



CENTRO DE ESTUDIOS FINANCIEROS

VIRIATO, 52	28010 MADRID	914 44 49 20
PONZANO, 15	28010 MADRID	914 44 49 20
G. DE GRÀCIA, 171	08012 BARCELONA	934 15 09 88
ALBORAYA, 23	46010 VALENCIA	963 61 41 99

www.cef.es

info@cef.es

Índice Tema 1

1. Introducción.
 - 1.1. Prehistoria.
 - 1.2. La Era Mecánica.
 - 1.3. Generaciones de Ordenadores.
 - 1.4. Arquitectura Von Neumann.
 - 1.5. Arquitectura de procesadores: CISC y RISC.
2. PDA o PALMTOP.
 - 2.1. Características físicas.
 - 2.2. Funcionalidades.
 - 2.3. Limitaciones
3. Tablet PC.
 - 3.1. Características físicas.
 - 3.2. Funcionalidades.
 - 3.3. Limitaciones.
4. Ordenadores personales.
 - 4.1. Placa Base.
 - 4.2. Procesador.
 - 4.3. Memoria RAM.
5. Equipos departamentales o miniordenadores.
 - 5.1. Definición.
 - 5.2. Características.
 - 5.3. Funcionalidades.

6. Mainframes o superordenadores.

6.1. Definición.

6.2. Características.

6.3. Funcionalidades.

6.3.1. Clasificación de superordenadores atendiendo a la implementación Hardware.

6.3.2. SISD.

6.3.3. SIMD.

6.3.4. MISD.

6.3.5. MIMD.

7. Unidades de medida del rendimiento de los ordenadores.



CENTRO DE ESTUDIOS FINANCIEROS

VIRIATO, 52	28010 MADRID	914 44 49 20
PONZANO, 15	28010 MADRID	914 44 49 20
G. DE GRÀCIA, 171	08012 BARCELONA	934 15 09 88
ALBORAYA, 23	46010 VALENCIA	963 61 41 99

www.cef.es

info@cef.es

TEMA 1

Tecnologías actuales de ordenadores: de los dispositivos de mano «Palmtop» a los superordenadores. Base tecnológica. Componentes, funcionalidades y capacidades.

1. INTRODUCCIÓN.

1.1. PREHISTORIA.

La Computación, y por tanto, las Ciencias de la Computación, tienen su origen en el cálculo, es decir, en la preocupación del ser humano por encontrar maneras de realizar operaciones matemáticas de forma cada vez más rápida y más fácilmente. Pronto se vio que con ayuda de aparatos y máquinas las operaciones podían realizarse de forma más rápida y automática.

El primer ejemplo que encontramos en la Historia es el ábaco, aparecido hacia el 500 AC en Oriente Próximo, que servía para agilizar las operaciones aritméticas básicas, y que se extendió a China y Japón, siendo descubierto mucho más tarde por Europa.

También es digno de señalar el conocido Mecanismo de Antikythera, recuperado en 1900, construido alrededor del año 80 AC, en la isla griega de Rodas, ubicada en el mar Egeo. Era un artefacto de cálculo astronómico con mecanismos de precisión. El usuario, por medio de una perilla, podía accionar un simulador en miniatura del movimiento del Sol, la Luna y varios planetas, teniendo a la vista la fecha en que se había dado, o se daría, tal combinación. Es tanta su sofisticación que ha sido llamado la primera computadora de Occidente.

Por otra parte, los matemáticos hindúes, árabes y europeos fueron los primeros que desarrollaron técnicas de cálculo escrito. El matemático árabe Al'Khwarizmi, alrededor del año 830 DC, escribe un libro de Aritmética, traducido al latín como *Algoritmi de numero Indorum*, donde introduce el sistema numérico indio (sólo conocido por los árabes unos 50 años antes) y los métodos para calcular con él. De esta versión latina proviene la palabra algoritmo.

1.2. LA ERA MECÁNICA.

A finales del siglo XVI y comienzos del XVII comienza lo que denominamos Era Mecánica, en la que se intenta que aparatos mecánicos realicen operaciones matemáticas de forma prácticamente automática. En 1610, John Napier (1550-1617), inventor de los logaritmos, desarrolló las Varillas de Napier, que servían para simplificar la multiplicación. En 1641, el matemático y filósofo francés Blaise Pascal (1623-1662), con tan sólo 19 años, construyó una máquina mecánica para realizar adiciones, la Pascalina, para ayudar a su padre. Por su parte, Gottfried Wilhelm Leibniz (1646-1716) propuso el sistema binario para realizar los cálculos, construyendo una máquina que podía multiplicar, en incluso teóricamente, realizar las cuatro operaciones aritméticas. Sin embargo, la tecnología disponible le imposibilita la realización de las operaciones con exactitud. No obstante, un estudiante alemán de la Universidad de Tubingen, Wilhelm Schickard (1592-1635) ya había construido una máquina de estas características entre 1623 y 1624, de la que hace unas breves descripciones en dos cartas dirigidas a Johannes Kepler. Por desgracia, al menos una de las máquinas quedó destruida en un incendio, y el propio Schickard murió poco después, víctima de la peste bubónica.

Los trabajos de Pascal y Leibniz tuvieron su continuación en 1727, cuando Jacob Leupold propuso algunas mejoras sobre el mecanismo de Leibniz. En 1777, Charles Mahon (1753-1816), Conde de Stanhope, construyó una máquina aritmética y otra lógica, esta última llamada Demostrador de Stanhope. En 1825, el francés Charles Xavier Thomas de Colmar diseña una máquina calculadora que posteriormente consigue comercializar con éxito.

Una mención muy especial requiere el desarrollo de un telar automático por el francés Joseph Jacquard (1752-1834), en 1801. En efecto, analizando las operaciones repetitivas que requería la producción de telas, este inventor imaginó conservar la información repetitiva necesaria bajo la forma de perforaciones en tarjetas. Estas perforaciones eran detectadas mecánicamente, asegurando el desplazamiento adecuado de las guías del hilado, pudiendo una sola persona tejer complicados patrones codificados en las perforaciones de las tarjetas.

Fue Charles Babbage (1791-18171) el que diseñó una verdadera máquina procesadora de información, capaz de autocontrolar su funcionamiento. Desesperado por los errores contenidos en las tablas numéricas de la época y dándose cuenta de que la mayoría de los cálculos consistían en tediosas operaciones repetitivas, este profesor de la Universidad de Cambridge proyecta e inicia la construcción de un nuevo tipo de calculadora. En 1821 presentó a la Royal Society una máquina capaz de resolver ecuaciones polinómicas mediante el cálculo de diferencias sucesivas entre conjuntos de números, llamada Máquina Diferencial. Obtuvo por ello la medalla de oro de la Sociedad en 1822.

Más tarde, Babbage empezó a trabajar en la Máquina Analítica, en cuya concepción colaboró directamente Ada Augusta Byron, Condesa de Lovelace, hija de Lord Byron. El objetivo perseguido era obtener una máquina calculadora de propósito general, controlada por una secuencia de instrucciones, con una unidad de proceso, una memoria central, facilidades de entrada y salida de datos, y posibilidades de control paso a paso, es decir, lo que hoy conocemos como programa. Ada Lovelace, a quien se reconoce como la primera programadora de la Historia, y en honor de quien se puso el nombre de Ada al conocido lenguaje de programación, ayudó a Babbage económicamente, vendiendo todas sus joyas, y escribió artículos y programas para la referida máquina, algunos de ellos sobre juegos. Sin embargo, este proyecto tampoco pudo realizarse por razones económicas y tecnológicas.

En el 1854, George Boole publica las leyes del pensamiento sobre las cuales son basadas las teorías matemáticas de Lógica y Probabilidad. Boole aproximó la lógica en una nueva dirección, redu-

ciéndola a una álgebra simple, incorporando lógica en las matemáticas. Comenzaba el álgebra de la lógica llamada Algebra Booleana. Su álgebra consiste en un método para resolver problemas de lógica que recurre solamente a los valores binarios 1 y 0 y a tres operadores: AND (y), OR (o) y NOT (no).

1.3. GENERACIONES DE ORDENADORES.

PRIMERA GENERACIÓN.

Para tabular el censo de 1890, el gobierno de Estados Unidos estimó que se invertirían alrededor de diez años. Un poco antes, Herman Hollerith (1860-1929), había desarrollado un sistema de tarjetas perforadas eléctrico y basado en la lógica de Boole, aplicándolo a una máquina tabuladora de su invención. La máquina de Hollerith se usó para tabular el censo de aquel año, durando el proceso total no más de dos años y medio. Así, en 1896, Hollerith crea la Tabulating Machine Company con la que pretendía comercializar su máquina. La fusión de esta empresa con otras dos, dio lugar, en 1924, a la International Business Machines Corporation (IBM).

Sin embargo, en el censo de 1910, el sistema de Hollerith fue sustituido por uno desarrollado por James Powers. En 1911 James Powers constituyó la Power's Tabulating Machine Company, convirtiéndose en el principal competidor de Hollerith.

En 1900, en el Congreso Internacional de Matemáticas de París, David Hilbert (1862-1943) pronunció una conferencia, de título «Problemas matemáticos», en la que proponía una lista de 23 problemas que estaban sin resolver (algunos todavía lo están).

Dos de estas cuestiones fueron: ¿es la matemática completa?, es decir, ¿puede ser demostrada o refutada cualquier sentencia matemática? y ¿es la matemática consistente?, es decir, ¿es cierto que sentencias tales como $0 = 1$ no pueden demostrarse por métodos válidos? En 1931, Kurt Gödel (1906-1978) fue capaz de responder a estas dos preguntas, demostrando que cualquier sistema formal suficientemente potente es inconsistente o incompleto.

Otra de las cuestiones era: ¿son las matemáticas decidibles? es decir, ¿hay un método definido que pueda aplicarse a cualquier sentencia matemática y que nos diga si esa sentencia es cierta o no? Esta cuestión recibió el nombre de *entscheidungsproblem*.

En 1936, Alan Turing (1912-1954) contestó a esta cuestión en el artículo «On Computable Numbers». Para resolver la cuestión, Turing construyó un modelo formal de computador, la Máquina de Turing, y demostró que había problemas tales que una máquina no podía resolver. Al mismo tiempo, en Estados Unidos contestaba a la misma cuestión Alonzo Church, basándose en una notación formal, que denominó cálculo lambda, para transformar todas las fórmulas matemáticas a una forma estándar. Basándose en estos resultados, entre 1936 y 1941, el ingeniero alemán Konrad Zuse (1910-1957), diseñó y construyó su serie de computadores electromecánicos binarios, desde el Z1 hasta el Z3. Sin embargo, estos computadores no tuvieron mucha difusión, ni siquiera dentro de su país, ya que el gobierno nazi nunca confió en los trabajos de Zuse.

En 1938, Claude Shannon (1916-) demostró cómo las operaciones booleanas elementales se podían representar mediante circuitos conmutadores eléctricos, y cómo la combinación de circuitos podía representar operaciones aritméticas y lógicas complejas. Además demostró cómo el álgebra de Boole se podía utilizar para simplificar circuitos conmutadores. El enlace entre lógica y electrónica estaba establecido.

Al desencadenarse la Segunda Guerra Mundial, la necesidad de realizar complicados cálculos balísticos y la exigencia de descodificar los mensajes cifrados del otro bando, impulsó el desarrollo de los computadores electrónicos de propósito general. El propio Turing fue reclutado en Bletchley Park, en Inglaterra, para descifrar los mensajes que encriptaba la máquina alemana Enigma, para lo que fue necesario construir la computadora Colossus.

En la Universidad de Harvard, Howard Aiken (1900-1973) en colaboración con IBM, empezó, en 1939, la construcción del computador electromecánico Mark I, en la que trabajó como programadora Grace Murray Hopper. Pero para cuando se terminó en 1944, ya habían aparecido las primeras computadoras totalmente electrónicas, que eran mucho más rápidas.

Por otro lado, en la Universidad del Estado de Iowa, entre 1937 y 1942, John Vincent Atanasoff (1903-1995) y Clifford Berry, diseñaron y construyeron la ABC (Atanasoff-Berry Computer). Terminada en 1942, fue la primera computadora electrónica digital, aunque sin buenos resultados y nunca fue mejorada. En 1941, John W. Mauchly (1907-1980) visitó a Atanasoff y observó de cerca su impresionante maquinaria, teniendo la oportunidad de revisar su tecnología. Más tarde, Mauchly y J. Presper Eckert, Jr (1919-1995), diseñaron y construyeron, entre los años 1943 y 1946, el computador eléctrico de propósito general ENIAC. Existe una gran controversia respecto a que Mauchly copiara muchas de las ideas y conceptos del profesor Atanasoff para construir la computadora ENIAC. En cualquier caso, en las últimas fases de su diseño y construcción aparece la importante figura de John Von Neumann (1903-1957), que actúa como consultor.

Von Neumann escribió en 1946, en colaboración con Arthur W. Burks y Herman H. Goldstine, *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*, que contiene la idea de Máquina de Von Neumann, que es la descripción de la arquitectura que, desde 1946, se aplica a todos los computadores que se han construido.

Con estos fundamentos, Eckert y Mauchly construyen en la Universidad de Manchester, en Connecticut (EE.UU.), en 1949 el primer equipo con capacidad de almacenamiento de memoria, la EDVAC. Eckert y Mauchly forman una corporación para construir una máquina que se pueda comercializar, pero, debido a problemas financieros, se vieron obligados a vender su compañía a la Remington Rand Corp. Trabajando para esta compañía fue que se concluyó el proyecto Univac, en 1951.

También por esta época Maurice Wilkes construye la EDSAC en Cambridge (Inglaterra) y F.C. Williams construye en Manchester (Inglaterra), la Manchester Mark I.

Estas máquinas se programaban directamente en lenguaje máquina, pero a finales de los 50, se produjo un gran avance en la programación, apareciendo los primeros lenguajes de alto nivel (LANs).

SEGUNDA GENERACIÓN.

A partir de mediados de los 50 y hasta entrados los 60 se trabaja con ordenadores de segunda generación. Desde un punto de vista tecnológico las válvulas de vacío son sustituidas por transistores. Los transistores fueron descubiertos en los primeros 50 en los laboratorios BELL de la empresa americana ATT, que son los laboratorios de investigación aplicada más importantes del mundo desde hace más de 50 años.

Como consecuencia de la utilización de transistores disminuyó mucho el tamaño y consiguientemente los costes y la necesidad de refrigeración.

La segunda generación busca nuevos clientes y nuevas aplicaciones y sus ordenadores empiezan a instalarse en las grandes corporaciones siendo dedicados a su gestión. La programación se simplifica notablemente ya que en los 50 aparecen los lenguajes simbólicos: primero el ensamblador que no era un lenguaje universal sino específico para cada máquina, y después los lenguajes simbólicos de alto nivel, como el FORTRAN o el COBOL, que tenían un carácter universal y eran válidos para cualquier máquina.

TERCERA GENERACIÓN (1969-1977).

A partir del circuito integrado, se producen nuevas máquinas, mucho más pequeñas y rápidas que las anteriores. Así, aparecen las IBM 360/91, IBM 195, SOLOMON (desarrollada por la Westinghouse Corporation) y la ILLIAC IV, producida por Burroughs, el Ministerio de Defensa de los EE.UU. y la Universidad de Illinois.

Seymour Cray (1925-1996) revoluciona el campo de la supercomputación con sus diseños: en 1964, el CDC 6600, que era capaz de realizar un millón de operaciones en coma flotante por segundo; en 1969, el CDC 7600, el primer procesador vectorial, diez veces más rápido que su predecesor.

En cuanto a los avances teóricos, a mediados de los 60, un profesor de Ciencias de la Computación, Niklaus Wirth, desarrolla el lenguaje PASCAL, y en Berkeley, el profesor Lotfi A. Zadeh, publica su artículo «Fuzzy Sets», que revoluciona campos como la Inteligencia Artificial, la Teoría de Control o la Arquitectura de Computadores.

En 1971, Intel introduce el primer microprocesador. El potentísimo 4004 procesaba 4 bits de datos a la vez, tenía su propia unidad lógico-aritmética, su propia unidad de control y 2 chips de memoria. Este conjunto de 2.300 transistores que ejecutaba 60.000 operaciones por segundo se puso a la venta por 200 dólares. Muy pronto Intel comercializó el 8008, capaz de procesar el doble de datos que su antecesor y que inundó los aparatos de aeropuertos, restaurantes, salones recreativos, hospitales, gasolineras...

A partir de aquí nacieron las tecnologías de integración a gran escala (LSI) y de integración a muy gran escala (VLSI), con las que procesadores muy complejos podían colocarse en un pequeño chip.

Sin embargo, hasta este momento, por motivos económicos, complejidad de uso y dificultad de mantenimiento, los computadores habían sido patrimonio de universidades, organismos militares y gubernamentales, y grandes empresas.

En 1975, Popular Electronics dedicó su portada al primer microcomputador del mundo capaz de rivalizar con los modelos comerciales, el Altair 8800.

CUARTA GENERACIÓN.

El Altair 8800, producido por una compañía llamada Micro Instrumentation and Telemetry Systems (MITS), se vendía a 397 dólares, lo que indudablemente contribuyó a su popularización. No obstante, el Altair requería elevados conocimientos de programación, tenía 256 bytes de memoria y empleaba lenguaje máquina. Dos jóvenes, William Gates y Paul Allen, ofrecerían al dueño de MITS, un software en BASIC que podía correr en el Altair. El software fue un éxito y, posteriormente, Allen y Gates crearon Microsoft.

Paralelamente, Steven Wozniak y Steven Jobs, también a raíz de ver el Altair 8800 en la portada de Popular Electronics, construyen en 1976, la Apple I. Steven Jobs con una visión futurista presionó a Wozniak para tratar de vender el modelo y el 1 de abril de 1976 nació Apple Computer. En 1977, con el lanzamiento de la Apple II, el primer computador con gráficos a color y carcasa de plástico, la compañía empezó a imponerse en el mercado.

En 1981, IBM estrena una nueva máquina, la IBM Personal Computer, protagonista absoluta de una nueva estrategia: entrar en los hogares. El corazón de esta pequeña computadora, con 16 Kb de memoria (ampliable a 256), era un procesador Intel, y su sistema operativo procedía de una empresa recién nacida llamada Microsoft.

En 1984, Apple lanza el Macintosh, que disponía de interfaz gráfico para el usuario y un ratón, que se hizo muy popular por su facilidad de uso.

SITUACIÓN ACTUAL.

Las tendencias dominantes de los últimos años de la Informática las podemos condensar en los siguientes puntos:

- Tecnología ULSI (Ultra Large Scale Integration).
- RISC (Reduced Instruction Set Computers).
- Multiprocesadores (computadores con varias Unidades Centrales de Proceso).
- Multicomputadores (sistemas de varios computadores conectados trabajando en común).
- Procesamiento distribuido y paralelo.
- Comunicación por red, boom de Internet.
- Soporte óptico en comunicaciones y almacenamiento de datos.

1.4. ARQUITECTURA VON NEUMANN.

Este científico matemático ocupa un lugar privilegiado en la historia de la computación debido a sus múltiples e importantísimos aportes a las computadoras de la primera generación. Nació el 28 de diciembre de 1903 en Budapest (Hungría), llegando a ser uno de los más brillantes matemáticos de la era de la computación.

Entre sus trabajos teóricos podríamos citar los relativos a lógica matemática, axiomática de teoría de conjuntos, espacios de Hilbert, álgebras de operadores, teoría de la medida, teoría ergódica, etc. Por otro lado, podrían merecer especial significación la calificación de aplicaciones, la fundamentación matemática de la mecánica, la creación de la teoría de juegos, o sus notables aportaciones a las teorías de sistemas y autómatas.

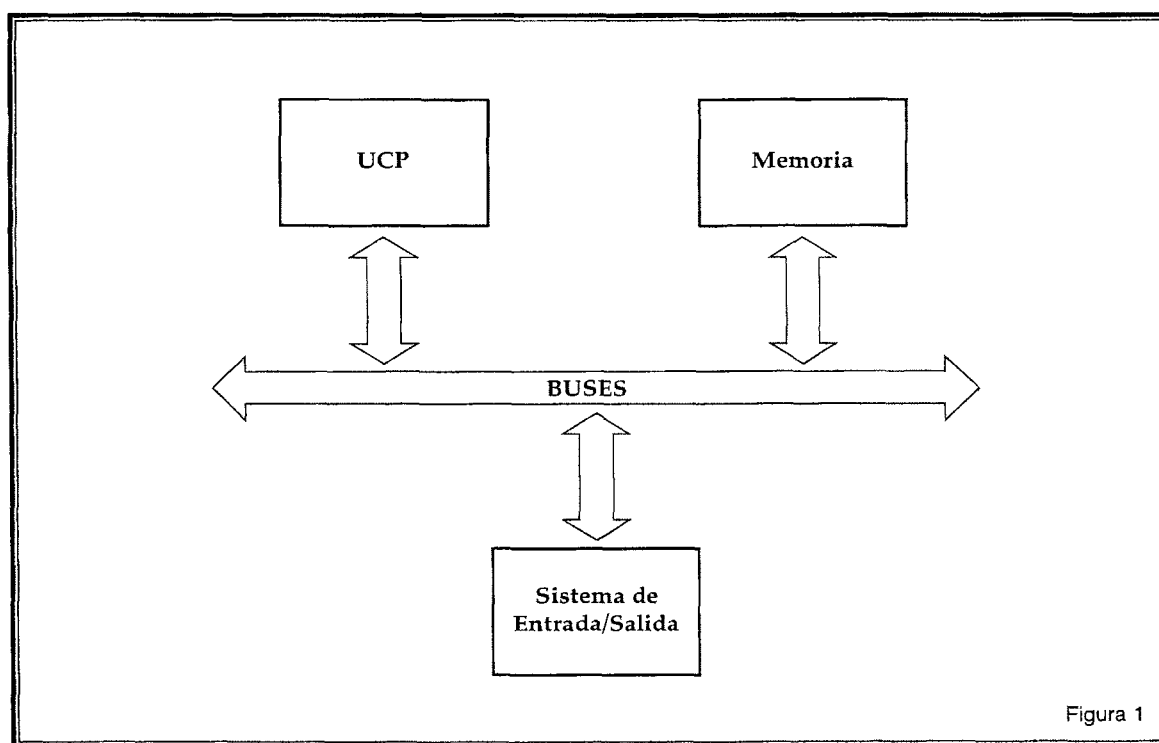
Durante la segunda guerra mundial fueron aprovechados sus conocimientos en hidrodinámica, balística, meteorología, teoría de juegos y estadísticas. En 1944 contribuyó en forma directa en los diseños de fabricación de computadoras, asesorando a Eckert y John Mauchly, creadores de la ENIAC y que construyeron además la UNIVAC en 1950. Durante esa década trabajó como consultor para la IBM colaborando con Howard Aiken para la construcción de la computadora Mark I de Harvard.

En 1946, en colaboración con Arthur W. Burks y Herman H. Goldstine, Von Neumann escribió uno de los artículos más influyentes en la moderna historia de los computadores: «Preliminary Discussion of the Logical Design of an Electronic Computing Instrument». Las ideas que contiene este artículo, que de forma conjunta se conocen con el nombre de Máquina de Von Neumann o Arquitectura Von Neumann, han proporcionado los fundamentos para la construcción y el desarrollo de todos los computadores hasta el momento.

Se supone que las radiaciones a las que estuvo expuesto durante sus trabajos sobre la bomba atómica fueron la causa de su muerte por cáncer el 8 de febrero de 1957 en Washington DC.

A este autor se debe la denominada «Arquitectura Von Neumann». Se trata de una arquitectura genérica de un ordenador, con independencia de la realización concreta (PC, PC-compatible, Macintosh, Sun, Silicon...). Data de los primeros ordenadores (años 40), pero la abstracción sigue siendo válida hoy. Inicialmente eran unidades diferenciables físicamente.

Un único bus permite la comunicación entre la UCP, la memoria y el sistema de entrada / salida, tal y como se detalla en la siguiente figura:



Las Unidades funcionales que aparecen en este modelo son las siguientes:

1. Unidades de entrada/salida (E/S).
2. Unidad central de proceso (CPU) ó (UCP).
3. Memoria.

Unidades de entrada/salida:

- Entrada:

Recibe datos o instrucciones desde el exterior, convirtiendo la información exterior en información digital (procesable por el ordenador), posiblemente mediante algún convertidor analógico/digital.

Ejemplos: teclado, ratón, cámara, micrófono, lector de tarjetas, lector de código de barras, OCR, escáner.

En entornos industriales o médicos: sensores en una planta o en la UCI de un hospital.

- Salida:

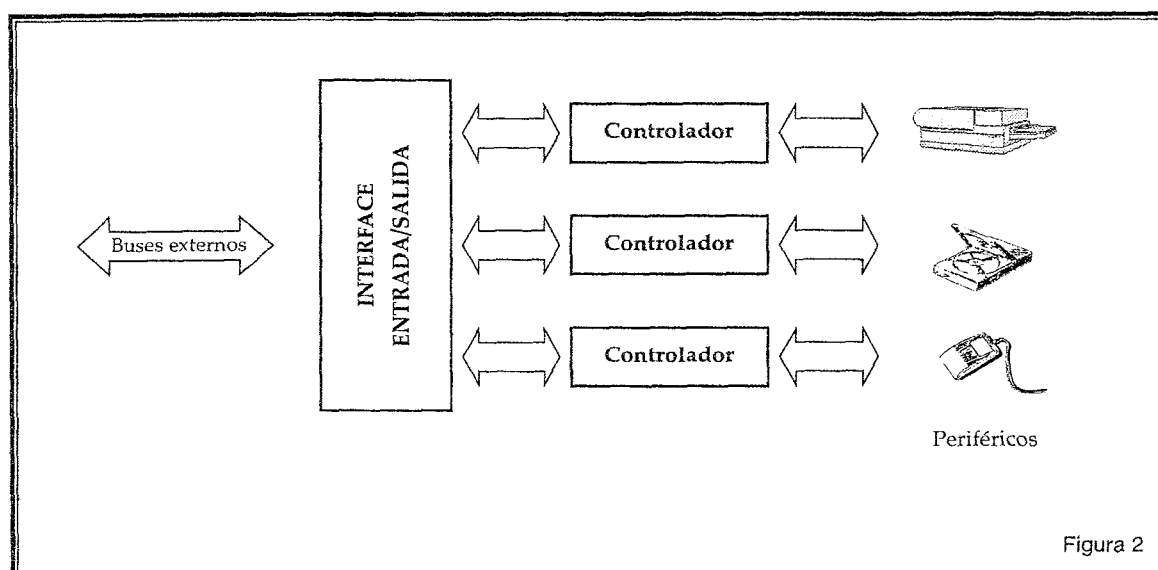
Muestra resultados generados por el ordenador. Transforma información digital a un código interpretable por los humanos: caracteres, gráficos, sonidos...

Ejemplos: pantalla, impresora, altavoces...

En entornos industriales: actuadores, que reciben la información del ordenador a través de un convertidor digital/analógico.

Las tareas de comunicación de los ordenadores con su entorno suele estar gestionada a través de controladores.

Funcionamiento:



Hay que tener en cuenta que:

- Una dirección de E/S equivale a un puerto.
- La velocidad del periférico es distinta de la velocidad CPU, por lo que son necesarias técnicas de sincronización.

CPU

- Unidad de control, UC:

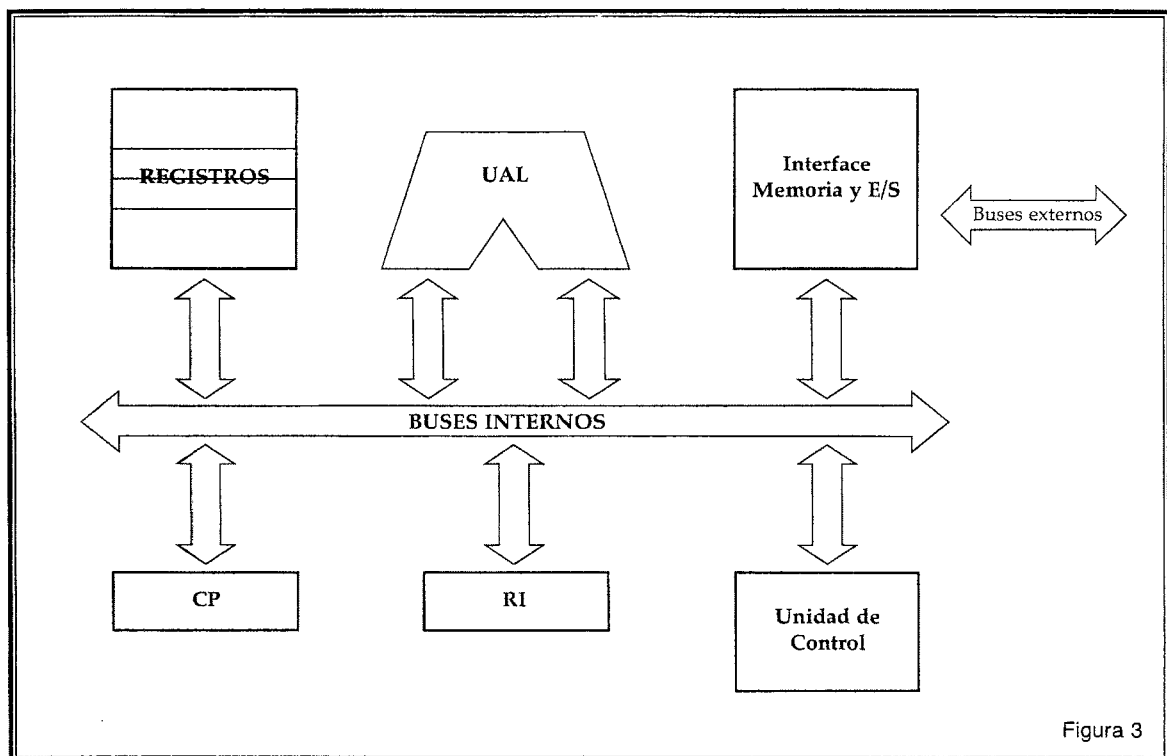
Ejecuta (lleva a cabo) las instrucciones de los programas, interpretando las señales de estado. Para ello utiliza las señales de control.

- ALU:

Unidad aritmético lógica: realiza las operaciones aritmética y lógicas. Comunica sus resultados y su estado a la UC.

- Registros de la CPU. Elementos de memoria asociados a la CPU y que sirven para el almacenamiento temporal de información: datos y/o instrucciones.

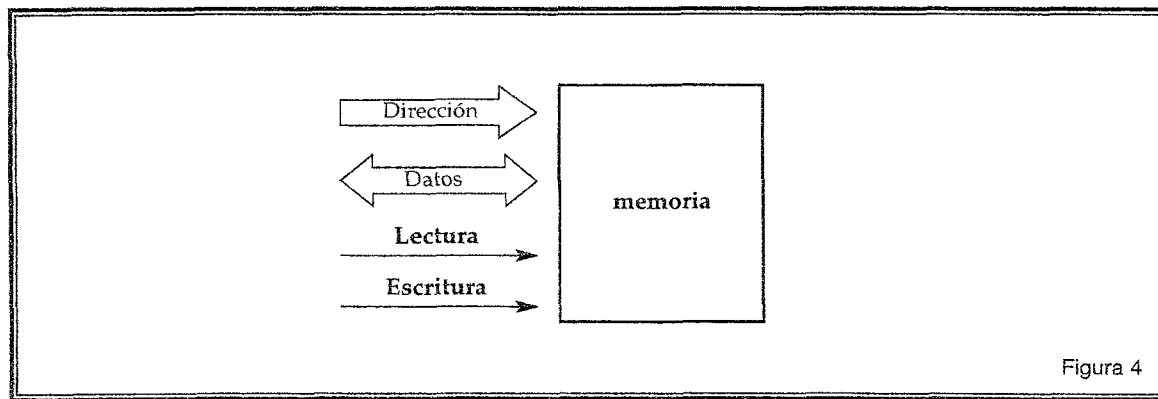
Su funcionamiento se puede esquematizar en la siguiente figura:



MEMORIA:

- Memoria principal, MP: almacena datos e instrucciones que van a ser inmediatamente ejecutados. Gran velocidad de acceso a los datos y pequeña capacidad.

Consta de ROM y RAM, aunque existen otros tipos de memoria relacionados con MP (como la caché). La memoria RAM puede almacenar tanto datos como programas (instrucciones), concepto de programa almacenado. Su funcionamiento se puede esquematizar en la siguiente figura:



- Memoria secundaria o masiva, MS: almacena datos e instrucciones de forma permanente, cuando no van a ser ejecutados. Capacidad grande, pero velocidad de acceso a los datos menor. Ejemplos: discos (magnéticos u ópticos), cintas magnéticas, etc.

Los datos en la memoria MP y MS tendrán longitud de n palabras.

Las unidades de memoria más importantes son las siguientes:

UNIDADES DE CAPACIDAD	
1 KB Kilobyte = 2^{10} Bytes (1024 bytes)	1 PB Petabyte = 2^{10} T
1 MB Megabyte = 2^{10} K	1 EB Exabyte = 2^{10} P ó 2^6 bytes
1 GB Gygabyte = 2^{10} M	1 ZB Zettabyte = 2^{10} E
1 TB Terabyte = 2^{10} G	1 YB Yottabyte = 2^{10} Z

CONEXIÓN ENTRE UNIDADES-BUSES:

Las unidades del ordenador se comunican a través de buses. Un bus se trata de una o varias líneas -eléctricas u ópticas- a través de las cuales se comunican las distintas unidades de un ordenador.

Los buses son de distintos tipos:

- Bus de datos: transmite datos entre entradas, salidas, memoria y la CPU.
- Bus de direcciones: transmite direcciones entre la CPU y la memoria.
- Bus de control: controla las línea (selección de línea, aviso, reloj) a nivel eléctrico para ejecutar las instrucciones.

Las unidades de los ordenadores suelen denominarse:

- CPU (Unidad Central de Proceso), compuesta por la UC (Unidad de Control) y la ALU (Unidad Aritmético y Lógica).
- Ordenador central = CPU + MP (memoria principal).
- Periféricos: entrada + Salida ó E/S, Memoria Secundaria.

1.5. ARQUITECTURA DE PROCESADORES: CISC Y RISC.

La arquitectura de un computador viene definida por el repertorio de instrucciones, el tipo de direccionamientos y el tipo de operandos que es capaz de ofrecer al compilador. Existen dos alternativas de arquitecturas para solucionar los problemas derivados del salto semántico. Por un lado, hay toda una familia de microprocesadores conocida como CISC (Complex Instruction Set Computer). Mientras que por otra parte, se ha desarrollado otra clase de microprocesadores llamados RISC (Reduced Instruction Set Computer). Estas dos formas de arquitectura no sólo se diferencian en la complejidad del conjunto de instrucciones, sino que poseen filosofías distintas para solucionar otros problemas referentes al diseño de un computador. Sin embargo, ambas poseen juegos de instrucciones basadas en registros. Es decir, basan el almacenamiento interno de los operandos en los registros de la CPU (otras alternativas consisten en utilizar una pila o un acumulador).

CISC:

Históricamente la memoria ha tenido un elevado coste y en consecuencia se han ido buscado arquitecturas que optimizasen el uso de la misma. Las arquitecturas de computadores que aparecieron en primer lugar fueron CISC. Por tanto, no es de extrañar que una de las mayores preocupaciones de esta familia haya sido siempre el ahorro de memoria. Como hemos comentado, los lenguajes de alto nivel se han impuesto en la programación al lenguaje básico del computador, el ensamblador.

Así pues CISC es una arquitectura que persigue la programación fácil y al mismo tiempo un uso eficiente de memoria.

- Ayudar al compilador es ayudar al programador. Resulta evidente que hoy en día sería imposible atender a la demanda de requerimientos y prestaciones software que hay en el mercado si no existiesen los lenguajes de alto nivel. Sin embargo, debido al salto semántico entre estos y el ensamblador, la escritura de los compiladores se vuelve tremendamente complicada. La escritura sencilla de un compilador depende mucho del repertorio de instrucciones del que se disponga en cada arquitectura. Para CISC la simplificación de los compiladores es obvia. Cuantas más instrucciones máquina den soporte directo a sentencias de lenguajes de alto nivel, más fácil será escribir un buen compilador. Además, poder ejecutar una instrucción de alto nivel en solamente una de bajo nivel permite una optimización del hardware, debido a que cada instrucción está diseñada para un paso específico y se puede conocer mejor cuáles son sus características. Pero no sólo basta con dar soporte hardware a instrucciones de alto nivel para simplificar la escritura de los compiladores. También es necesario disponer de un gran número de modos de direccionamiento, ya que de esta manera el compilador podrá decidir fácilmente cuál utilizar en cada caso.
- Se ahorra memoria y los programas son rápidos si son cortos. Otra de las preocupaciones de la filosofía CISC es que la utilización de los lenguajes de alto nivel tiene una influencia negativa sobre el tamaño de la memoria requerida por las aplicaciones. Para paliar este problema los compiladores tienen que optimizar el código ensamblador resultante para una ejecución efi-

ciente. En base a esto, CISC siempre ha intentado acercar el código máquina a los lenguajes de alto nivel, evitando así montones de instrucciones equivalentes que ocuparían más memoria. Un programa corto posee mayores prestaciones que otro de mayores dimensiones. En primer lugar, porque debemos tener en cuenta que cuanto más rápido se ejecuten los programas, mayor rendimiento se obtiene del computador. Los programas cortos son más rápidos porque, además de tener que ejecutar menor número de instrucciones, se deben captar menos bytes de la memoria para ejecutarlo. En segundo lugar, porque la gestión de memoria, sobre todo en un entorno que se utilice paginación y memoria virtual, se ve simplificada por la disminución de los fallos de página.

- La compatibilidad entre generaciones atrae al usuario. Uno de los aspectos más importantes en el mundo de la informática es la compatibilidad entre distintas versiones de computadores y de software. CISC intenta buscar flexibilidad de programación en los lenguajes de alto nivel para distintas plataformas, permitiendo la portabilidad entre diferentes sistemas. Es interesante de cara al usuario poder disponer de computadores orientados a la optimización del cálculo en áreas de aplicación diferentes (cálculo científico, cálculo gráfico...), pero que puedan ejecutar los mismos programas sin necesidad de tener que recompilarlos. La compatibilidad entre unas versiones y otras ofrece al usuario seguridad a la hora de invertir su dinero en la adquisición de un sistema informático. Por otro lado, es de vital importancia para los diseñadores de computadores poder depurar, modificar y actualizar las instrucciones que brinda una arquitectura de la forma más sencilla y fiable posible. Como veremos más adelante, CISC consigue estos propósitos a partir de la utilización de la microprogramación.

Una arquitectura CISC se ve identificada, como hemos visto, con el deseo de simplificar los compiladores y de aumentar las prestaciones del computador. Pero CISC no sólo es eso, además hay que resaltar todo un conjunto de características significativas. Históricamente casi todas las arquitecturas han sido CISC, influenciadas básicamente por el ahorro de memoria (recordemos que hasta ahora era de un alto coste), por el afán de construir supercomputadores que fueran capaces de hacerlo todo por sí solos y por un intento de mantener la compatibilidad de una familia a otra. Actualmente los computadores CISC más conocidos son la familia 8086 de Intel (que se extiende hasta el Pentium actual, aunque este último empiece a presentar características RISC), pero históricamente ha habido otros como: IBM 3090, NSC 32016, familia MC68000 de Motorola, VAX.

Como hemos comentado en la explicación de la filosofía CISC, esta familia de arquitecturas posee un gran repertorio de instrucciones. Su tamaño se debe tanto al elevado número de operaciones que permite realizar, como al número de direccionamientos que es capaz de soportar. Las instrucciones del repertorio separan del nivel ensamblador para acercarse a las sentencias de los lenguajes de alto nivel. Se trata de instrucciones que realizan operaciones complejas. El formato de las instrucciones es muy irregular. La longitud de las palabras varía dependiendo de la instrucción de la que se trate y sobre todo del modo de direccionamiento que se esté utilizando. Las instrucciones necesitan de múltiples ciclos de instrucción para su ejecución. Además, no todas precisan del mismo número de ellos, sino que tardan más o menos dependiendo de la instrucción en concreto. Por ejemplo, se necesitan ciclos de reloj extra en aquellos casos en los que se necesite información adicional para la ejecución (como leer algún dato de memoria). Los modos de direccionamiento varían desde el registro-registro hasta el de memoria-memoria, incluyendo modos especiales para la indexación en arrays.

Debido al amplio repertorio de instrucciones y a la complejidad de las mismas se hace casi impensable el uso de una unidad de control cableada. En lugar de ello el sistema está microprogramado. Esta característica facilita el diseño del procesador, ahorrando la infinidad de líneas de control que supondría hacer la unidad de control cableada y la enorme cantidad de transistores que sería necesario incorporar. Cada instrucción es ejecutada mediante una serie de microinstrucciones almacenadas en

una memoria, típicamente una ROM, ubicada en el circuito del procesador. La captación de instrucciones es aproximadamente 10 veces más rápida desde una ROM que desde la memoria principal y no supone ningún problema acceder constantemente al microprograma. Por ello, los diseñadores incluyen microcódigo para tantas instrucciones como les sea posible.

La microprogramación permite diseñar, depurar y modificar fácilmente instrucciones realmente complicadas. Pero no sólo eso, cambiando algunos trozos de microcódigo disponemos de un computador específico para tareas de investigación, gráficos, videojuegos...

Los registros son mayoritariamente de uso dedicado. Dentro de un procesador existen dos tipos de registros. Los registros de propósito general son aquellos que el procesador puede utilizar para almacenar datos temporales, variables locales... Por su parte, los registros de uso dedicado son aquellos que están reservados para tareas muy específicas (almacenar el contador de programa, el puntero de la pila...). El número de registros de propósito general es reducido en las arquitecturas CISC. Por un lado, se debe a que el elevado número de modos de direccionamiento provoca que casi todo el tránsito de datos se produzca de memoria a memoria. Por otro lado, la mayor parte del espacio del chip se utiliza para la decodificación y la ejecución, así como para el almacenamiento del microcódigo, dejando poco espacio para estos registros. El compilador que se use ha de ser capaz de maximizar el rendimiento de los pocos registros de propósito general que hay, con el fin de lograr una ejecución mucho más eficiente del programa. En cambio, sí son abundantes los registros de uso dedicado que controlan el tránsito de datos, y el estado del procesador. Algunos de estos registros son utilizados para almacenar el puntero de la pila, para realizar la gestión de las interrupciones y para almacenar los códigos de condición. Por último, cabe destacar que los computadores basados en arquitecturas CISC suelen incorporar una memoria intermedia rápida (caché) para agilizar cálculos y para almacenar datos temporales muy usados.

RISC:

Mientras se desarrollaban los procesadores CISC, se realizaron estudios detallados sobre las características de la ejecución de las instrucciones generadas por los lenguajes de alto nivel. La mayoría de ellos fueron realizados por David A. Patterson y Tanenbaum. Los aspectos más relevantes de sus estudios se centraron en: determinar qué instrucciones son las que realmente ejecuta la CPU, el tipo de operandos y su interacción con la memoria y el secuenciamiento de las instrucciones. Asimismo, es muy importante darse cuenta de que realmente es un compilador el que genera el código máquina y no el programador. Como consecuencia, la filosofía RISC está muy encauzada a ayudar al diseño de compiladores optimizados.

Todos los estudios realizados para analizar el comportamiento de los programas han señalado que el 80 por 100 de las operaciones que se realizan en un programa descrito con un lenguaje de alto nivel utilizan, sólo, un 20 por 100 de las operaciones disponibles en el repertorio de instrucciones. De hecho, las instrucciones que más importancia tienen son las de asignación y las sentencias condicionales típicas *if* y *loop*. Esto implica, por un lado, que el movimiento de datos es de vital importancia, y por otro que se debe optimizar el secuenciamiento de instrucciones (comparaciones y saltos) necesarios para realizar las operaciones condicionales. Sin embargo, no sólo no hay que encontrar las sentencias más utilizadas, sino aquellas operaciones descritas en lenguajes de alto nivel que provocan la ejecución de más instrucciones en lenguaje máquina y, como consecuencia, sean más lentas.

Todos estos estudios revelan que la mayoría de las referencias que se dan en un programa se hacen a variables escalares simples. Además, estas variables están muy bien localizadas por ser la mayoría de ellas locales al procedimiento que las referencia. No sólo son locales las variables escalares, sino que las referencias a estructuras más complejas como arrays y matrices precisan de una referencia

previa a un índice o puntero que también suele ser local al procedimiento. Otro aspecto relevante que se puede extraer de estos estudios es que cada instrucción referencia en promedio 0,5 operandos de memoria y 1,4 registros. Esto nos hace pensar que no es tan necesario disponer de modos de direccionamiento complejos que permitan operar directamente con posiciones de memoria, ya que la proporción de referencias a registros es mucho más elevada. Así pues, es muy importante poder disponer de un acceso y almacenamiento rápido de los operandos, sobre todo para aquellos que son locales dentro de un procedimiento. Esta es una de las razones básicas que motivan que las arquitecturas RISC también sean conocidas como arquitecturas load/store.

Es importantísimo poder realizar un diseño que optimice las llamadas y retornos a procedimientos y funciones, debido a la gran cantidad de tiempo que consumen. En primer lugar, debemos dilucidar el número de parámetros y de variables con los que trabaja un procedimiento. Los estudios realizados por Tanenbaum y el equipo de RISC de Berkeley coincidieron en que la aparición de nuevas palabras en la llamada a un nuevo procedimiento era muy pequeña y las referencias seguían siendo, en su mayoría, locales. Asimismo, el número de parámetros que se pasaban en cada llamada no superaba los seis en un 98 por 100 de las veces. En segundo lugar, se debe tener en cuenta el nivel de anidamiento al que se suele llegar en una aplicación. Este aspecto es extremadamente importante en los lenguajes de alto nivel, ya que la mayoría de ellos dan soporte a la programación estructurada. Según los estudios del grupo de Berkeley, un programa suele permanecer estable dentro de una amplitud de profundidad de invocación bastante pequeña. Esto supone un apoyo más a la idea de que las referencias a operandos están muy localizadas.

Como ya hemos comentado, no es el programador quién genera el código máquina de un programa. Esta tarea se deja a cargo de los compiladores. Una de las máximas de un diseño RISC es ofrecer una arquitectura que ayude a un diseño fácil y eficiente de los mismos, puesto que cuanto mejores sean estos, más eficaces serán los programas compilados en tamaño y velocidad. El trabajo del compilador se divide en diversas fases en las que transforma expresiones de un nivel de abstracción de-terminado a otro menor. Las decisiones de optimización se dan en cada una de esas fases sin saber con detalle cómo será el código resultante. Así pues, la dificultad en el diseño de los compiladores no reside en que deben traducir instrucciones complejas, sino que aparece en la traducción global de programas grandes, complejos y con un gran número de interacciones. Las optimizaciones más típicas son las siguientes:

- Integración de procedimientos: las llamadas a procedimientos suponen una carga de tiempo elevada. Se debe reducir tal carga sustituyendo la llamada por el procedimiento en sí.
- Eliminación de las subexpresiones comunes: debe ser posible sustituir dos instancias de un mismo cálculo por una copia.
- Propagación de constantes: el compilador debe ser capaz de sustituir todas aquellas variables a las que se les asigna una constante por la constante.
- Reducción de la altura de la pila: se debe reorganizar el árbol de la expresión para poder evaluarla de una forma más eficiente.
- Propagación de copia: el compilador debe reconocer aquellas expresiones que, aún estando situadas dentro de un bucle, no cambian. En tales casos debería ser capaz de extraerlas, siempre y cuando no sean susceptibles de ser cambiadas por la CPU u otros dispositivos.
- Optimización de bucles: el compilador debe reconocer aquellas expresiones que, aún estando situadas dentro de un bucle, no cambian. En tales casos debería ser capaz de extraerlas, siempre y cuando no sean susceptibles de ser cambiadas por la CPU u otros dispositivos.

- Gestión de registros: es muy importante que se reserven registros para guardar aquellos datos a los que se accede con más frecuencia. De esta manera se reduce el tráfico a memoria y se consigue mayor velocidad en las operaciones.
- Reducción de la complejidad: se deben poder reemplazar aquellas operaciones caras en tiempo de ejecución por otras más sencillas que realicen la misma tarea. Por ejemplo, si accedemos a la posición [a,b] de un array bidimensional de dimensión [A,B], la posición se calcula como: $\text{inicio_array} + [(b \cdot A) + a] \cdot \text{tamaño_dato}$. Se podría simplificar esto si, trabajando en la misma fila, si se guardase en un registro el valor $(b \cdot A) + a \cdot \text{tamaño_dato}$, ahorrando tiempo de acceso.
- Optimización del desplazamiento de saltos: el compilador debe escoger el desplazamiento de salto más corto que consiga el objetivo.

Los requerimientos que todo diseñador de compiladores espera de la arquitectura de una máquina y en los que RISC basa su forma de ayuda son:

- Regularidad: las operaciones, los tipos de operandos y los modos de direccionamiento deberían ser ortogonales. Es decir deberían ser independientes unos de otros, permitiendo cualquier combinación de estas tres características del sistema. De este modo, se facilita la optimización de código por parte del compilador.
- Proporcionar primitivas: lo importante no es aportar operaciones complejas a nivel de arquitectura, ya que este tipo de instrucciones son en muchos casos inutilizables. Por otro lado, acercar demasiado una arquitectura a un determinado lenguaje de alto nivel impide la generación de código eficiente para otro distinto.
- Simplificar compromisos entre alternativas: un diseñador de compiladores debe decidir qué secuencia de instrucciones máquina es la más apropiada para cada fragmento de programa. Pero esta decisión no es nada fácil porque influyen características relacionadas con la caché y la segmentación.
- Proporcionar instrucciones que interpreten todas las cantidades que se conocen en tiempo de compilación como constantes. Es muy importante que un buen compilador interprete en tiempo de compilación todas aquellas expresiones que se pueden conocer antes de la ejecución.

ESTRUCTURA RISC.

Las arquitecturas RISC no sólo se caracterizan por tener un repertorio de instrucciones restringido, sino que engloban una gran cantidad de aspectos que la diferencian de forma drástica a la opción CISC. Algunos computadores basados en la arquitectura RISC son el RISC I, el Motorola 88000, el MIPS R4000, el IBM 801.

- RISC es una arquitectura load/store.

El número de referencias por instrucción en un lenguaje de alto nivel es elevado y la mayoría de las instrucciones sólo requieren un simple flujo de datos. Como resultado, la gran mayoría de los direccionamientos en las instrucciones RISC son registro-registro. Es decir, se cargan los operandos en los registros mediante una operación de tipo load, se realizan las operaciones pertinentes entre los registros y los resultados se almacenan en memoria mediante una instrucción de tipo store. A este modelo de arquitectura, exclusiva de los RISC, se le conoce

como load/store. Evidentemente, los registros del procesador son los que brindan los tiempos de acceso más cortos debido tanto a la tecnología de acceso como al número de bits necesarios para direccionarlos. En una arquitectura de tipo load/store lo ideal sería poder tener todos los operandos que se necesiten en la ejecución de un programa ubicados en registros. Pero entonces el coste del hardware se vería incrementado notablemente. Se necesita una estrategia para ubicar aquellos operandos a los que se accede con más frecuencia y reducir el tráfico registro-memoria generado en las operaciones tipo load y store. Una estrategia consistiría en confiar al compilador la maximización del uso de los registros. La otra estrategia, más acorde con la filosofía RISC, es disponer de una cantidad elevada de registros para mantener ubicadas las variables durante un periodo de tiempo mayor. Siguiendo la segunda estrategia, cabe considerar que con cada llamada de procedimiento, el aspecto local de las variables cambia y se debe realizar un paso de parámetros. Si todos los registros fuesen visibles para cada procedimiento, se deberían salvar en memoria muchos de ellos con cada llamada. Por esta razón se utiliza un conjunto de numerosos registros de propósito general, al que se llama fichero de registros. Se define una ventana o marco de visibilidad para cada procedimiento. Es decir, cada procedimiento posee asociado un marco de registros del procesador para almacenar sus operandos. En cada momento sólo es visible una ventana de registros que se direccionan como si sólo existieran ellos. Los marcos se dividen en tres secciones que suelen ser de la misma longitud:

- Registros de parámetros: estos registros contienen el conjunto de parámetros que se pasan del procedimiento padre al que está en curso. A través de ellos un procedimiento hijo se puede comunicar con su padre sin que exista un flujo real de datos entre ambos.
- Registros locales: los registros locales sirven para almacenar aquellos operandos que el compilador haya seleccionado con fines optimizadores, debido sobre todo al gran número de asignaciones en las que se ven involucrados.
- Registros temporales: los registros temporales sirven para realizar el paso de argumentos desde un procedimiento a sus hijos. Del mismo modo que ocurriría con los registros de parámetros, no es necesario un flujo real de datos entre los dos procedimientos.

Tal como podemos observar, existe solapamiento entre diversas ventanas, siempre y cuando se realicen llamadas y exista paso de argumentos. Sin embargo, el número de procedimientos activos en un sistema y el nivel de anidamiento son imprevisibles y es posible disponer de un número infinito de ventanas. Por ello, el aspecto real del fichero de registros es el de un buffer circular. Sólo se mantienen marcos o ventanas para aquellos procedimientos que sean más recientes. Los más antiguos se han de guardar en memoria para, posteriormente, cargarlos de nuevo en el fichero de registros. Por último, es interesante anotar que en muchas ocasiones los computadores RISC incorporan una pequeña memoria caché dedicada exclusivamente para instrucciones, consiguiendo así una mejora adicional.

- RISC es una arquitectura basada en la sencillez.

La arquitectura RISC apuesta por la sencillez del diseño de todas sus características: tipo de operaciones, modos de direccionamiento y formato de instrucciones.

En las arquitecturas RISC no se realizan operaciones complicadas como se suele hacer en las CISC. Su objetivo es proporcionar operaciones básicas que se usen con gran frecuencia, dejando de lado otras más sofisticadas pero de un uso tan restringido que no merece la pena contemplar. Los modos de direccionamiento son pocos y sencillos. Debido a la gran utiliza-

ción de las variables locales y a la estrategia de adoptar un fichero de registros, el modo de direccionamiento más importante es el efectuado a registro. También se utilizan otros, como el de desplazamiento y el relativo al contador de programa. El resto de los modos de direccionamiento que posean una mayor complejidad se pueden obtener de los simples vía software. El formato de las instrucciones es sencillo y regular. Existen pocos formatos distintos (3 o 4 como máximo). El tamaño de las instrucciones es idéntico en todos ellos (32 o 64 bits suele ser lo habitual) y están alineadas en los límites de una palabra. Por último, las posiciones de los campos dentro de una instrucción son fijos, especialmente el de código de operación. Debido a la sencillez de las instrucciones, es posible ejecutar una instrucción en cada ciclo de máquina. La reducción del tiempo necesario en la ejecución tiene varias consecuencias. La primera de ellas es que es posible la reorganización de la ejecución de las instrucciones por parte del compilador de una forma sencilla. En segundo lugar, es factible poder ejecutar varias instrucciones simultáneamente, mediante una técnica denominada Pipeline. En tercer lugar, no es necesario disponer de un microcódigo para poder ejecutar una instrucción, sino que pueden estar directamente cableadas en la CPU sin que suponga grandes complicaciones de circuitería. Por último se produce una reducción del tiempo invertido en los ciclos de diseño, permitiendo la aplicación de las tecnologías más vanguardistas en el desarrollo de los circuitos y la posibilidad de producir grandes mejoras en las prestaciones de una versión respecto a la siguiente. Las arquitecturas RISC simplifican también la estructura del procesador, reduciendo notablemente el consumo de potencia y la superficie de circuito integrado necesaria para construirlo. Esta reducción de tamaño permite ubicar una gran cantidad de registros en el procesador y se aprovecha para desarrollar otras funciones tales como: unidad para el procesamiento en punto flotante, unidad de administración de memoria y funciones de control de la memoria caché.

Ventajas de RISC:

- Un procesador CISC gasta el 80 por 100 de su tiempo ejecutando solamente el 20 por 100 de sus instrucciones.
- La ejecución en RISC es más rápida que en CISC.
- Los compiladores comerciales no usan todas las instrucciones del procesador CISC.

Principales fabricantes de procesadores:

FABRICANTE	PROCESADOR	TENDENCIA
SUN	SPARC	RISC
SILICON GRAPHICS	MIPS	RISC
MOTOROLA	POWER	RISC
HEWLETT PACKARD	PA	RISC
DEC	ALPHA	RISC
INTEL	PENTIUM	CISC – RISC

2. PDA O PALMTOP (PERSONAL DIGITAL ASSISTANT)

Los PDA o «Personal Digital Assistant» son en realidad pequeños ordenadores de bolsillo tremendamente potentes y capaces de realizar numerosas funciones más allá de las de mera agenda electrónica, si bien este nombre, junto con los de ordenador de bolsillo (pocket pc) o palmtop son más utilizados por el público. Aunque inicialmente la concepción básica era la de una mera agenda electrónica, estos dispositivos han evolucionado mucho y en la actualidad constituyen una extensión misma del ordenador personal, que podremos sincronizar con éste y que nos permitirá llevar con nosotros la información necesaria. Igualmente, las nuevas tecnologías móviles nos permiten que gracias a nuestro PDA podamos realizar acciones tales como consultar el correo electrónico o visitar un sitio web en cualquier momento y en cualquier lugar.

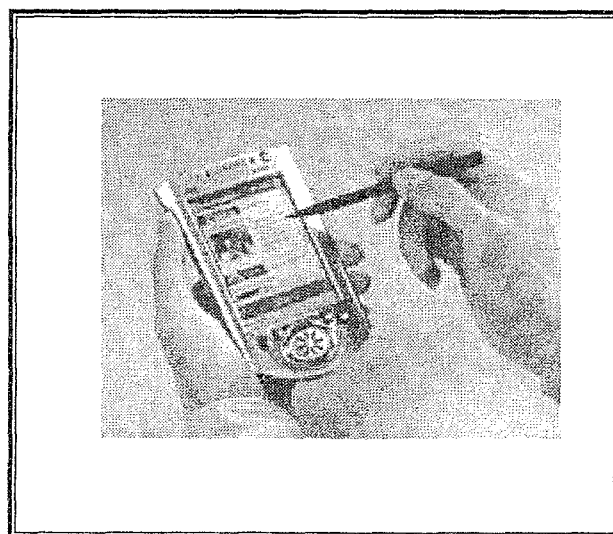
Pero más allá de las funciones y software con las que viene equipado el PDA, lo que lo hace verdaderamente potente es la posibilidad de personalización casi ilimitada que nos confiere pues podremos cargar las aplicaciones que más nos interesen para configurar de este modo un dispositivo según las exigencias del cliente. En el momento actual, las PDA se dividen en dos grandes grupos, los Pocket PC y los dispositivos con sistema operativo Palm.

El primero de estos grupos está constituido por aquellos modelos que vienen equipados con la versión móvil del popular Windows de Microsoft, en tanto que los segundos son los modelos con sistema operativo Palm OS, una versión creada por y para estos pequeños accesorios y que ha ganado adeptos entre buena parte de la comunidad de usuarios. Sin embargo, y aunque el funcionamiento interno de ambos sistemas es radicalmente distinto, ambos convergen hacia un interfaz muy similar y que está dominado por la sencillez de manejo y la completa accesibilidad, permitiéndonos utilizarlo en cualquier momento y en cualquier lugar para satisfacer cualquier necesidad.

En suma, una PDA u ordenador de bolsillo, es un potente dispositivo, a disposición del usuario para configurarlo en la forma que más sencillo y más útil resulte y cuyas notas básicas serían las siguientes. El trabajo no siempre se acaba cuando salimos de la oficina. Y cuando la «oficina» es un edificio, un terreno o una clase, necesita una solución informática que se pueda adaptar a sus necesidades y a su ubicación.

2.1 CARACTERÍSTICAS FÍSICAS:

- Reducido tamaño.
- Ligeros.
- Pantalla táctil (accesible mediante un pequeño puntero).
- Multimedia: altavoz, micrófono y grabadora de voz.
- Conexión de periféricos: ranura «Secure Digital» para dispositivos de almacenamiento externo, conexión para módulos de expansión (teclado, mouse...).



- Conexión inalámbrica: Módulo de comunicación Bluetooth, infrarrojos. Wi-fi.
- Funcionamiento con baterías.
- Capacidad de almacenamiento por encima de los 64 MB que se puede ampliar mediante el uso de tarjetas de memoria Flash.
- Fabricantes: Nec, HP, Casio, Palmote, Sony ...
- Microprocesadores específicos: Intel® XScale™ de 400 MHz, Texas Instruments OMAP 1510 144MHz, etc.

2.2. FUNCIONALIDADES.

- Sistemas operativos: Windows Pocket PC, PalmOS, Linux Familiar.
- Pensados para trabajar de forma inalámbrica: Wi-fi o BlueTooth.
 - Acceder a LAN's e Internet.
 - Descargar aplicaciones.
 - Acceder a recursos remotos.

2.3. LIMITACIONES.

- Potencia de computación.
- Capacidad de almacenamiento.
- Duración de las baterías.
- Periféricos.
- Capacidad de visualización.
- Comunicaciones.
- Software específico.

3. TABLET PC.

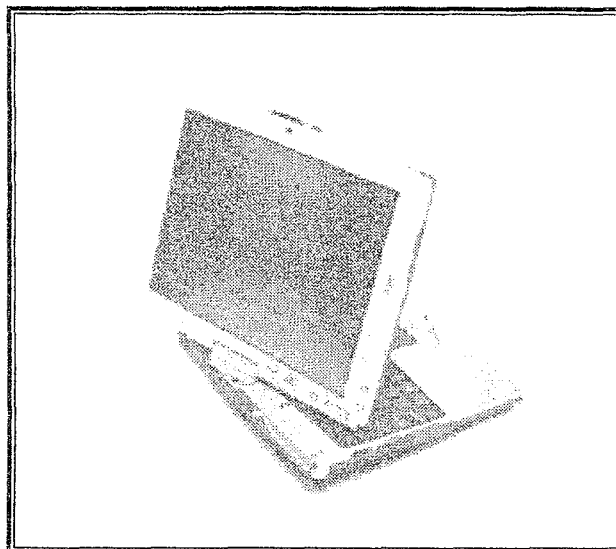
El Tablet PC, ultradelgado, ligero y fácil de transportar, combina la potencia de un ordenador portátil o de sobremesa con la comodidad de un PDA. Gracias a la tecnología móvil más reciente y a una total compatibilidad con Windows, el Tablet PC le permite trabajar como el usuario decida.

Se define Microsoft Tablet PC, como un portátil mucho mas funcional y un bloc de notas con posibilidades inimaginables reunidos en uno solo. Utilizando el sistema operativo Microsoft Windows XP

Tablet PC Edition, que es una versión superior de Windows XP Professional, estamos ante un PC portátil de última generación, con posibilidad de ejecutar cualquier aplicación Windows incluidas las de la suite Office. Si además le añadimos reconocimiento de voz y lápiz digital, los límites están en nuestra imaginación. Microsoft cuenta con numerosos Partners dedicados a la distribución de hardware: ACER, Compaq, Fujitsu, NEC, Toshiba y ViewSonic, entre otros, para la fabricación de Tablet PC donde se ejecuta Windows XP Tablet PC Edition. Tablet PC se presenta en diversos modelos. Algunos serán «Tablet PC Pizarrón», sin un teclado integrado, mientras que otro modelo es «Tablet PC Convertible» con un diseño más convencional con forma de concha como una computadora portátil. Los convertibles vienen equipados con pantallas giratorias por lo que pueden convertirse también en Pizarras. Cualquiera de los modelos es muy ligero -de 1,5 a 2 kg de peso- y ofrecerá todas las características que necesita el usuario de portátiles, incluyendo batería de larga duración, alta calidad de presentación, posibilidad de creación de redes y soporte integrado para periféricos gracias a los puertos USB y los puertos del servidor de seguridad. Además, ofrecen soporte para bases de acoplamiento y una serie de periféricos externos.

3.1. CARACTERÍSTICAS FÍSICAS.

- Ordenador ultraportátil común al que se le han añadido prestaciones de manejo y de forma.
- Usan procesadores móviles de bajo consumo como son los Pentium 3 Mobile y los Crusoe de Transmeta. Se están introduciendo los Intel Centrino.
- Memorias SODIMM (Small Outline Dual Inlinne Memory Module) convencionales.
- Discos duros de 2.5" como los de cualquier otro portátil.
- Pantalla TFT táctil
- Nuevas formas de control mediante voz y escritura manual.



3.2 FUNCIONALIDADES.

- Sistemas operativos: Windows XP Tablet Edition.
- Pensados para trabajar de forma inalámbrica:
 - Acceder a LAN's e Internet.
- Aplicaciones compatibles con los entornos PC convencionales.
- Fabricantes: Fujitsu, Compaq, Toshiba, Acer...
- Otra propuesta en este sentido son Smartdisplay. «Pantallas inteligentes» pensadas para ofrecer los contenidos del ordenador personal en cualquier punto de una ubicación.

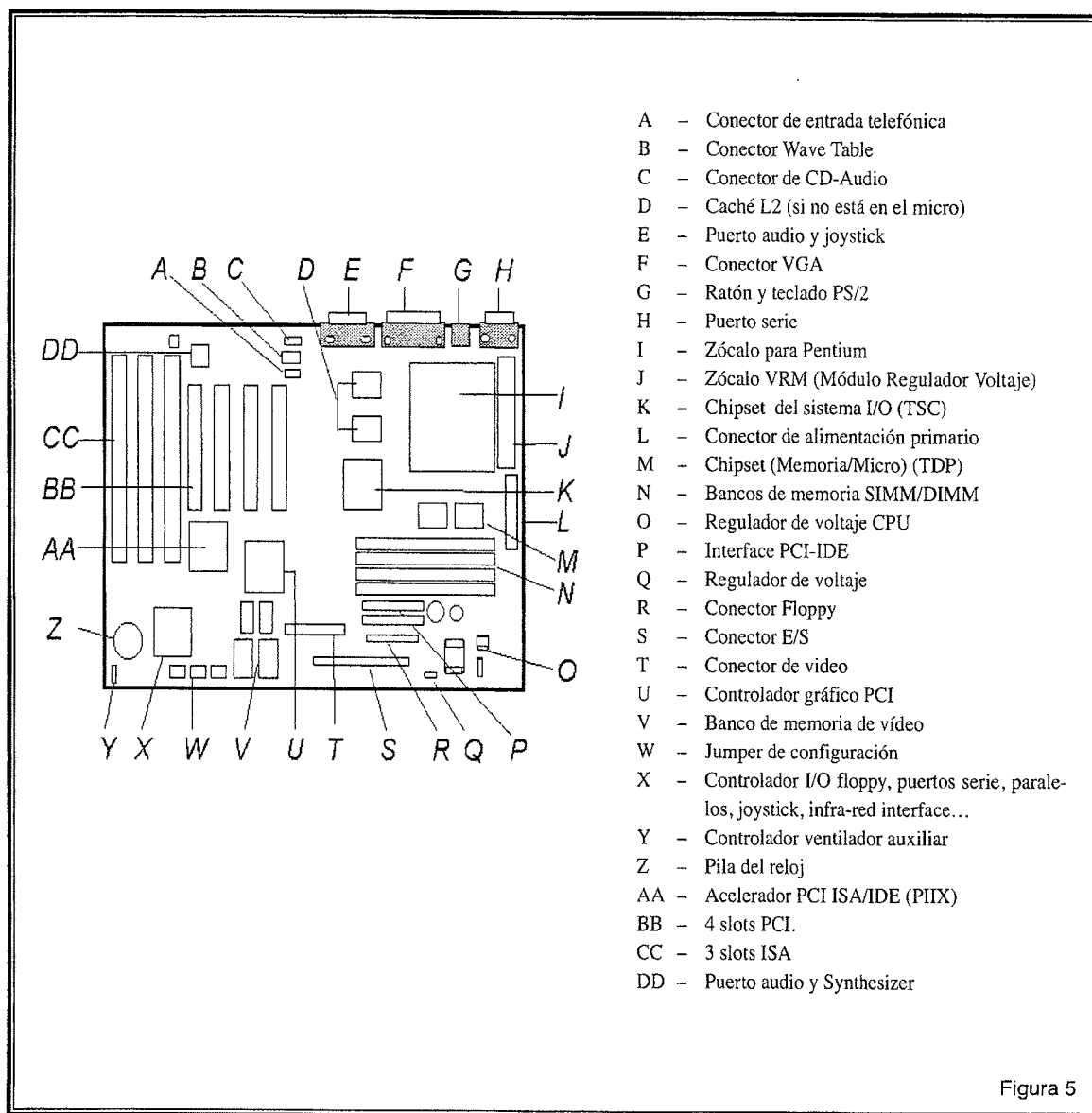
3.3. LIMITACIONES.

- Potencia de computación.
- Capacidad de almacenamiento.
- Duración de las baterías.

4. ORDENADORES PERSONALES (PC)

4.1. PLACA BASE (MOTHER BOARD).

Esquema Placa Base (Placa ATX):



Los elementos que componen una placa base son: zócalo de microprocesador, zócalos de memoria, diversos conectores internos y externos, ranuras de expansión y una serie de chips que efectúan diversas tareas de un PC. La BIOS (Basic Input Output System) es un software específico almacenado en un chip de memoria EPROM (memoria de sólo lectura que puede ser borrada y programada), cuyo contenido permanece inalterable al apagar el PC, aunque pueda reprogramarse y es el primer software en ejecutarse en el proceso de arranque de una placa base. La batería, una pequeña pila, viene incorporada en todas las placas base y su función básica es mantener la alimentación eléctrica del reloj de tiempo real (RTC), así como diversos parámetros sobre el disco duro, y otros de usuario, que son utilizados por la BIOS. Las placas base actuales traen un zócalo donde se inserta el microprocesador. Actualmente se utilizan los zócalos ZIF (Zero Insertion Force); no es necesario usar la fuerza para insertar el microprocesador y los zócalos de 478 pines (Intel). Cuando se monta un microprocesador es necesario configurar algunos parámetros como la frecuencia base de su funcionamiento y el multiplicador, que es el factor por el que multiplicada la frecuencia base y nos da la frecuencia real de funcionamiento. Esta configuración, que anteriormente se hacía vía física mediante la utilización de «Jumpers», ahora se hace vía software.

La memoria RAM es la fuente principal de almacenamiento de información con la que se comunica continuamente el microprocesador. La RAM es una memoria volátil de lectura y escritura, lo cual implica que su contenido sólo se mantiene mientras el PC esté encendido. Están en actualidad los módulos de memoria DIMM de memoria SDRAM y los RIMM de memoria RDRAM. La SDRAM (Synchronous Dynamic RAM) es un tipo de memoria que opera de manera síncrona con el bus de memoria, empleando la misma señal del reloj, con tiempos de acceso entre 8 y 12 nanosegundos.

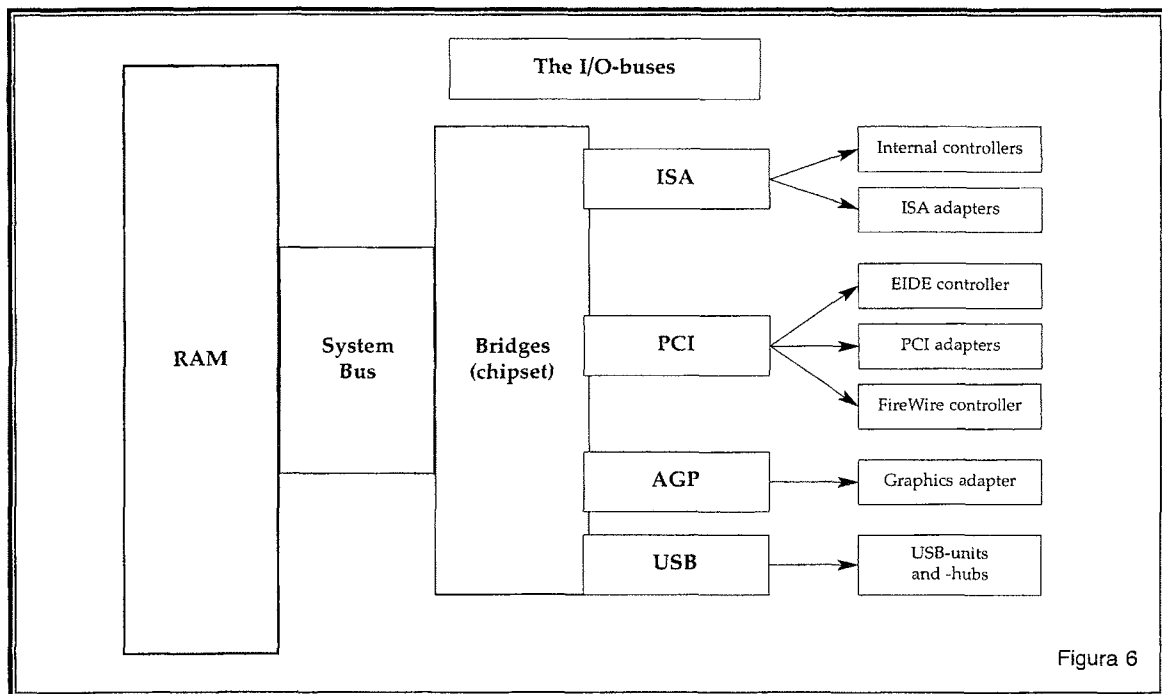
Los módulos DIMM de memoria SDRAM pueden funcionar a la misma frecuencia que la del bus del sistema (400, 800 Hz). La cantidad de memoria que puede admitir una placa base depende del zócalo que tenga, en un zócalo DIMM admite los módulos de memoria existentes en el mercado desde 64 MB hasta 1 Giga.

La memoria caché es una memoria mucho más rápida; su función es almacenar los últimos datos e instrucciones leídas de la memoria RAM, lo que permite que el siguiente acceso a una posición de memoria ya leída anteriormente se realice más rápido, pues el acceso tiene lugar en la memoria caché bastante más rápida. La memoria caché actual se compone de dos niveles (L2) y (L1) y ambas vienen integradas dentro del propio microprocesador. La L2 es de mayor tamaño y más lenta que la L1.

Todas las placas base permiten la conexión de cualquier periférico adicional en forma de tarjeta. Ello se debe a la utilización de las ranuras de expansión. Actualmente el bus PCI (Peripheral Component Interconnect) se ha convertido en el bus local de 64 bits más extendido en las placas base para PC. Una característica esencial del bus PCI es que es independiente del sistema, cualquier periférico desarrollado para un PC puede funcionar en cualquier otro sistema que incorpore arquitectura y conectores PCI. El PCI es un bus que oferta altas prestaciones por sus características especiales:

- Modo Burst. Una vez que al bus se le proporciona una dirección inicial puede transferir múltiples conjuntos de datos.
- Soporte Bus Mastering. Es la capacidad que tienen los dispositivos PCI de tomar el control del bus y realizar transferencias directas de información.

Una evolución actual es el bus PCI Express para plataformas domésticas, un desarrollo conjunto de fabricantes como Intel, Microsoft, DELL, HP e IBM. El objetivo de este bus es conseguir la unificación de todas las tarjetas de expansión de la placa base con un mismo tipo de bus.



El AGP (Accelerated Graphics Port) es un bus desarrollado por INTEL para uso exclusivo de las tarjetas gráficas. Permite alcanzar una transferencia máxima de 2 Gbits/s en modo 8x. Una clave para explicar la diferencia de rendimiento entre los dos tipos de bus citados es que AGP utiliza la técnica pipeline lo cual permite a la tarjeta gráfica AGP pueda llevar a cabo varias peticiones a la memoria principal sin tener que esperar a recibir los resultados de la petición anterior. Otro punto clave consiste en la llamada memoria AGP, que es una parte de la memoria RAM del sistema que se asigna dinámicamente para uso exclusivo de la tarjeta gráfica AGP.

La placa base de un ordenador es el dispositivo sobre el que se montan los demás componentes del PC, tales como el microprocesador, las diferentes tarjetas de expansión y la memoria. La función principal de la placa base es la de servir de vía de comunicación entre los citados componentes, proporcionando las líneas eléctricas necesarias y las señales de control para que todas las transferencias de datos se lleven a cabo de manera rápida y fiable.

En la arquitectura de un PC existen varios caminos para la información, siendo el centro neurálgico el microprocesador, ya que es el componente que maneja y procesa toda esta información. Podemos diferenciar tres canales de comunicación principales en una placa base:

- Entre el microprocesador y la memoria (RAM y cache).
- Entre el microprocesador y las tarjetas de expansión (ISA, PCI, AGP).
- Entre el microprocesador y los diferentes dispositivos de almacenamiento.

El chipset de la placa base integra todas estas funciones de control. La velocidad con que se mueven los datos en el interior de un PC está muy estrechamente relacionada con el chipset integrado en la placa base, así como el resto de la electrónica asociada. Desde la arquitectura PCI Intel fijó como estándar una frecuencia máxima de comunicación de 66 Mhz entre microprocesador, memoria y caché. La frecuencia de funcionamiento del microprocesador y de otros componentes como la memoria

es un múltiplo de esos 66 Mhz. Por otro lado, la velocidad con la que se mueve la información a través del bus PCI es de 33 Mhz, justo la mitad de los 66 Mhz fijados como base. Esta frecuencia se obtiene mediante un circuito divisor. Actualmente (abril 2004) las distintas frecuencias base de las placas han aumentado, situándose entre los 400 y los 800 Mhz de las últimas placas para Pentium 4.

Las diferentes funciones lógicas que suele integrar cualquiera de los chipsets actuales son las siguientes:

1. Soporte para el microprocesador. Una de las principales funciones del chipset es la detección correcta del microprocesador y el pleno soporte de todas sus funciones. Esta es una de las principales razones de la rápida evolución de los chipsets: nuevos procesadores cada vez más rápidos necesitan nuevos chipsets que les proporcionen un soporte completo. Cada chipset se diseña pensando en un procesador o familia de procesadores concretos. Igualmente, el chipset es el responsable directo de que la placa base soporte mas de un microprocesador, en el caso de placas base duales o con mas de dos microprocesadores.
2. Controlador de memoria (MMU, Memory Management Unit). Gestiona la memoria RAM del sistema y en general todo el subsistema de memoria, incluidos los diferentes niveles de memoria caché.
3. Controlador IDE/ATAPI para discos duros y otros dispositivos de almacenamiento que cumplan con el estándar IDE/ATAPI. Esta parte del chipset es la encargada de controlar los dos conectores IDE que habitualmente suelen integrar todas las placas bases actuales. Directamente relacionados con esta función están los modos de transferencia. El que una placa base soporte los recientes modos de transferencia UltraDMA es responsabilidad directa del chipset.
4. Control de los periféricos y del bus de E/S. Las placas base actuales disponen de una serie de buses, principalmente PCI y AGP. El chipset es el responsable de la gestión de los buses PCI y de ofrecer el soporte para el bus gráfico AGP. Algunas de las denominaciones de los chipsets históricos de Intel están en línea directa con esta función, con denominaciones como PCIset y AGPset. Esta función también incluye el soporte para nuevas tecnologías, como USB 2.0 o IEEE 1394.
5. Controlador de Interrupciones. Gestiona todo el sistema de interrupciones del PC.
6. Reloj de Tiempo Real (RTC). Mantiene la hora del sistema.
7. El soporte para la gestión de energía. Todos los chipsets actuales soportan una serie de funciones para gestión y ahorro de energía eléctrica
8. Controlador de Acceso Directo a Memoria (DMA, Direct Memory Access). Permite el acceso directo a la memoria a determinados dispositivos, sin pasar por el microprocesador, lo que agiliza el rendimiento de ciertas operaciones con dispositivos específicos como los discos duros. El DMA es controlado por una parte del chipset denominada controlador de DMA. Igualmente, el driver es el que soporta la función de arbitraje de bus (bus mastering), que es una mejora del DMA que permite que un dispositivo tome directamente el control del bus del sistema para llevar a cabo las transferencias de datos.
9. Controlador de infrarrojos (IrDA). Controla la conexión de dispositivos que funcionen mediante rayos infrarrojos.
10. Controlador de teclado. Controla toda la actividad y el funcionamiento del teclado.
11. Controlador PS/2. Para el control de teclados y ratones con este formato.

Además de INTEL existen otros fabricantes de chipsets como:

- AMD. Entre sus productos tenemos AMD-760 para sus microprocesadores Athlon XP.
- VIA Technologies. Entre sus más recientes desarrollos se encuentran el chipset P4X400 para Pentium 4 y el KT400, este ultimo destinado a los más recientes modelos de microprocesadores Athlon XP de AMD.
- SiS (Silicon Integrated Systems). Su principal chipset para Pentium 4 es el SiS 651. Entre sus desarrollos también se encuentra el SiS 745 para AMD Athlon XP.
- Ali (Acer Laboratories Inc). Responsable de los chipsets Aladdin P4 para Pentium 4 y del M1687 para AMD Athlon XP.
- NVidia. Modelos nForce, nForce2 y nForce3 Pro, todos ellos para microprocesadores de AMD.

PUERTOS Y CONECTORES DE UNA PLACA BASE.

- Conectores externos. Dos puertos serie, controlados por un chip UART (controlador serie) de alta velocidad, compatible con el chip UART 16550 original de National Semiconductor. Estos dos puertos se suelen denominar COM1 y COM2 respectivamente. El COM1 se usa habitualmente para la conexión de ratones serie, mientras que el COM2 queda libre para la conexión de dispositivos serie tales como módems externos. Existen conectores serie externos de tipo DB9 y DB25 (de 9 y 25 patillas, respectivamente).
- Un puerto paralelo multimodo para la conexión de dispositivos paralelo, generalmente impresoras y en menor medida escáneres y otros dispositivos paralelo. El término multimodo hace referencia a que el puerto es capaz de soportar tres modos de funcionamiento característicos: SPP (Single Paralell Port), ECP (Extended Capabilities Port) y EPP (Enhanced Paralell Port). Éstos suelen ser los tres modos habituales de transmisión del puerto paralelo, en orden creciente de prestaciones. El conector es de tipo hembra con 25 pines agrupados en dos filas.
- Varios puertos USB (Universal Serial Bus). Tienen forma estrecha y rectangular y permiten la conexión en caliente de dispositivos que cumplan este estándar. Actualmente la mayoría de placas base soporta la especificación USB 2.0.
- Puertos IEEE 1394b (Firewire). Permiten la conexión en caliente de dispositivos que cumplan este estándar de alta velocidad. Actualmente los principales dispositivos IEEE 1394b son sistemas de video digital (DV) y unidades de almacenamiento externas.
- Puertos PS/2, uno para teclado y otro para ratón. Ambos son conectores de tipo mini-DIN de seis patillas. Este suele ser el tipo habitual de conectores para ratón y teclado en las actuales placas base ATX.
- Puerto para juegos, en el que habitualmente se suelen conectar dispositivos como palancas o mandos de juegos (joysticks y gamepads), o dispositivos de audio, tales como teclados MIDI.
- Conectores de audio, generalmente para clavijas de tipo jack estéreo, siendo los más habituales los de entrada y/o salida de línea (line in/line out), entrada de micrófono (mic in) y salida de altavoces (speaker out).
- Conector VGA para la tarjeta grafica: es un conector estándar para tarjeta grafica. Consta de 15 pines agrupados en tres filas.

Conectores internos.

- Conector para disquetera. Donde se inserta la banda de cable para datos de la disquetera.
- Conectores IDE. Para la conexión de dispositivos IDE, principalmente discos duros y lectores de CD-ROM.
- Conectores para el refrigerador del microprocesador. Denominados generalmente Fan Power.
- Conector para arranque desde red (WakeOn-LAN).
- Conector para arranque desde línea telefónica mediante módem (WakeOn-Ring). Permite encender y apagar remotamente nuestro equipo a través de la línea telefónica mediante un módem que soporte esta función.
- Conector para módulo de infrarrojos (irDA).
- Controlador de Entrada/Salida: similar al chipset, es un chip que integra, la mayoría de funciones de entrada y salida de información de la placa base. Dentro de este chip se suelen integrar, entre otros, los siguientes elementos: una UART (Universal Asynchronous Receiver Transmisor, Receptor/Transmisor Asíncrono Universal), el controlador de puertos serie, el controlador del puerto paralelo, la controladora de la disquetera. Muchos chips de E/S incluyen además el reloj de tiempo real y el controlador de teclado.

4.2 PROCESADOR.

Esquema Genérico Pentium:

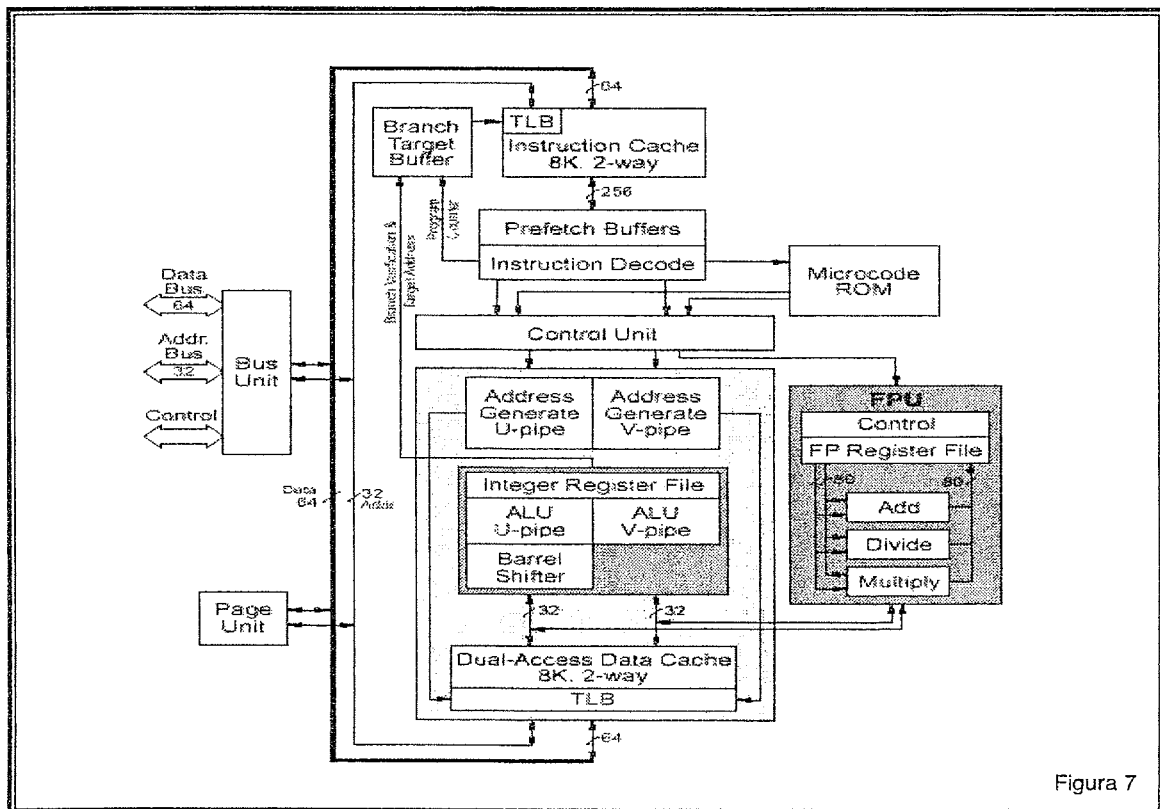


Figura 7

Conocida técnicamente como Unidad Central de Proceso, la CPU (Central Processing Unit) es el cerebro del ordenador y la encargada del procesamiento de todas las instrucciones de programas y dispositivos. El microprocesador es, sin duda, uno de los elementos claves para medir la potencia de nuestro PC, aunque no el único, ya que la capacidad real de un PC es una mezcla de las prestaciones conjuntas de todos sus componentes, especialmente de algunos fundamentales como el disco duro, la memoria, la arquitectura de la placa base y, por supuesto, el microprocesador. La calidad del microprocesador va a determinar en gran medida la fiabilidad y la estabilidad de nuestro PC durante su funcionamiento habitual. El microprocesador alojado en nuestro sistema va a ser el factor que determinara el tipo de placa base necesaria para ser integrada en el PC.

El PC es un dispositivo digital binario, lo que viene a indicar que trabaja únicamente con ceros y unos, manipulándolos a través de lo que se denominan funciones lógicas y produciendo una determinada salida. Existen diferentes funciones lógicas, también llamadas booleanas: AND, OR, XOR... Cada función lógica es diferente de las demás y producirá diferentes resultados en función de las entradas que se le suministren. Los posibles resultados de una función lógica se especifican mediante las denominadas tablas de verdad. Estas funciones son implementadas por los dispositivos digitales mediante las denominadas puertas lógicas, dispositivos digitales basados generalmente en transistores de tamaño microscópico.

En el caso de los dispositivos digitales, los ceros y los unos se implementan en la práctica mediante niveles de tensión. Generalmente, un voltaje positivo se identifica como un uno, y un voltaje nulo (tierra) se identifica como un cero. El hardware que se encarga de todo esto se le conoce genéricamente como conmutador digital, y en la práctica puede tener varias implementaciones. Los conmutadores digitales han sufrido una constante evolución a lo largo de los últimos tiempos: relés, válvulas de vacío, hasta el desarrollo de lo que hasta el día de hoy ha sido la base de la electrónica y la microelectrónica: el transistor, un pequeño conmutador que permite el manejo de ceros y unos, las dos señales del sistema binario con el que funciona la inmensa mayoría de los ordenadores y dispositivos digitales. El transistor fue desarrollado por los laboratorios Bell en 1948 y sus ventajas, frente a los sistemas anteriores, son enormes: es mucho más pequeño, más fiable y con un consumo eléctrico mucho menor.

Inicialmente, el transistor era un componente que se podía integrar dentro de un circuito electrónico, como un elemento más. Hoy en día, existe una gran cantidad de tipos de transistores de propósito específico. Una de las mayores revoluciones de la electrónica fue en su momento el desarrollo del circuito integrado, consistente en un determinado número de transistores fabricados todos en una única pieza e interconectados entre sí. A los circuitos integrados se les conoce más frecuentemente como chips. El primer circuito integrado fue desarrollado por Texas Instruments en 1959 y contenía seis transistores en una misma superficie de material semiconductor.

Para la fabricación de los chips se emplean materiales muy específicos: los denominados semiconductores. La mayoría de los materiales existentes en la naturaleza conducen totalmente la electricidad o bien son aislantes. Sin embargo, existen determinados materiales que sólo conducen la electricidad bajo determinadas condiciones. De ahí que se les denomine semiconductores. De entre todos ellos, el más empleado es el silicio. Otro semiconductor muy conocido es el arseniuro de galio (Ga-As), más difícil de encontrar en la Tierra y empleado solo en la fabricación de chips más específicos.

Con el tiempo, se fueron desarrollando las técnicas de integración de transistores en circuitos integrados, por lo que aparecieron diferentes escalas de integración. Una de ellas es LSI (Large Scale Integration), consistente en la creación de circuitos integrados que previamente habían sido contruidos a partir de múltiples componentes discretos. Los dispositivos que usan esta técnica generalmente

integran varios cientos de transistores. El paso siguiente fue VLSI (Very Large Scale Integration), con la que se desarrollaron circuitos que podían contener varios millones de transistores. Esta es la tecnología empleada en la fabricación de los actuales microprocesadores.

Cuando echamos un vistazo a un microprocesador, en realidad lo que vemos no es el microprocesador en sí, sino el encapsulado externo que lo envuelve y que contiene los contactos necesarios que le permiten comunicarse con el resto de componentes ensamblados en la placa base. El núcleo interno del microprocesador es un rectángulo de silicio alterado mediante procesos químicos, con una superficie cada vez más pequeña, debido a los continuos avances en el proceso de integración. Este núcleo contiene millones de puertas lógicas en forma de transistores.

Durante el proceso de fabricación de los chips, y en concreto de los microprocesadores, el silicio es contaminado (dopado) con otra sustancia, lo que introduce impurezas que le confieren su capacidad de semiconducción. Además, el silicio es cristalizado en grandes obleas o cristales que posteriormente son divididas en secciones rectangulares más pequeñas de las que finalmente saldrán los chips. Concretamente, una oblea de silicio puede dar lugar a más de cien microprocesadores con la tecnología existente actualmente. Más tarde se pasa a la fase de implantación de la lógica del microprocesador. Inicialmente, el microprocesador, una vez cortado de la oblea, es un trozo de material sin una función concreta. Después, durante una fase posterior, se implanta la lógica que lo dotará de todas sus funciones de cálculo. Este diseño se hace inicialmente a nivel lógico, es decir, se definen las operaciones que realizará el microprocesador, qué instrucciones aceptará, qué partes lógicas lo van a formar, etc. Posteriormente, este diseño se desarrolla a niveles de detalle cada vez mayores. Cualquier error en esta fase de diseño lógico será vital más tarde en la fabricación masiva de las unidades.

Una vez finalizado el proceso de diseño lógico y después de una depuración exhaustiva del mismo, se pasa a la fabricación de los chips en serie. Estos se fabrican mediante un proceso denominado fotolitografía. Los transistores y las conexiones entre ellos se crean en el semiconductor aplicando diferentes capas de varios materiales cada una en su lugar preciso y exacto. Un microprocesador actual puede estar compuesto de 12 o más capas. Un ordenador de gran precisión toma el diseño del microprocesador y lo separa en sus diferentes capas cada una de ellas se denomina máscara.

El semiconductor es tratado con una sustancia sensible a la luz y después se le aplica luz a través de la máscara. Así se crean las diferentes capas del chip. Este proceso se repite para cada máscara. Todo esto se hace a un nivel microscópico, por lo que el instrumental necesario para la fabricación es extremadamente preciso y costoso.

La siguiente fase es la del encapsulado, consistente en ensamblar el diminuto chip en una placa de circuito impreso con los diferentes contactos con los que luego se insertara en la placa base. Este es también un proceso delicado, ya que hay que conectar las diferentes patillas del encapsulado con los delicados y diminutos contactos del núcleo del microprocesador, también denominado die.

Finalmente se pasa a la fase de prueba y calibración de velocidad. No todas las unidades fabricadas tienen la misma calidad, debido a lo delicado del proceso, por lo que son probadas inicialmente a la frecuencia para la que han sido diseñadas, bajando dicha frecuencia progresivamente hasta aquella en la que el funcionamiento es estable. Por este motivo, microprocesadores diferentes generalmente son producidos en el mismo proceso, pero es al final en esta prueba donde son diferenciados, pasando a «etiquetarse» con una frecuencia característica según este proceso de selección final.

Un microprocesador es capaz de manejar dos elementos de información: instrucciones y datos. Las instrucciones forman los programas que se ejecutan en la máquina y los datos son los elementos

con los que trabajan esos programas. Un microprocesador dispone de conjuntos de instrucciones muy diversas: operaciones matemáticas simples, comparaciones, manejo de cadenas de caracteres, estructuras de programación, etc.

El elemento que marca el ritmo al que se ejecutan estas instrucciones es un reloj oscilador de cuarzo, de tal forma que mientras más alta sea su frecuencia de oscilación, se obtendrá en términos generales una mayor velocidad de trabajo. Los diferentes elementos que forman los actuales PC disponen de varias frecuencias de funcionamiento, medidas en Megahercios. Un Megahercio equivale a un millón de hercios o pulsos de reloj. Un pulso de reloj es la mínima unidad de tiempo en la que se lleva a cabo el procesamiento. A veces también se le denomina ciclo de reloj.

Existen instrucciones que emplean un solo ciclo y otras que necesitan más de un ciclo. Asimismo, muchos microprocesadores actuales permiten la ejecución de varias instrucciones en un solo ciclo. Mientras mayor sea el número de instrucciones que el microprocesador permita ejecutar en un solo ciclo, mayores serán sus prestaciones. Hoy en día, debido a los avances que se han producido en el campo de la microelectrónica y los procesos de fabricación de microprocesadores, lo habitual ya es hablar de Gigahercios. Un Gigahercio equivale a 1.000 Megahercios. En la actualidad es corriente encontrar microprocesadores de más de 3 Ghz de frecuencia de funcionamiento. Todas las frecuencias de reloj de una placa base actual son generadas por un mismo oscilador de cuarzo.

A la frecuencia base a la que funciona la placa se la denomina frecuencia del bus del sistema (FSB, Front Side Bus). Las frecuencias del resto de elementos de la placa base se obtienen mediante circuitos multiplicadores o divisores. La velocidad de proceso de un microprocesador se mide en Megahercios (Mhz), o lo que es lo mismo, en millones de ciclos de reloj por segundo. Supongamos un microprocesador que funcione a 3 Ghz, o lo que es lo mismo, 3.000 millones de ciclos por segundo. Si una instrucción emplea cinco ciclos de reloj para su ejecución, podemos calcular rápidamente el número de instrucciones por segundo que es capaz de procesar este microprocesador:

$$3.000.000.000/5 = 600 \text{ millones de instrucciones por segundo, también denominadas MIPS.}$$

El número de MIPS constituye una de las unidades empleadas más habitualmente para expresar la potencia de cálculo de un microprocesador.

Otra unidad habitual son los Megaflops, o millones de operaciones en coma flotante que es capaz de realizar el microprocesador por segundo. Los microprocesadores disponen también de determinados buses de datos, o caminos por los que fluye la información. Estos suelen ser dos: el bus interno, por el que se comunica la información dentro del propio microprocesador, y el bus externo, con el que el microprocesador se comunica con el exterior. Generalmente, los actuales microprocesadores de Intel hasta ahora disponían de un bus interno de 32 bits. No obstante, con la aparición de microprocesadores como el Itanium de Intel y los modelos Opteron y Athlon 64 de AMD se inauguró la era de los 64 bits en la historia de los microprocesadores tradicionales.

Otro parámetro importante es el número de bits que usa el microprocesador para direccionar la memoria, lo que indica el tamaño máximo de memoria RAM que un ordenador puede manejar o direccionar. Así, un antiguo 286 disponía de 24 bits para direccionamiento de memoria, lo que limitaba a 16 Mb la cantidad máxima de memoria que podía manejar. Los microprocesadores 386 y 486 ya disponían de 32 bits para direccionamiento de memoria, lo que les permitía manejar un máximo de 4.096 Mb de memoria, cantidad que no todas las placas base actuales son capaces de alojar.

Con la tecnología actual es posible integrar en el interior del microprocesador muchos millones de transistores. Todos ellos se organizan en un circuito llamado die de un tamaño no superior a los 4 cm². Este circuito se envuelve en un encapsulado cerámico o plástico que lo protege, y en el que se realizan todas las conexiones con las patillas que permiten el ensamblado del microprocesador en el zócalo correspondiente de la placa base. El aumento en la complejidad de los modernos microprocesadores obedece a lo que se denomina Ley de Moore, según la cual cada 18 meses la tecnología avanza lo suficiente como para integrar el doble de transistores en el interior de un microprocesador.

Otra parte integrante de todo microprocesador es su unidad de control, que es el módulo que controla el flujo de información a través del microprocesador. Concretamente, la unidad de control realiza funciones como la alimentación de las instrucciones, decodificación de las mismas, control de la ejecución y almacenamiento de los resultados.

Otro componente es la unidad de ejecución de enteros. La mayoría del trabajo de cálculo del microprocesador se hace sobre números enteros, a diferencia de los números en coma flotante, de cuya ejecución se encarga la unidad de coma flotante (FPU, Floating Point Unit), más conocida como coprocesador matemático. Los microprocesadores antiguos disponían de una sola unidad de ejecución de enteros. Los actuales microprocesadores incorporan varias de estas unidades, lo que permite la ejecución de varias operaciones con enteros de manera simultánea.

Los procesadores que incorporan más de una unidad de enteros se denominan superescalares. La otra unidad de cálculo, la de coma flotante, se integra dentro de todos los microprocesadores desde el 486 de Intel. Anteriormente era necesario insertarla como un chip adicional en la placa base, en un zócalo preparado a tal efecto.

En todo microprocesador se integra una memoria caché muy rápida, la denominada caché L1 (nivel 1). Su tamaño suele oscilar entre 16 y 128 Kb y generalmente se emplea la mitad para datos y la otra mitad para instrucciones, aunque puede darse el caso contrario, es decir, que se use la misma caché indistintamente para datos y para instrucciones. En este último caso, la caché se denomina caché unificada.

El rendimiento que es capaz de ofrecer un microprocesador es algo de vital importancia. Este rendimiento es el resultado de numerosos factores, y no sólo de la frecuencia de funcionamiento. Podemos calcular el rendimiento real como el resultado de la fórmula:

$$\text{Rendimiento} = \text{IPC} \times \text{Frecuencia}$$

El IPC es el número de instrucciones que el microprocesador es capaz de ejecutar en cada ciclo de reloj. El IPC viene a determinar lo que se denomina paralelismo de una microarquitectura. Este factor es fundamental, y es el causante de la diferencia de prestaciones existentes entre los microprocesadores Pentium 4 de Intel y Athlon XP de AMD.

El segundo factor de la ecuación indica la frecuencia de funcionamiento en Mhz. Esta frecuencia de trabajo es, a su vez, el producto del FSB (frecuencia de bus del sistema) por un factor de multiplicación. Estos no son todos los factores que determinan el rendimiento. En menor medida, existen otros como la implementación que se haga de una arquitectura en particular y el juego de instrucciones del chip que influyen decisivamente en el rendimiento de cierto número de aplicaciones. El trabajo de un microprocesador es ejecutar instrucciones en un lenguaje máquina, el único lenguaje que el microprocesador es capaz de entender. Las diferentes instrucciones que un determinado microprocesador es capaz de ejecutar reciben el nombre de conjunto de instrucciones, y suele ser diferente entre distintos tipos de microprocesadores.

En principio, se puede pensar que un microprocesador se debe diseñar para que ejecute el mayor número de instrucciones diferentes, pero no es así, ya que en tal caso sería necesario el empleo de un número muy elevado de transistores y esto dificultaría el diseño. Este criterio es el que define las dos principales arquitecturas de microprocesadores existentes en la actualidad: CISC y RISC (vistas más arriba).

Los microprocesadores más antiguos ejecutaban las instrucciones de manera secuencial. Hasta que no acababa de ejecutarse completamente una instrucción no se ejecutaba la siguiente. Esto era bastante ineficiente, puesto que cada instrucción se ejecuta siguiendo una serie de pasos en orden lineal. Y entre unos pasos y otros se producen tiempos muertos. La solución consistía en ejecutar más de una instrucción de manera simultánea, eso sí, cada una de ellas en una fase diferente de ejecución. Esta técnica, muy usada en numerosos sistemas, se denomina segmentación encauzada o *pipelining*. El *pipelining* incrementa de manera espectacular el rendimiento de un microprocesador, por lo que es una técnica empleada masivamente en todos los modelos actuales. Sin embargo, tiene algunos inconvenientes. El principal es la dependencia de datos, es decir, que existan instrucciones en ejecución simultánea que compartan los mismos datos, por lo que la siguiente instrucción no se podrá ejecutar hasta que la anterior haya terminado de procesar el dato necesario. No obstante, los microprocesadores actuales han sido optimizados lo suficiente como para eliminar en gran medida este problema. Los pasos típicos en la ejecución de cualquier instrucción en un microprocesador son:

1. **Lectura de la instrucción.** El primer paso consiste en cargar la instrucción que se va a ejecutar, lo que implica una lectura directa de memoria. Algunos microprocesadores disponen de unidades de prealimentación (*pre-fetch*), que leen de memoria la siguiente instrucción mientras se ejecuta la instrucción actual, con lo que se gana un tiempo fundamental.
2. **Descodificación.** Esta fase se encarga de determinar el tipo de instrucción, la longitud de la misma y si requiere leer datos de memoria.
3. **Generación de dirección.** Para aquellas instrucciones que acceden a memoria es necesario obtener la dirección de memoria a la que necesitan acceder.
4. **Ejecución** propiamente dicha de la instrucción.
5. **Obtención del resultado.** Generalmente ese resultado deberá ser escrito en memoria.

Existen tres modos de operación del microprocesador bien conocidos desde hace tiempo: el modo real, el modo protegido y el modo virtual.

El **multiproceso** o **multiprocesamiento** consiste en la ejecución de programas o procesos en un sistema con más de un microprocesador. En teoría esto debería doblar las capacidades en general: doble velocidad y mitad de tiempo, pero en la práctica no es axial, aunque sí que es cierto que se consigue un aumento considerable en el rendimiento. Para que un sistema admita multiproceso deben darse una serie de factores:

- La placa base debe soportar multiproceso. Esto implica que la placa base debe integrar un chip-set con soporte para multiproceso, así como varios zócalos, tantos como microprocesadores se vayan a utilizar. Soporte del propio microprocesador. No todos los microprocesadores admiten la posibilidad de multiproceso.
- El sistema operativo debe soportar el multiproceso. Entre los sistemas operativos con soporte para multiproceso se encuentran Windows NT, Windows 2000, Windows XP Profesional, Windows 2003 Server, Linux, diferentes versiones de UNIX, BeOS, y algunos otros sistemas operativos de uso más específico.

El sistema operativo, funcionando en multiproceso, divide un proceso en varios subprocesos o hilos de ejecución denominados threads, que se ejecutan simultáneamente, repartiéndose la carga entre los diferentes microprocesadores. Si la aplicación que hay que ejecutar no ha sido desarrollada pensando en estas funciones, el multiproceso no funcionaría adecuadamente, aunque siempre se podría sacar provecho del sistema al ejecutar varias tareas independientes de manera simultánea, lo que se denomina multitarea.

Se denomina overclocking al hecho de hacer que un microprocesador funcione a una frecuencia mayor de aquella para la que se ha diseñado. Así, por ejemplo, si tenemos un Pentium 4 a 1,8 Ghz y configuramos la placa base para que funcione a 2 Ghz, estamos haciéndolo funcionar como si se tratase de un Pentium 4 a 2 Ghz. Esto puede funcionar, pero nunca debemos perder de vista que estamos forzando un componente por encima de sus especificaciones y que en cualquier momento puede fallar.

El diseño del Pentium 4 se basa en la denominada microarquitectura NetBurst. Esta arquitectura se basa en la combinación de cuatro tecnologías para aumentar el rendimiento del microprocesador:

1. Tecnología de hipercanalización (hyper pipelined). Aumenta el número de canales de desarrollos anteriores de Intel, con objeto de que el microprocesador sea capaz de manejar más instrucciones por ciclo de reloj. Concretamente, en el caso del Pentium 4 el número de etapas es de 20, frente a las 10 del Pentium III.
2. Motor de ejecución rápida. En el caso del Pentium 4, algunas de sus partes funcionan al doble de la frecuencia del microprocesador, en concreto dos unidades de ejecución de enteros.
3. Caché de traza de ejecución. Es una memoria caché que puede almacenar hasta 12.000 microoperaciones listas para ser ejecutadas, sin tener que decodificarse previamente. De esta manera se dispara la tasa de aciertos de caché en la ejecución de instrucciones.
4. Un bus del sistema a 400 Mhz. En realidad es un bus de 400 Mhz cuadruplicado, que se puede adaptar a las frecuencias del microprocesador, eligiendo determinadas frecuencias dentro del rango de los 100 a los 400 Mhz. Esta frecuencia hace posible además tasas de transferencia teóricas de hasta 3,2 Gb/s con memoria RDRAM PC800 (800 Mhz).

MICROPROCESADORES ACTUALES:

32 bits (Bus de direcciones):

- AMD K6-III.
- Pentium Celeron «A» (con caché).
- Pentium III.
- AMD Athlon (K7).
- AMD Duron.
- Pentium 4.

64 bits:

- Itanium (Intel).
- Opteron (AMD).

4.3. MEMORIA RAM.

El bus de direcciones contiene un número de líneas determinado y cada una puede contener un bit. Al número de bits del bus de direcciones, al igual que al de datos, se le denomina ancho de banda del bus. Mientras mayor sea éste, o lo que es lo mismo, mientras mayor número de bits tenga, mayor cantidad de memoria podrá ser direccionada. La memoria SDRAM se implementa en los módulos DIMM y en un gran número de tarjetas gráficas. Su principal característica es que se trata de una memoria síncrona, por lo que utiliza un reloj para sincronizar la lectura y la escritura en los chips de memoria. Este reloj está sincronizado con el reloj interno del microprocesador, lo que hace que el rendimiento en lectura/escritura entre microprocesador y memoria se dispare considerablemente. El interés principal por la memoria SDRAM estriba en el hecho de que es capaz de alcanzar frecuencias de acceso iguales ó superiores a los 100 Mhz. De hecho, este es el tipo de memoria que se integra en los DIMM denominados PC100 y PC133.

Dentro de las memorias que tienen un modo de funcionamiento síncrono se incluyen las siguientes: CDRAM (cache DRAM), RDRAM (Rambus DRAM), ESDRAM (Enhanced SDRAM) y la SDRAM habitual, conocida como JEDEC DRAM síncrona estándar, con la variante SDRAM DDR de doble tasa de transferencia.

Conceptos sobre Memorias:

- RAM: la memoria principal del ordenador. En sistemas operativos modernos tipo Windows, la velocidad y especialmente la cantidad de RAM es un factor determinante del rendimiento.
- DRAM: memoria RAM Dinámica, es decir, necesita refresco cientos de veces por segundo, para retener los datos en sus celdas de memoria. Se construyen mediante condensadores.
- SRAM (Static RAM): memoria RAM que no necesita refresco tan frecuente (mayor rapidez). Es unas cinco veces más rápida y dos veces más cara y dos veces más grande que los módulos DRAM.

Debido a su costo más bajo y el tamaño más pequeño, DRAM se prefiere como memoria principal en los computadores, mientras que SRAM, por su velocidad, se usa principalmente para memorias caché.

- ns: nanosegundos. Estos suelen aparecer en cada chip que conforman el módulo mediante uno o dos números precedidos de un guión (-). Esto nos indica el tiempo de acceso de la memoria. Nos especificará la velocidad para la que está fabricada la memoria. Ejemplo: si son módulos de 7.5 ns lo que ofrece es una velocidad teórica de $(1000/7.5) = 133\text{MHz}$. Esto también nos indica el PC de la memoria, en este caso PC2100
- SDRAM (Synchronous Dynamic Random Access Memory): memoria de acceso aleatorio dinámico y síncrono. Trabaja a la misma velocidad interna del procesador.

- **DDR (Double Data Rate SDRAM).** SDRAM con ratio doble de datos, su antigüedad data del mismo tiempo que la SDRAM pero no es hasta ahora cuando comienza a extenderse. Esta memoria se diferencia de la SDRAM convencional en su capacidad de activar la salida de datos no sólo al comienzo del ciclo de reloj del procesador sino también a su fin. Esto teóricamente aumenta por dos la capacidad de envío de datos al sistema.
- **PC:** la denominación PC sirve para diferenciar las capacidades de velocidad de los diferentes módulos que podemos encontrar. En el caso de la memoria DDR tenemos PC1600, que hace referencia a los módulos preparados para funcionar a 100MHz, PC2100 para los módulos de 133MHz, PC2400 para los módulos de 150MHz y los próximos módulos PC2700 preparados para 166MHz.
- **RDRAM o Rambus DRAM:** un nuevo tipo de memoria, de diseño totalmente distinto al de la SDRAM; teóricamente ofrece mejor rendimiento basada en el bus Rambus.
- **DIMM: (Dual inline Memory module).** Módulos de memoria duales en línea, esto significa que dentro de un solo modulo se incluyen realmente dos módulos para lograr los 64Bits necesarios para no tener que instalar los módulos por pares. Los que hayáis tenido la oportunidad de usar memoria SIMM (single inline memory module) os acordaréis que necesitabais dos módulos iguales para poder ampliar el ordenador. Las memorias en módulo DIMM convencionales utilizan 168 contactos, mientras que los DIMM – DDR utilizan 184 contactos.
- **RIMM: módulo de memoria RDRAM (Rambus).** 168 contactos. Bus de datos más estrecho, de sólo 16 bits (2 bytes) pero funciona a velocidades mucho mayores, de 266, 356 y 400 MHz. Además, es capaz de aprovechar cada señal doblemente, de forma que en cada ciclo de reloj envía 4 bytes en lugar de 2. Inconveniente: coste.
- **SODIMM («Small Outline Dual in-line Memory Module»)** es un tipo de DIMM de perfil pequeño. Ha sido diseñado para ordenadores en los que se dispone de poco espacio, por ejemplo «laptops», tablet y portátiles e impresoras. Utiliza convencionalmente 144 o 200 contactos y sus dimensiones han sido reducidas a 66 x 25 mm.

5. EQUIPOS DEPARTAMENTALES O MINIORDENADORES.

5.1. DEFINICIÓN.

(Equipo departamental = Miniordenador)

Equipos modulares para entornos multiusuario, multitarea y multiprocesador de instalación y manejo no muy complicados diseñados para facilitar el trabajo en ambientes cooperativos o para trabajo en grupo.

5.2. CARACTERÍSTICAS.

- Resulta cada vez más difícil trazar una línea divisoria entre: superordenadores (Procesamiento Simétrico o Masivamente paralelos), microordenadores y equipos departamentales o miniordenadores.

- El pionero de los smtas. Mini fue DEC con su VAX / PDP-11.
- Estos equipos fueron la base sobre la que se pudo desarrollar la arquitectura Cliente/Servidor.
- Mejor relación calidad/precio que en las soluciones de superordenador.
- Alternativa a la arquitectura maestro/esclavo propuesta por SNA de IBM.
- Mejora de la portabilidad e interoperabilidad del equipo lógico mediante el soporte de interfaces estándares.
- Redundancia de componentes para conseguir sistemas tolerantes a fallos.
- Gestión de memoria basada en caché.
- Muchas de las soluciones actuales se basan en plataformas Intel.
- Migración de las tradicionales aplicaciones de mainframe a estas plataformas (Downsizing).

5.3. FUNCIONALIDADES.

- Server principal de una empresa pequeña o mediana en la que se ofrecería un precio bajo al usuario y múltiples aplicaciones de gestión estandarizadas.
- Server asociado o incluso departamental instalado en una dependencia separada y dependiente de la sede central de la organización. Se concebía la máquina departamental como una herramienta de productividad con un precio bajo que evitaba el empleo masivo de líneas de teleproceso hacia los mainframes corporativos.
- Server autónomo para aplicaciones poco repetitivas o especializadas como las de Automatización de Oficinas con soporte directo del proveedor.

Ejemplos: iServers o xServers de IBM, PowerEdge de Dell, Enterprise 10000 de Sun etc.

6. MAINFRAMES O SUPERORDENADORES.

6.1. DEFINICIÓN.

Ordenador multiusuario, multitarea y multiprocesador de muy altas capacidades de cómputo, orientado al procesamiento de grandes volúmenes de información tanto en la gestión, como en el cálculo científico de grandes organizaciones y empresas.

6.2. CARACTERÍSTICAS.

- Se suelen utilizar habitualmente en modo batch.
- Para su uso en modo interactivo se hace necesaria la utilización de monitores de teleproceso, como por ejemplo: CICS (Costumer Information Control System).

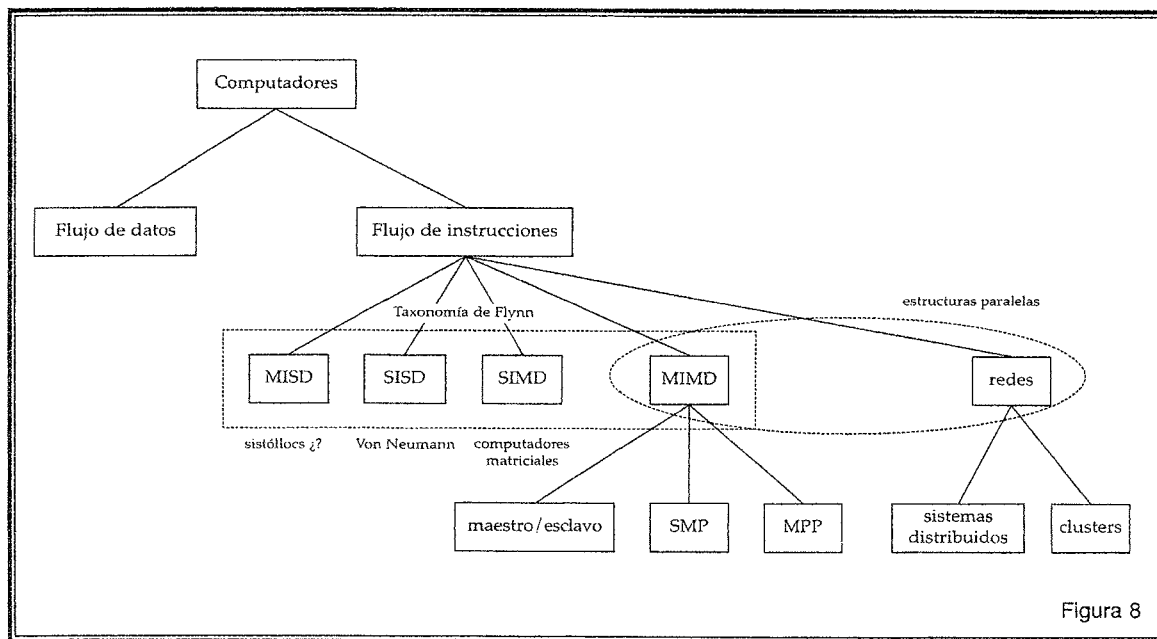
- Suelen ser sistemas altamente propietarios con difícil integración en entornos heterogéneos. Ejemplo: arquitectura SNA de IBM.
- Existen pocos fabricantes de este tipo de ordenadores. Tradicionalmente el más representativo ha sido IBM.
- La alta capacidad de cálculo se consigue normalmente mediante «paralelización» a distintos niveles.
- En la actualidad se asiste a un proceso de «downsizing» mediante la construcción de ordenadores virtuales (multicomputador) con prestaciones similares a las de un mainframe.

6.3. FUNCIONALIDADES.

6.3.1. Clasificación de superordenadores atendiendo a la implementación Hardware.

Se basa en:

- Flujo de datos (sencillo/múltiple): el flujo de ejecución es conducido por los datos: una instrucción será ejecutada cuando los operandos estén disponibles.
- Flujo de Instrucciones (sencillo/múltiple): las instrucciones se van ejecutando según el orden en el que se encuentran en memoria.

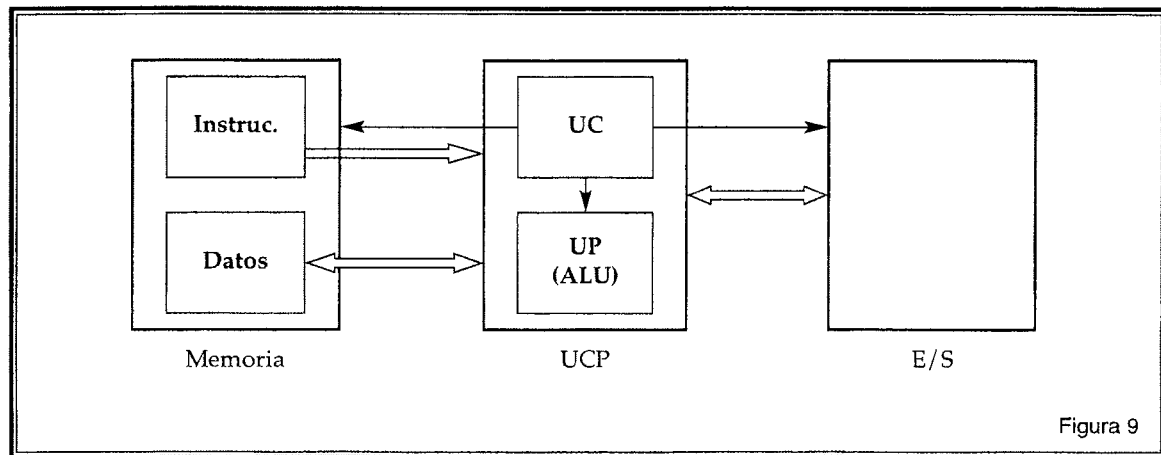


6.3.2. SISD.

Flujo único de instrucciones.

Flujo único de datos:

- Monoprocesadores, procesadores segmentados
- VAX-11, IBM 360...

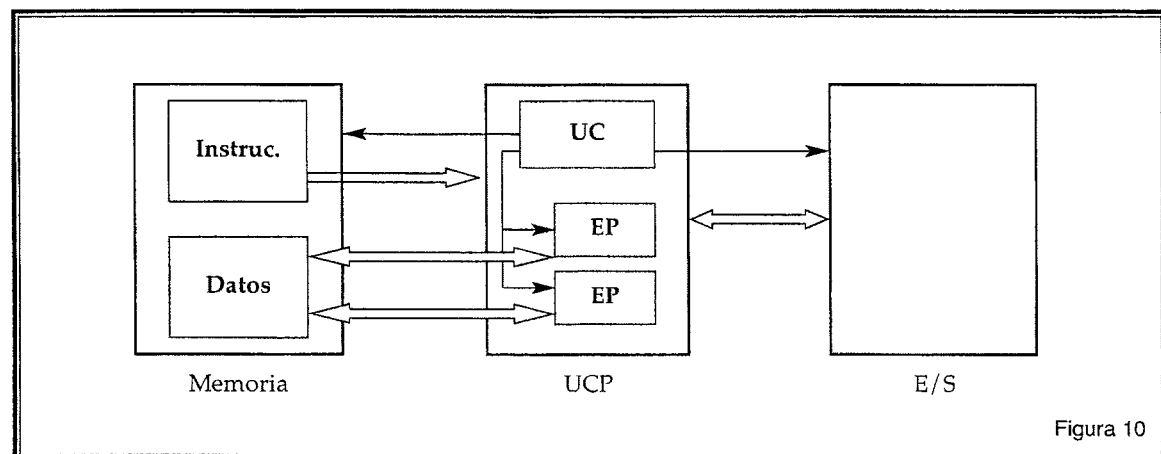


6.3.3. SIMD.

Flujo único de instrucciones.

Flujo múltiple de datos:

- Una instrucción actúa sobre varios elementos de procesamiento (EP).
- Procesadores vectoriales y matriciales (PVP=Parallel Vector Processing).
- Ejemplo: Connection Machine 2, ILLIAC IV, CRAY, FUJITSU...



6.3.4. MISD.

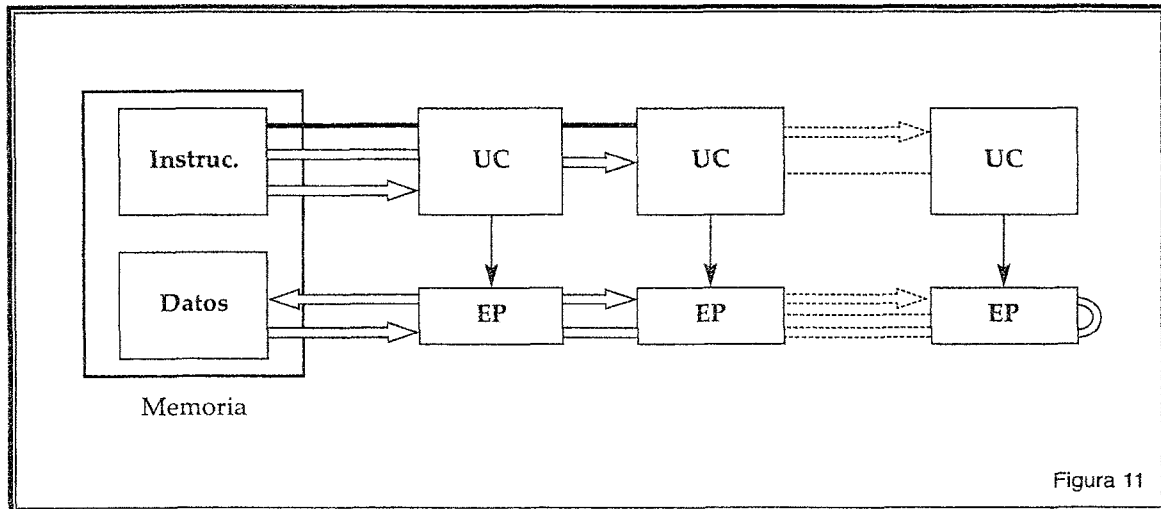
Flujo múltiple de instrucciones.

Flujo simple de datos:

- Varios flujos de instrucciones independientes y un único flujo de datos.
- Arquitecturas sistólicas y arquitecturas desacopladas.

No implementada comercialmente.

Arquitecturas Sistólicas. Se caracterizan por un flujo de datos regular (rítmico) a través de numerosos elementos de proceso.



6.3.5. MIMD.

Flujo múltiple de instrucciones.

Flujo múltiple de datos:

- Multiprocesadores MMC (Multiprocesadores de Memoria Compartida) y MMD (Multiprocesadores de Memoria Distribuida).
- Ejemplo: Cm*, BBN Butterfly, CM-5...

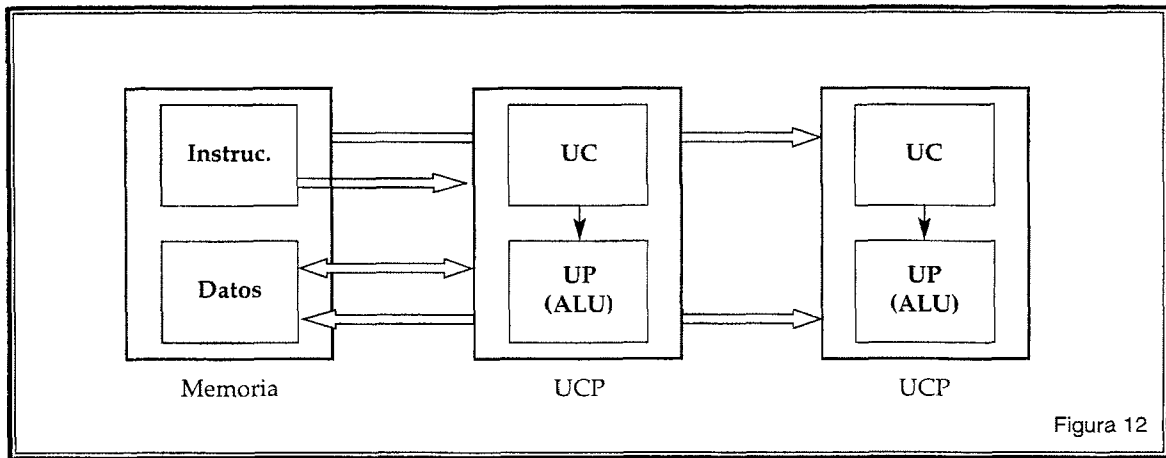


Figura 12

6.3.5.1. MMC o SMP (Sistema de Multiprocesamiento Simétrico).

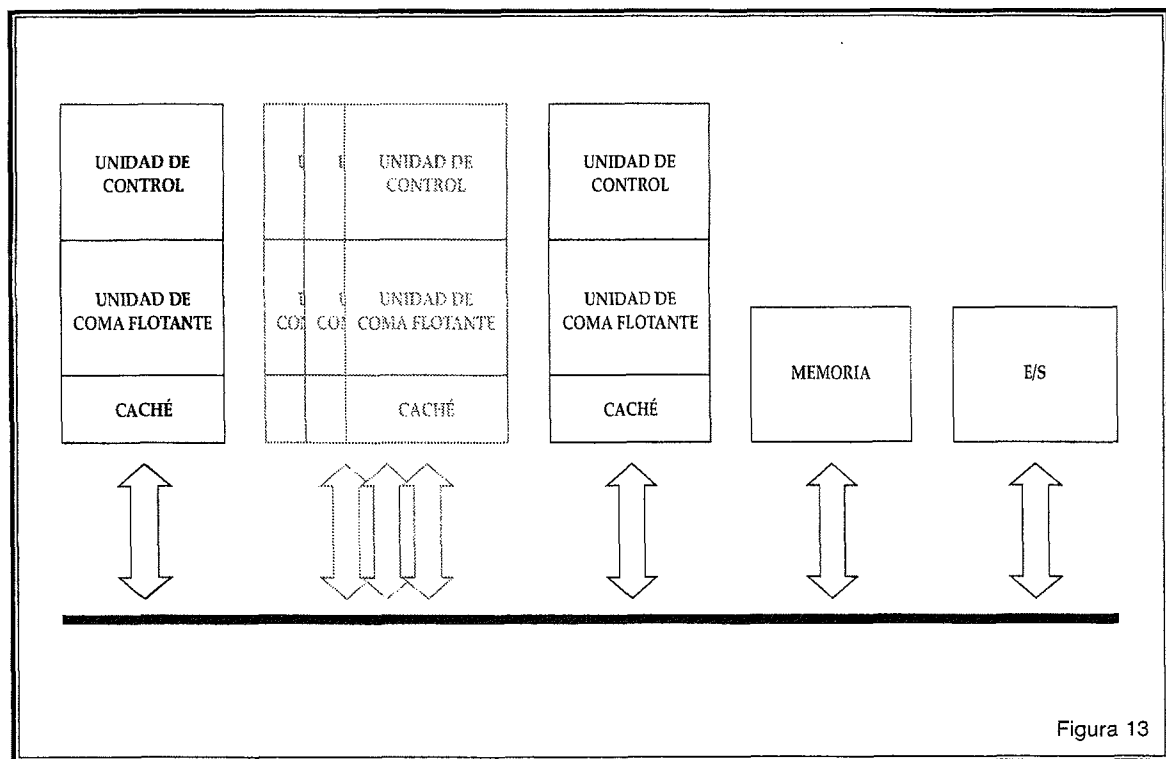
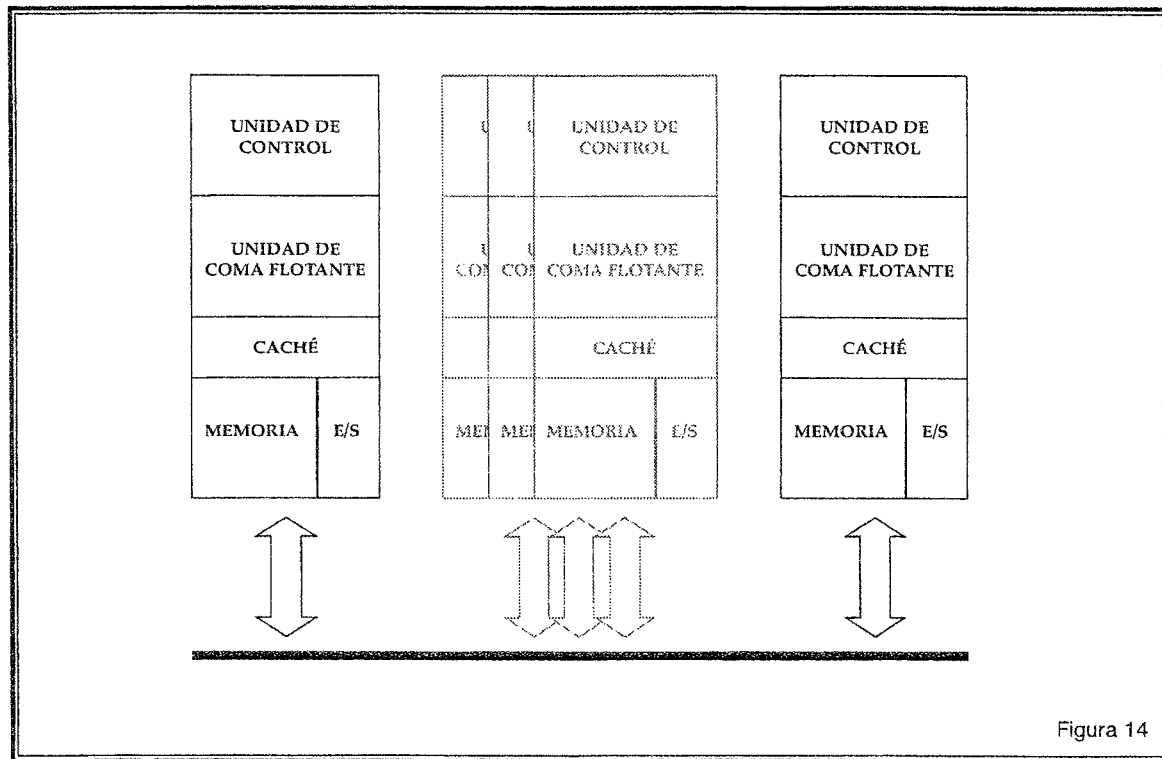


Figura 13

Los sistemas MMC se pueden clasificar en:

- **UMA (Uniform Memory Access).** 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.
- **NUMA (Non Uniform Memory Access).** Sistema de memoria compartida donde el tiempo de acceso varía en función de dónde se encuentre localizado el acceso.

6.3.5.2. MMD o MMP (Massively Parallel Processor).



7. UNIDADES DE MEDIDA DEL RENDIMIENTO DE LOS ORDENADORES.

NOMBRE	VALOR	RANGO
MFLOPS	megaFLOPS = 10^6	Rango de las workstations actuales
GFLOPS	gigaFLOPS = 10^9	Rango de los actuales supercomputadores
TFLOPS	teraFLOPS = 10^{12}	Rango de los supercomputadores que están apareciendo en la actualidad
PFLOPS	petaFLOPS = 10^{15}	El sueño. Los ingenieros se preguntan qué problemas se podrán resolver con esta potencia
EFLOPS	exaFLOPS = 10^{18}	Probablemente necesite una tecnología nueva

BIBLIOGRAFÍA

- CULLER, D. E., SINGH, J.P., GUPTA, A. *Parallel Computer Architecture. A Hardware/Software Approach*. Morgan Kauffman Publishers, Agosto 1998.
- HWANG, K. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. McGraw-Hill, 1993.
- STONE, H.S. *High-Performance Computer Architecture*. 3.^a Edición. Addison Wesley, 1993.
- PATTERSON, D. A., HENNESSY, J. L. *Estructura y diseño de Computadores: Interficie circuitería/programación*. Vol. 1. Reverté, 2000.
- PATTERSON, D. A., HENNESSY, J. L. *Estructura y diseño de Computadores: Interficie circuitería/programación*. Vol. 3. Reverté, 2000.
- TANENBAUM, A. S. *Organización de computadoras: Un enfoque estructurado*. Prentice Hall, 2000.
- STALLINGS, W. *Organización y Arquitectura de computadores: Diseño para optimizar prestaciones*. Prentice Hall, 2000.



