia de instrucciones. Los procesadores ejecutan los programas procesando instrucciones máquina de una en una. Cada

⁸³ Inde (82)

⁸⁴ Iden (82)

instrucción se ejecuta mediante una secuencia de operaciones (captar instrucción, captar operandos, realizar la operación, almacenar los resultados).

Esta perspectiva del computador no es completamente cierta. En el nivel de microoperación se generan al mismo tiempo múltiples señales de control. La segmentación de las instrucciones, al menos en cuanto al solapamiento de las operaciones de captación y ejecución, se ha utilizado desde hace tiempo. Ambos casos son ejemplos de funciones que se realizan en paralelo. Es el mismo enfoque de la organización superescalar, que aprovecha el paralelismo entre instrucciones. Un procesador superescalar dispone de varias unidades de ejecución que pueden ejecutar en paralelo varias instrucciones de un programa.

A medida que la tecnología de los computadores se ha desarrollado, y ha disminuido el costo del hardware del computador, los diseñadores de computadores han visto más y más posibilidades en el paralelismo, normalmente para mejorar las prestaciones, y en algunos casos para mejorar la fiabilidad.

A continuación veremos las tres organizaciones paralelas de mayor éxito:

2.1.2 Tipos de sistemas paralelos según la Clasificación de Flynn

Clasificación clásica de arquitectura de computadores que hace alusión a sistemas con uno o varios procesadores. Flynn la publicó por primera vez en 1966 y por segunda vez en 1970.

Esta taxonomía se basa en el flujo que siguen los datos dentro de la máquina y de las instrucciones sobre esos datos.

Se define como flujo de instrucciones al conjunto de instrucciones secuenciales que son ejecutadas por un único procesador y como flujo de datos al flujo secuencial de datos requeridos por el flujo de instrucciones.

Con estas consideraciones, Flynn clasifica los sistemas en cuatro categorías:

2.1.2.1 SISD (Single Instruction stream, Single Data stream)

Los sistemas de este tipo se caracterizan por tener un único flujo de instrucciones sobre un único flujo de datos, es decir, se ejecuta una instrucción detrás de otra. Este es el concepto de arquitectura serie de Von Neumann donde, en cualquier momento, sólo se ejecuta una única instrucción, en otras palabras, un único procesador interpreta una única secuencia de instrucciones para operar con los datos almacenados en una única memoria.

Un ejemplo de estos sistemas son las máquinas secuenciales convencionales monoprocesador.

2.1.2.2 SIMD (Single Instruction stream, Multiple Data stream)

Una única instrucción máquina controla paso a paso la ejecución simultánea y sincronizada de un cierto número de elementos de proceso. Cada elemento de proceso tiene una memoria asociada, de forma que cada instrucción es ejecutada por cada procesador con un conjunto de datos diferentes. Los procesadores vectoriales y matriciales pertenecen a esta categoría.

El procesamiento es síncrono, la ejecución de las instrucciones sigue siendo secuencial como en el caso anterior, todos los elementos realizan una misma instrucción pero sobre una gran cantidad de datos. Por este motivo existirá concurrencia de operación, es decir, esta clasificación es el origen de la máquina paralela.

El funcionamiento de este tipo de sistemas es el siguiente. La Unidad de Control manda una misma instrucción a todas las unidades de proceso (ALUs). Las unidades de proceso operan sobre datos diferentes pero con la misma instrucción recibida.

Existen dos alternativas distintas que aparecen después de realizarse esta clasificación:

- Arquitectura Vectorial con segmentación:

Una CPU única particionada en unidades funcionales independientes trabajando sobre flujos de datos concretos



- Arquitectura Matricial (matriz de procesadores):

Varias ALUs idénticas a las que el procesador de instrucciones asigna una única instrucción pero trabajando sobre diferentes partes del programa.

2.1.2.3 MISD (Multiple Instruction stream, Single Data stream)

Sistemas con múltiples instrucciones que operan sobre un único flujo de datos. múltiples secuencias de instrucciones y una secuencia de datos:

Se transmite una secuencia de datos a un conjunto de procesadores, cada uno de los cuales ejecuta una secuencia de instrucciones diferente. Esta estructura nunca ha sido implementada

Los sistemas MISD se contemplan de dos maneras distintas:

- Varias instrucciones operando simultáneamente sobre un único dato.
- Varias instrucciones operando sobre un dato que se va convirtiendo en un resultado que será la entrada para la siguiente etapa. Se trabaja de forma segmentada, todas las unidades de proceso pueden trabajar de forma concurrente.

Ejemplos de estos tipos de sistemas son los arrays sistólicos o arrays de procesadores. También podemos encontrar aplicaciones de redes neuronales en máquinas masivamente paralelas.

2.1.2.4 MIMD (Multiple Instruction stream, Multiple Data stream)

Un conjunto de procesadores ejecuta simultáneamente secuencias de instrucciones diferentes con conjuntos de datos diferentes. Los SMP, los clusters y los sistemas NUMA son ejemplos de esta categoría.

En la organización MIMD los procesadores son de uso general; cada uno es capaz de procesar todas las instrucciones necesarias para realizar las transformaciones apropiadas de los datos



Figura Nro 72. Formas de Organización de Computadores⁸⁵

2.1.3 Categorías de Ordenadores Paralelos

Clasificación moderna que hace alusión única y exclusivamente a los sistemas que tienen más de un procesador (i.e máquinas paralelas).

Existen dos tipos de sistemas teniendo en cuenta su acoplamiento. Los sistemas fuertemente acoplados son aquellos en los que los procesadores dependen unos de otros. Los sistemas débilmente acoplados son aquellos en los que existe poca interacción entre los diferentes procesadores que forman el sistema.

Atendiendo a esta y a otras características, la clasificación moderna divide a los sistemas en dos tipos: Sistemas multiprocesador (fuertemente acoplados) y sistemas multicomputador (débilmente acoplados).

2.1.3.1 Multiprocesadores

Un multiprocesador puede verse como un computador paralelo compuesto por varios procesadores interconectados que comparten un mismo sistema de memoria.

Los sistemas multiprocesadores son arquitecturas MIMD con memoria compartida. Tienen un único espacio de direcciones para todos los procesadores y los mecanismos de comunicación se basan en el paso de mensajes desde el punto de vista del programador.

Dado que los multiprocesadores comparten diferentes módulos de memoria, pudiendo acceder a un mismo módulo varios procesadores, a los multiprocesadores también se les llama sistemas de memoria compartida.

Dependiendo de la forma en que los procesadores comparten la memoria, se clasifican en sistemas multiprocesador UMA, NUMA y COMA.

2.1.3.2 UMA (Uniform Memory Access)

Sistema multiprocesador con acceso uniforme a memoria. La memoria física es uniformemente compartida por todos los procesadores, esto quiere decir que todos los procesadores tienen el mismo tiempo de acceso a todas las palabras de la memoria. Cada procesador tiene su propia caché privada y también se comparten los periféricos.

Los multiprocesadores son sistemas fuertemente acoplados (tightly-coupled), dado el alto grado de compartición de los recursos (hardware o software) y el alto nivel de interacción entre procesadores, lo que hace que un procesador dependa de lo que hace otro.

El sistema de interconexión debe ser rápido y puede ser de uno de los siguientes tipos:

- Bus común
- Red crossbar
- Red multietapa

Este modelo es conveniente para aplicaciones de propósito general y de tiempo compartido por varios usuarios.

Existen varias categorías de sistemas UMA.

- **Sistema Simétrico**

Cuando todos los procesadores tienen el mismo tiempo de acceso a todos los componentes del sistema (incluidos los periféricos), reciben el nombre de sistemas multiprocesador simétrico.



⁸⁵ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education p. 629*



Los procesadores tienen el mismo dominio (prioridad) sobre los periféricos y cada procesador tienen la misma capacidad para procesar.

- **Sistema Asimétrico**

Los sistemas multiprocesador asimétrico, son sistemas con procesadores maestros y procesadores esclavos, en donde sólo los primeros pueden ejecutar aplicaciones y dónde en tiempo de acceso para diferentes procesadores no es el mismo. Los procesadores esclavos (attached) ejecutan código usuario bajo la supervisión del maestro, por lo tanto cuando una aplicación es ejecutada en un procesador maestro dispondrá de una cierta prioridad.

2.1.3.3 NUMA (Non Uniform Memory Access)

Un sistema multiprocesador NUMA es un sistema de memoria compartida donde el tiempo de acceso varía según donde se encuentre localizado el acceso.

El acceso a memoria, por tanto, no es uniforme para diferentes procesadores. Existen memorias locales asociadas a cada procesador y estos pueden acceder a datos de su memoria local de una manera más rápida que a las memorias de otros procesadores, debido a que primero debe aceptarse dicho acceso por el procesador del que depende el módulo de memoria local.

Todas las memorias locales conforman la memoria global compartida y físicamente distribuida y accesible por todos los procesadores.

2.1.4 Cluster Jerárquico

Otro modelo NUMA que nace como la mezcla del modelo UMA explicado anteriormente y el modelo NUMA anterior, es el cluster jerárquico en el que se combinan las memorias locales y las globales obteniendo una cierta escalabilidad del sistema.

Los procesadores aparecen distribuidos en clusters (1 sistema UMA o un 1 sistema NUMA). Estos clusters están conectados a la memoria global compartida. El sistema en su totalidad es un sistema NUMA, ya que el acceso a memoria es no uniforme por parte de los clusters.

La ventaja de estos sistemas con respecto a los sistemas UMA, es que el acceso a memoria local es mucho más rápido.

2.1.5 COMA (Cache Only Memory Access)

Los sistemas COMA son un caso especial de los sistemas NUMA. Este tipo de sistemas no ha tenido mucha transcendencia, al igual que los sistemas SIMD.

Las memorias distribuidas son memorias cachés, por este motivo es un sistema muy restringido en cuanto a la capacidad de memoria global. No hay jerarquía de memoria en cada módulo procesador. Todas las cachés forman un mismo espacio global de direcciones. El acceso a las cachés remotas se realiza a través de los directorios distribuidos de las cachés.

Dependiendo de la red de interconexión utilizada, se pueden utilizar jerarquías en los directorios para ayudar a la localización de copias de bloques de caché.

2.2 Arquitectura Superescalar

Tipo de micro arquitectura de procesador capaz de ejecutar más de una instrucción por ciclo de reloj. Para ello se usan múltiples cauces, con lo que varias instrucciones pueden iniciar su ejecución de manera independiente.

El término superescalar se emplea como contraposición a la arquitectura escalar, que solo es capaz de ejecutar una instrucción por ciclo de reloj. En la figura se representa un esquema de la organización superescalar. Se tienen múltiples unidades funcionales que admiten la ejecución en paralelo, y por separado, de varias instrucciones. Sin embargo, nótese que, en general, cada unidad funcional no tiene por qué admitir la ejecución de cualquier instrucción, hay dos unidades funcionales que pueden ejecutar instrucciones de operaciones con enteros, dos de instrucciones de operaciones con coma flotante, y una de instrucciones de transferencia con memoria (operaciones de carga y almacenamiento). Así, en este caso, pueden estar ejecutándose al mismo tiempo dos operaciones con enteros, dos operaciones con coma flotante y una

operación de transferencia a memoria.

El diseño superescalar es compatible con la segmentación. Esto es, como muestra la figura 71, las unidades funcionales pueden realizarse como cauces segmentados

El concepto de diseño superescalar apareció poco después que la arquitectura RISC. Si bien se puede realizar una implementación superescalar tanto de una arquitectura RISC como de una arquitectura CISC, la arquitectura de conjunto reducido de instrucciones de una máquina RISC se presta mejor a la utilización de técnicas superescalares

Lo esencial del diseño superescalar es su habilidad para ejecutar instrucciones de manera independiente en diferentes cauces. Un procesador superescalar es capaz de ejecutar más de una instrucción, solo si estas no presentan algún tipo de dependencia, como veremos más adelante

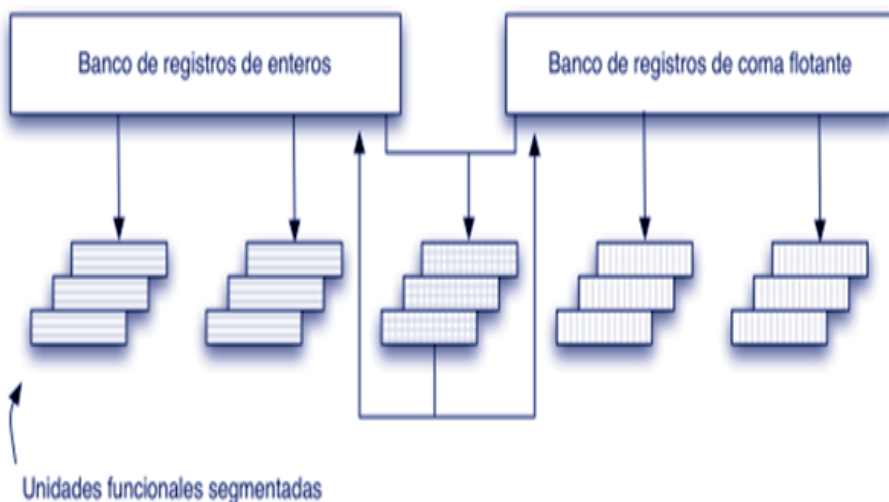


Figura Nro 73. Organización de los registros⁸⁶

La inmensa mayoría de las CPUs desarrolladas desde 1998 son superescalares.

La micro arquitectura superescalar utiliza el paralelismo de instrucciones además del paralelismo de flujo, éste último gracias a la estructura en pipeline. La estructura típica de un procesador superescalar consta de un pipeline con las siguientes etapas:

- Lectura (fetch).
- Decodificación (decode).
- Lanzamiento (dispatch).
- Ejecución (execute).
- Escritura (writeback).
- Finalización (retirement).

En un procesador superescalar, el procesador maneja más de una instrucción en cada etapa. El número máximo de instrucciones en una etapa concreta del pipeline se denomina grado, así un procesador superescalar de grado 4 en lectura (fetch) es capaz de leer como máximo cuatro instrucciones por ciclo. El

⁸⁶ WILLIAM STALLINGS (2006). *Organización y arquitectura de computadores –España: Pearson Education p. 697*



grado de la etapa de ejecución depende del número y del tipo de las unidades funcionales.

Un procesador superescalar suele tener unidades funcionales independientes de los tipos siguientes:

- Unidad aritmético lógica (ALU)
- Unidad de lectura/escritura en memoria (Load/Store Unit)
- Unidad de coma flotante (Floating Point Unit)
- Unidad de salto (Branch unit)

2.2.1 Explotación del paralelismo a nivel de instrucción

El hecho de que la implementación superescalar tenga varios cauces introduce un nuevo nivel de paralelismo. Los procesadores superescalares sacan provecho del paralelismo a nivel de instrucción, que hace referencia al grado en el que, en promedio, las instrucciones de un programa se pueden ejecutar en paralelo. Para ello implementan técnicas que permiten romper el flujo secuencial de instrucciones de un programa, para simultanear la ejecución de varias, en el mismo procesador

No hay procesador moderno que no use alguna forma de paralelismo a nivel de instrucción. Los cauces segmentados, escalares y supersegmentados son diferentes técnicas que utilizan el paralelismo a nivel de instrucción. En este tema se estudiarán las características de las implementaciones superescalares que explotan el paralelismo a nivel de instrucción de forma implícita, internamente. Es decir, aquellos procesadores superescalares cuyo hardware implementa las técnicas que permiten romper el flujo de instrucciones del programa, para ejecutarlas en paralelo, incluso de forma desordenada, y lo hacen de manera transparente al programador, y en tiempo de ejecución. Diferencias entre superescalar y supersegmentación Si bien el diseño superescalar es compatible con la supersegmentación, no son técnicas similares. Por ello, se dedica este apartado a la descripción de la supersegmentación.

Y es que la palabra supersegmentación nos lleva a la tentación de aplicar la ecuación superescalar + segmentado = supersegmentado, cuando en realidad la ecuación correcta sería segmentado+ segmentado=supersegmentado

La supersegmentación es una técnica que saca provecho del paralelismo a nivel de instrucción con el objeto de alcanzar mayores prestaciones.

2.2.2 Limitaciones fundamentales del paralelismo

La aproximación superescalar introduce la capacidad de ejecutar múltiples instrucciones en paralelo. Sin embargo, no se puede realizar la ejecución en paralelo de instrucciones de forma indiscriminada, debido a la existencia de ciertas limitaciones fundamentales del paralelismo a las que el sistema tiene que enfrentarse. Estas son: - Dependencia de datos verdadera

- Dependencia relativa al procedimiento
- Conflictos con los recursos
- Dependencia de salida
- Antidependencia



LECTURA SELECCIONADA N° 2

¿Por qué máquinas vectoriales?

Si bien es cierto que la segmentación se presenta como un mecanismo importante para mejorar el rendimiento de una computadora, es aún más cierto que existen límites en la mejora del rendimiento que la segmentación puede conseguir.

Habría que decir que los procesadores segmentados de alta velocidad son útiles para grandes aplicaciones científicas y de ingeniería, sin embargo, los programas científicos grandes, cuya ejecución es larga, tienen con frecuencia conjuntos de datos activos muy grandes que son accedidos a menudo, consiguiendo un rendimiento pobre de la jerarquía de memoria. El impacto resultante es una disminu-

ción en el rendimiento de la cache.

Las máquinas vectoriales proporcionan operaciones de alto nivel que trabajan sobre vectores (arrays lineales de números).

Las operaciones vectoriales pueden hacerse más rápidas que una secuencia de operaciones escalares sobre el mismo número de elementos de datos, y los diseñadores están motivados para incluir unidades vectoriales si el conjunto de las aplicaciones las puede usar frecuentemente.

Una operación vectorial normal puede sumar dos vectores de 64 elementos en punto flotante para obtener como resultado un vector de 64 elementos.

La instrucción vectorial es equivalente a un bucle completo, donde en cada iteración se calcula uno de los 64 elementos del resultado, actualizando los índices y saltando al comienzo.

Arquitectura vectorial básica.

Una máquina vectorial, normalmente, consta de una unidad escalar segmentada más una unidad vectorial. Todas las unidades funcionales de la unidad vectorial tienen una latencia de varios ciclos de reloj. Esto permite un ciclo de reloj de menor duración y es compatible con operaciones vectoriales de larga ejecución que pueden ser segmentadas a nivel alto sin generar riesgos. La mayoría de las máquinas vectoriales permiten que los vectores sean tratados como números en punto flotante, como enteros o como datos lógicos.

Tipos principales de arquitecturas vectoriales:

Máquinas vectoriales con registros. En este tipo de máquina todas las operaciones vectoriales -excepto las de carga y almacenamiento- operan con vectores almacenados en los registros. Estas máquinas son el equivalente vectorial de una arquitectura escalar de carga/almacenamiento. Ejemplos de máquinas vectoriales construidas en 1990 que utilizan una arquitectura vectorial son:

- Las máquinas de Cray Research: CRAY-1, CRAY-2, X-MP e Y-MP.
- Los supercomputadores japoneses: NEC SX/2, Fujitsu VP2000 y el Hitachi S820.
- Los minisupercomputadores: Convex C-1 y C-2.

Máquinas vectoriales memoria-memoria.

En este tipo de máquina todas las operaciones vectoriales son de memoria a memoria. Las primeras máquinas vectoriales fueron de este tipo, como por ejemplo las máquinas de CDC. Es necesario mencionar que éstas no tuvieron el mismo éxito que las máquinas vectoriales con registros.

En la actualidad estas arquitecturas se están empleando en labores tan diversas como veremos a continuación:

Los tipos de problemas encontrados comúnmente en las aplicaciones de computación en paralelo son:

- Álgebra lineal densa
- Álgebra lineal dispersa
- Métodos espectrales (tales como la transformada rápida de Fourier de Cooley-Tukey)
- Problemas de n-cuerpos (tales como la simulación Barnes-Hut)
- Problemas de grids estructurados (métodos de Lattice Boltzmann)
- Problemas de grids no estructurados (tales como los encontrados en el análisis de elementos finitos)



⁸⁷ JOHN PAUL SHEN y MIKKO H. LIPASTI (2005-2006) *Arquitectura de Computadores: Fundamentos de los procesadores superescalares*. México: McGraw-Hill



- Simulación de Montecarlo
- Lógica combinatorial (por ejemplo, técnicas criptográficas de fuerza bruta)
- Recorridos en grafos (por ejemplo, los algoritmos de ordenamiento)
- Programación dinámica
- Métodos de Ramificación y poda
- Modelos en grafos (tales como la detección de modelos ocultos de Markov y la construcción de redes bayesianas)
- Simulación de autómatas finitos



ACTIVIDAD N° 2

Esta actividad puede consultarla en su aula virtual.



TAREA ACADÉMICA N°1

Esta actividad puede consultarla en su aula virtual.



BIBLIOGRAFÍA BÁSICA DE LA UNIDAD IV

MSI. Genaro Méndez L. (2010). *Material Didáctico y Referencia del LENGUAJE EN-SAMBLADOR*. México: Instituto Tecnológico de Veracruz

IRMA FLORES RIOS (2012), *Arquitectura De Computadoras Ii.*. España: Red Ter-cer Milenio

JOHN PAUL SHEN y MIKKO H. LIPASTI (2005-2006) *Arquitectura de Computado-res: Fundamentos de los procesadores superescalares*. Mexico: McGraw-Hill

ENLACES DE INTERES

- http://informatica.uv.es/iiguia/AAC/AA/apuntes/aac_otras.pdf
- <http://www.tesis.ufm.edu.gt/pdf/2050.pdf>
- <http://wwdi.ujaen.es/asignaturas/ArqTel/teoria/Tema1.pdf>
- <http://serdis.dis.ulpgc.es/~itig-fc2/Teoria/Tema6/6.%20Entrada-Salida.pdf>
- <http://www.genbetadev.com/autor/jorge-rubira>



AUTOEVALUACIÓN DE LA UNIDAD IV

Elige la alternativa que corresponda:

1. Cuál de las siguientes características corresponde a un computador SIMD:
 - a. Posee múltiples procesadores con múltiples contadores de programa.
 - b. Posee un único procesador, con múltiples contadores de programa.
 - c. Posee múltiples procesadores, con un único contador de programa.
2. Cuál de las siguientes no es una característica de un computador SIMD:
 - a. Ejecuta con procesamiento síncrono entre sus múltiples procesadores.
 - b. Cada uno de los elementos ejecuta instrucciones diferentes en un mismo tiempo dado.
 - c. Existe concurrencia de operaciones.
3. Como opera un procesador MISD:
 - a. Cada procesador opera con la misma secuencia de datos pero diferentes instrucciones.
 - b. Cada procesador va tomando un resultado del procesamiento del procesador anterior.
 - c. Todos los procesadores que posee el sistema operan sobre una única instrucción.
4. Los sistemas multiprocesador se caracterizan por:
 - a. Cada procesador posee su propio sistema de memoria independiente
 - b. Se les denomina sistemas de memoria dedicada.
 - c. El conjunto de procesadores comparte un único sistema de memoria.
5. Un sistema UMA, con respecto a la memoria Cache se caracteriza por:
 - a. Cada procesador tiene su memoria cache.
 - b. Todos los procesadores comparten la misma memoria Cache.
 - c. Los procesadores poseen un tiempo de acceso diferente a cada localización de la memoria.
6. Los sistemas NUMA se diferencian de los sistemas UMA por:
 - a. Todos los procesadores comparten la misma memoria cache.
 - b. El tiempo de acceso al sistema varía de acuerdo a donde se encuentre localizado el acceso.
 - c. Cada procesador cuenta con su propio sistema de memoria principal.
7. Una característica de la arquitectura superescalar es:
 - a. La arquitectura superescalar solo puede ejecutar una instrucción por ciclo de reloj.
 - b. La arquitectura superescalar puede ejecutar más de una instrucción por ciclo de reloj debido a que son varios procesadores.
 - c. La arquitectura superescalar puede ejecutar más de una instrucción por ciclo de reloj debido a que posee elementos repetidos para proveer múltiples cauces.
8. De las siguientes cual no es una limitación del paralelismo:
 - a. Conflictos de recursos.
 - b. Dependencia total de la Frecuencia del sistema
 - c. Depende fuertemente del procedimiento





9. De los siguientes cual no se considera un sistema SIMD:
 - a. Procesador vectorial.
 - b. Procesador analógico
 - c. Procesador matricial.

10. Los sistema de tipo cluster se caracterizan por:
 - a. Ejecutan un único flujo de instrucciones
 - b. Procesan una secuencia de instrucciones y una secuencia de datos.
 - c. Utilizan un sistema de memoria débilmente acoplada

ANEXO: CLAVES DE LAS AUTOEVALUACIONES

N° 01		N° 02		N° 03		N° 04	
1	C	1	A	1	B	1	C
2	B	2	A	2	D	2	B
3	C	3	B	3	A	3	B
4	A	4	C	4	A	4	C
5	B	5	B	5	C	5	A
6	V	6	C	6	A	6	B
7	V	7	C	7	C	7	C
8	V	8	A	8	A	8	B
9	F	9	C	9	A	9	B
10	V	10	B	10	B	10	C
		11	B				



