

## **INTRODUCCIÓN A LA ARQUITECTURA DE COMPUTADORAS**

### **DEFINICIONES**

El computador es una máquina capaz de interpretar y ejecutar una serie de instrucciones. Esta constituido por un conjunto de bloques lógicos electrónicos, comunicados entre si mediante conectores de líneas o buces e interruptores, gobernados por unos controladores .

### **OBJETIVO**

Ejecutar programas formados por secuencias de instrucciones, de una manera rápida y cómoda (disponen de un sistema operativo y compiladores).

Al estudiar la arquitectura de los computadores hay que considerar los condicionantes físicos, así como también deben tenerse en cuenta el sistema lógico y las aplicaciones que han de soportar.

Los arquitectos de computadoras se han preocupado en especial de potenciar el EQUIPO FÍSICO aplicando los constantes avances de la Microelectrónica. En verdad, los costes de la circuiteria o hardware se han reducido continuamente pero a sido a costa del encarecimiento del sistema lógico o software.

La misión fundamental de un arquitecto de computadoras es definir el computador al nivel de lenguaje de máquina, esto es, especificar la máquina que ve el programador, y establecer los objetivos de velocidad y capacidad que debe tener la máquina para proporcionar los recursos precisos para el desarrollo y explotación de los programas en los lenguajes de alto nivel, que constituyen las aplicaciones del usuario.

Los análisis previos al diseño de la máquina han de impulsar al arquitecto de computadoras a crear una estructura que proporcione un buen rendimiento entendido por tal, el que se refiere al comportamiento conjunto de los elementos principales que intervienen en el sistema:

- 1)- Lenguaje de programación.
- 2)- Compilador.
- 3)- Sistema operativo.
- 4)- Máquina.

### **ESTRUCTURA CLÁSICA DEL COMPUTADOR**

Con el descubrimiento de la válvula en 1904, FLEMING abrió las puertas a la historia de la Electrónica, cuyos formidables avances actúan como motor de impulsión para las restantes ciencias y técnicas.

Al aplicar la tecnología electrónica en máquinas, estas se fueron haciendo más rápidas y potentes. Hasta mediados del siglo XX cada máquina estaba construida y cableada para soportar una misión concreta. Fue en esa época cuando JOHN VON NEUMANN continuando con su predecesor BABBAGE y otros investigadores, presento su MAQUINA DE PROGRAMA ALMACENADO (que los denomino computador).

La máquina de VON NEUMANN no estaba diseñada para resolver una aplicación concreta, sino que podía emplearse en diferentes trabajos previa la conveniente preparación del funcionamiento secuencial que debía llevar a cabo.

### **PROBLEMAS EN EL DESARROLLO DE LA ARQUITECTURA DE COMPUTADORES**

Hasta hace pocos años, ha sido tradicional que los arquitectos de computadores estableciesen su meta en diseñar máquinas con buenas características, pero teniendo en muy poca consideración las prestaciones del sistema lógico.

Las dos causas principales del escaso progreso en la arquitectura de computadores en las tres primeras décadas de su evolución han sido:

- a)- La falta de una adecuada definición de las funciones que debía de soportar el hardware y las que corrían a cargo del software.
- b)- El seguimiento a ultranza del modelo propuesto por VON NEUMANN, que, aunque revolucionario en su época, no estaba diseñado para manejar los nuevos sistemas operativos, lenguajes y aplicaciones. En

este sentido se puede afirmar que, desde la construcción de los computadores de tipo EDVAC hasta nuestros días, solo se han introducido unas escasas novedades entre las que se destacan:

- Empleo de registros índices para propiciar el direccionamiento indexado.
- Introducción de conjuntos de registros de propósito general.
- Direccionamiento indirecto.
- Procesadores de entrada y salida.
- Memoria virtual

Los mayores obstáculos que presenta la arquitectura por VON NEUMANN se centran en dos aspectos:

1)- El empleo de una memoria unidimensional, de tipo secuencial, en la que las informaciones se almacenan de forma consecutiva. Los modernos computadores precisan una memoria muy flexible, así como manejar datos en estructuras multidimensionales.

2)- La falta de distinción entre datos e instrucciones, lo que significa que cualquier información puede ser considerada como un dato o una instrucción. Esto es desaconsejable para la ejecución de programas escritos en lenguajes de alto nivel.

Otras razones, de carácter secundario, que han favorecido al desacoplamiento entre el equipo físico y el sistema lógico, son:

- Uso exclusivo de la aritmética binaria.
- Empleo de palabras de memoria de tamaño fijo.
- Limitación del número de registro.

Las arquitecturas avanzadas de computadores exigen cambios conceptuales que hacen referencia al uso de datos autodefinidos, bien sea a base de etiquetas o de descriptores, y al desarrollo de sistemas de procesamiento en paralelo.

## **INFLUENCIA DE LA TECNOLOGÍA EN LA EVOLUCIÓN DE LA ESTRUCTURA BÁSICA DE LOS COMPUTADORES**

Hasta llegar a los supercomputadores actuales, en los que se enfatizan todas las características físicas y lógicas para conseguir una elevadísima velocidad, la estructura básica del procesador central ha sufrido una evolución, relacionadas con la tecnología de fabricación.

### **PRIMERA ETAPA**

Los primeros computadores se construyeron siguiendo dos principios:

a)- El modelo propuesto por VON NEUMANN, que incluía el concepto de PROGRAMA ALMACENADO y el de la RUPTURA DE LA SECUENCIA en el programa. b)- La tecnología electrónica de la época, que hacía uso de las VÁLVULAS DE VACÍO.

En la FIGURA 1 se muestra el diagrama por bloques del procesador central correspondiente a la 1er. ETAPA.

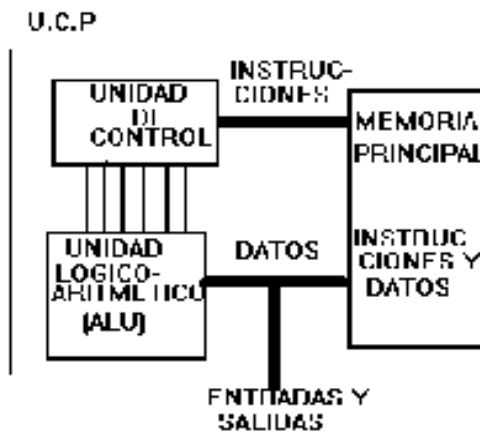
Explicación de la FIGURA 1: La unidad de control extrae la instrucción de la memoria principal, la interpreta y efectúa las siguientes operaciones:

- a)- Establece el conexionado eléctrico de la ALU.
- b)- Extrae los datos de la MEMORIA.
- c)- Ordena la ejecución a la ALU.
- d)- Almacena el resultado en la MEMORIA.

La ALU es el bloque operativo que desarrolla todas las posibles funciones aritmético - lógicas.

Tanto la LÓGICA DE CONTROL como la MEMORIA, estaban construidas con VÁLVULAS DE VACÍO con que la velocidad de funcionamiento de ambas era similar. La sencillez de la CPU y la escasez de registros internos de trabajo imponían una constante transferencia con la MEMORIA empleando un conjunto reducido y básico de instrucciones de máquina.

FIGURA 1: Diagrama general por bloques de estructura de los procesadores de la primera etapa.



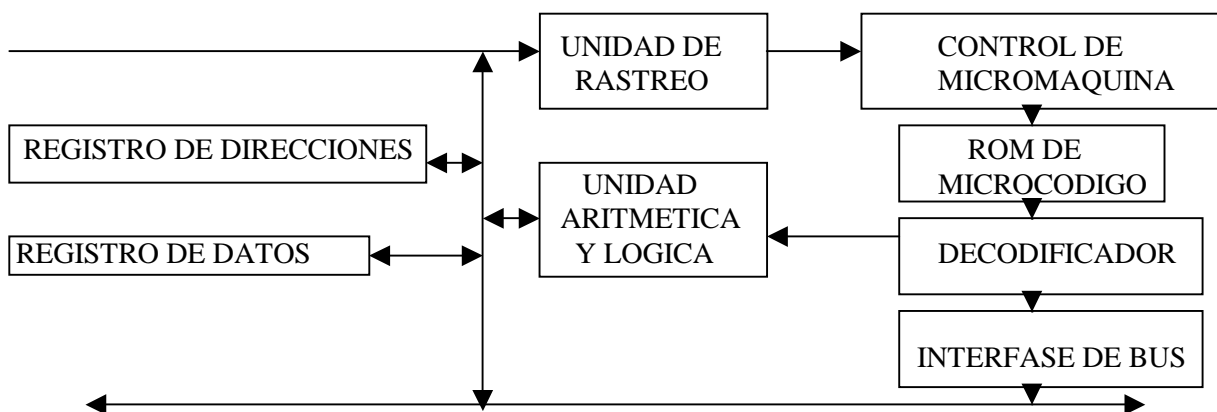
## SEGUNDA ETAPA

En esta etapa se produce un distanciamiento entre la tecnología usada en la construcción de la unidad de control y en memoria principal. Los circuitos integrados de pequeña y media escala de integración (SSI Y MSI) se usan para la construcción de la CPU, pero todavía no se han alcanzado cotas interesantes en la densidad de integración por lo que en la sección de la memoria principal se aplican otras tecnologías, como los NÚCLEOS DE FERRITA, cuyos tiempos de acceso eran elevados.

La velocidad de la memoria principal es mucho menor que la de la CPU (unas 10 veces), lo que provoca largos períodos de inactividad en la CPU, mientras se accedía a memoria.

En esta época las máquinas comienzan a soportar lenguajes de alto nivel, como el FORTRAN Y COBOL, que tenían que ser traducidos a lenguaje máquina por compiladores, antes de ser ejecutados. En un intento de simplificar la compilación o traducción, se propicio una potenciación de las instrucciones máquina para asemejarlas a las de alto nivel.

Surgieron juegos de instrucciones complejos, en los que cada instrucción equivalía a varias operaciones simples, llamadas OPERACIONES ELEMENTALES. Así se evitaban muchos accesos a la memoria principal, al mismo tiempo que se sacaba el máximo rendimiento de la rápida CPU. A este tipo de computadores se les denominó CISC (Computadores de juego de Instrucciones Complejo).



En muchos computadores la CPU paso a contener a la MEMORIA DE CONTROL, que se trataba de una rapidísima memoria en la que se almacenaban las operaciones elementales correspondientes a cada instrucción compleja o macrocódigo, llamándose MICROINSTRUCCION a cada una de las posiciones de esta MEMORIA DE CONTROL.

Se incrementa el proceso de decodificación de la MACROINSTRUCCION, pero se reduce el numero de accesos a la memoria principal.

Con este sistema se intentaba paliar el desfase de velocidades entre la CPU y la memoria principal. En la FIGURA 2 el computador quedaba limitado a la velocidad de la Memoria de Control, que, si bien se construía con semiconductores, con los que era mucho más rápida que la memoria principal, no permitía alcanzar las mismas velocidades que la lógica cableada.

En los computadores CISC microprogramados hubo un intercambio mediante la Memoria de Control, entre la LÓGICA CABLEADA de las instrucciones simples y la LÓGICA PROGRAMADA de las complejas. Así, se resolvía el problema derivado de las distintas velocidades entre las dos secciones principales del procesador. Además, se paliaba la DIVERGENCIA SEMÁNTICA con las instrucciones de los lenguajes de alto nivel.

El número de instrucciones complejas de los CISC solo estaba limitado por la capacidad de la memoria de control.

### **TERCERA ETAPA**

El vertiginoso desarrollo tecnológico de los CIRCUITOS INTEGRADOS consiguió alcanzar la ALTA ESCALA DE INTEGRACIÓN, que permitía la fabricación de memorias electrónicas rápidas y de cierta capacidad.

Aparece la CACHE de memoria ULTRARRAPIDA, de tipo tapón, que se encarga de guardar la información de uso más frecuente de la memoria principal, para disminuir el número de accesos a esta última. Estas memorias equilibran la diferencia de velocidades entre la CPU y la memoria principal. En la FIGURA 2: La Unidad de Control se alimenta desde la Memoria Cache, o bien, cuando la información que necesita no está en ella, directamente desde la memoria principal. El contenido que guarda la CACHE está optimizado para que sea el de más uso, consiguiendo velocidades de 5 a 10 veces mayores que la memoria principal.

Los computadores con memoria cache intermedia poseen una CPU que, al tener que decodificar instrucciones complejas, tardan más en esta función que en el acceso a la memoria, por lo que este tipo de instrucción deja de parecer tan interesante.

Para mejorar el diseño de los computadores se pasó a realizar un análisis de lo que sucedía con los programas en general y se comprobó:

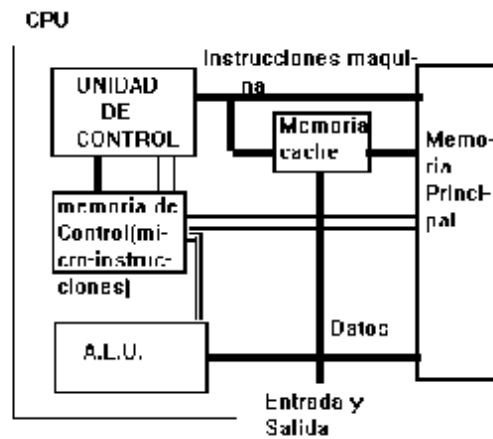
- 1)- Un mismo programa escrito en lenguaje de alto nivel, tenía muchas posibles traducciones a lenguaje máquina.
- 2)- Los diseñadores de los compiladores empleaban conjuntos reducidos de instrucciones máquina.
- 3)- Ciertas macroinstrucciones específicas apenas se empleaban.
- 4)- Aproximadamente el 50% del tiempo lo invierte la CPU en ejecutar instrucciones simples del tipo: CARGA-ALMACENAMIENTO Y BIFURCACION.

La frecuencia de utilización de las instrucciones máquina se puede analizar desde dos aspectos:

- a)- Análisis estadístico: cuando se estudian los listados de los programas y las instrucciones que aparecen en ellos.
- b)- Análisis dinámico: se tiene en cuenta las instrucciones que se van realizando en la ejecución de los programas, no las que aparecen en los listados. Téngase en cuenta que una instrucción de BIFURCACIÓN condicional de un bucle, aunque solo aparezca una vez en el listado del programa, se ejecuta muchas veces. Han existido investigadores que, sobre diferentes modelos de máquina y de programas, han estudiado la frecuencia de utilización de las instrucciones máquina.

Una conclusión generalizada es que la mitad de las instrucciones máquina se usan menos del 2%, por lo que podían eliminarse sin afectar prácticamente, al rendimiento. Dichas instrucciones solo son eficaces en aplicaciones específicas.

FIGURA 2: La memoria cache almacena la información más utilizada por la memoria principal.



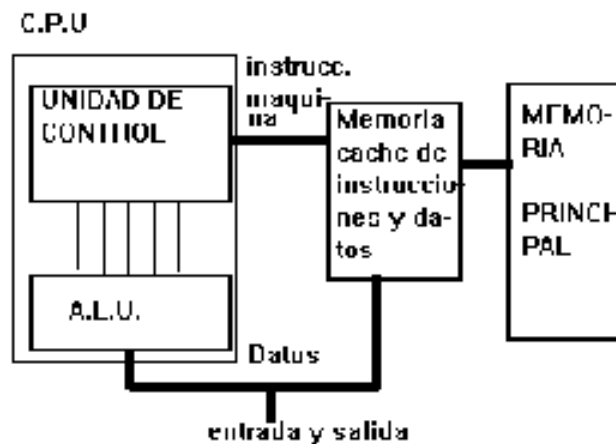
#### CUARTA ETAPA

Con el fin de mejorar la velocidad de la CPU y equilibrarla con la de la memoria cache, se han adoptado los siguientes criterios en el diseño de la arquitectura de las computadoras:

- 1)- Eliminación de la microcodificación. Las instrucciones serían del tipo elemental, no existiendo instrucciones complejas.
- 2)- Reducción del tiempo del ciclo máquina, como la consecuencia de la simplificación de las instrucciones.
- 3)- Interpretación directa de las instrucciones por el hardware y ejecución de cada una de ellas en un solo ciclo máquina.
- 4)- Selección del mínimo número de instrucciones simples.

Con este nuevo enfoque surgieron los computadores RISC, (Computadores de Conjuntos de Instrucciones Reducidos), cuya arquitectura básica correspondía al diagrama de la figura 1, aunque con la inclusión de una memoria cache, se ve en FIGURA 3.

FIGURA 3: Diagrama general por bloques de los computadores RISC en los que desaparece la Memoria de Control.



Los computadores RISC solo disponen de las instrucciones máquina más importantes y usadas, que, generalmente, suelen ser menos de 50.

Como, por otra parte, son sencillas y se llevan a cabo mediante lógica cableada, y no mediante microinstrucciones almacenadas, se pueden ejecutar en un ciclo máquina. En este tipo de computadores se ha incrementado el número de registros de propósito general para evitar accesos a memoria y ayudar a los

compiladores a analizar los datos y controlar los flujos de forma óptima al conocerse las variables más usadas que se han asignado a los registros.

Las instrucciones RISC tienen normalizados tanto su formato como su longitud, lo que favorece el funcionamiento segmentado o PIPELINE, al estar ubicados todos sus campos en sitios determinados. Sin embargo, dada la limitación del juego de instrucciones RISC únicamente a las funciones elementales, requieren empleo de coprocesadores especializados.

Universidades como las de Berkeley y Stanford y empresas como IBA, Nixdorf, Hewlett Parkard, Bull, Acorn, Inmos, metaforth, disponen de máquinas RISC con amplias perspectivas de futuro.

### **FRECUENCIA DE UTILIZACIÓN DE LAS INSTRUCCIONES**

Una de las principales características de la arquitectura de un computadora es su juego de instrucciones. Se han realizado múltiples estudios sobre la utilización de las instrucciones en los computadores comerciales, con objeto de optimizar futuros diseños.

Estos estudios se orientan al análisis de la FRECUENCIA DE UTILIZACIÓN de las instrucciones, e, incluso, las secuencias de instrucciones más frecuentes, que podrían ser integradas en una sola instrucción.

El análisis de la frecuencia de utilización de las instrucciones pueden ser ESTÁTICO O DINÁMICO. En el primer caso, se consideran, únicamente, los listados de los programas. En el segundo, se analizan las instrucciones que se van leyendo al ejecutar los programas.

Las medidas estáticas evalúan las necesidades de almacenamiento de los programas y las dinámicas la cantidad de accesos a memoria que hay que realizar para ejecutar un programa.

### **CONCLUSIONES DE LOS ESTUDIOS REALIZADOS:**

a)- La mitad de las instrucciones se dedican a mover informaciones en el computador. Es el grupo más importante de instrucciones.

b)- Las BIFURCACIONES constituyen el segundo grupo de instrucciones más empleado. Ello se debe a que una gran cantidad de los bucles empleados son muy cortos.

c)- Aproximadamente el 50% de las instrucciones de los repertorios se usan menos de 2%. Esto hace suponer que dichas instrucciones se podrían eliminar sin afectar a las prestaciones de la máquina. Pero algunas de esas instrucciones pueden ser importantes en algunas aplicaciones concretas.

Hasta la década del 80 se ha propugnado por un rico y variado juego de instrucciones como exponente de una mejor arquitectura.

Esta tendencia se basaba en los siguientes hechos:

1)- Un amplio juego de instrucciones facilita el diseño de compiladores.

2)- Los grandes juegos de instrucciones reducen el tamaño de los programas.

3)- La microprogramación y el precio decreciente de las memorias, permiten grandes juegos de instrucciones a precio reducido.

4)- Como las microinstrucciones son más rápidas que instrucciones, el poner muchas funciones en forma de microprogramas hace a la máquina más rápida.

5)- El empleo de modelo de ejecución a pila es superior al de registros.

### **ARQUITECTURA DE ORDENADORES.**

**RISC:**(Reduced Instruction Set Computer). Intenta incorporar en la máquina en forma de circuito solo aquellos conceptos lógicos que son necesarios para tener un sistema completo o altamente conveniente a causa de la frecuencia con la que ocurren, aumentando de esta manera la sencillez y velocidad del procesador. Aún existen procesadores RISC "puros" que ofrecen soluciones informáticas de gran importancia.

**CISC:**(Complex Instruction Set Computer). Imperante en el mercado hasta el momento. Se basa en la creencia que un procesador debe proveer el mayor número posible de instrucciones complejas, con el fin

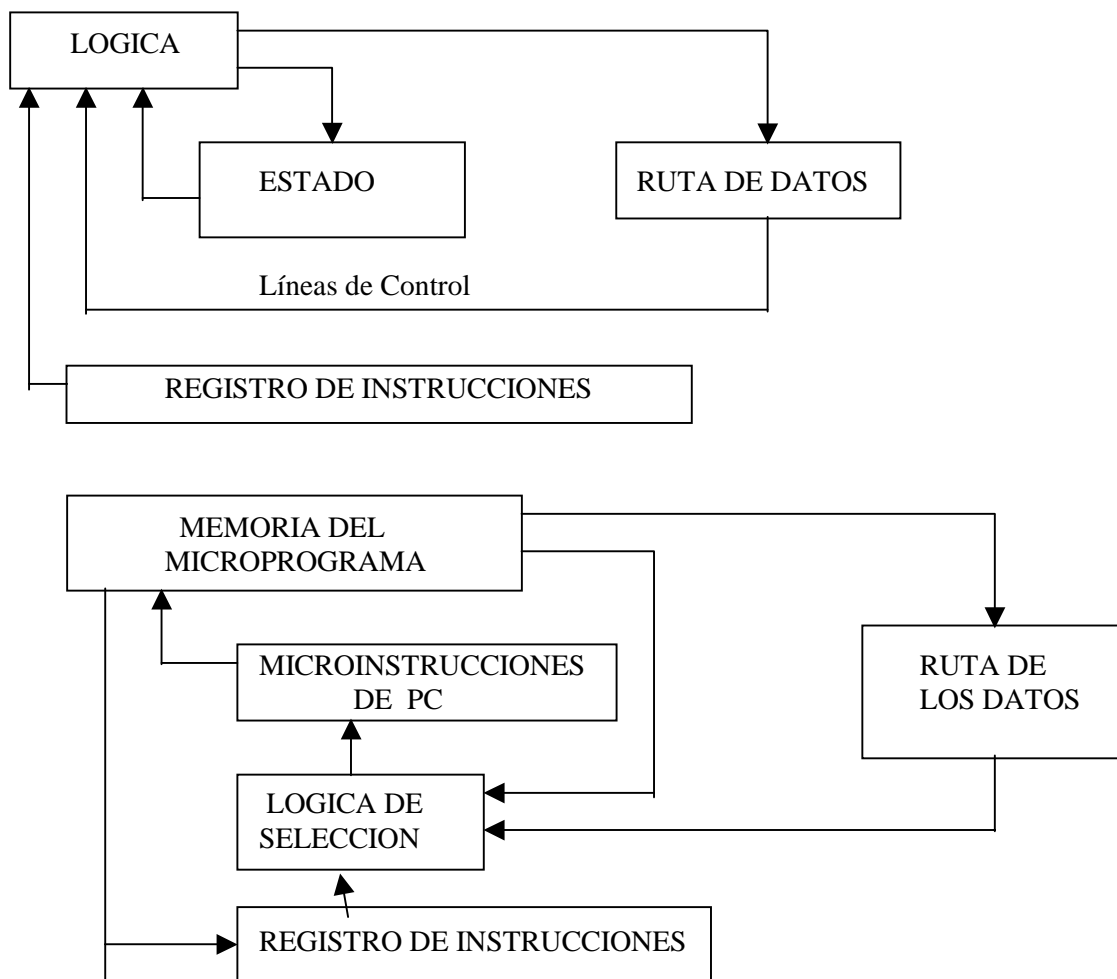
de disminuir la brecha entre los lenguajes de alto nivel y las instrucciones del procesador, para así lograr reducir el número de instrucciones ejecutadas.

### ARGUMENTOS PRESENTADOS A FAVOR DE ARQUITECTURA RISC

- 1)- El uso de memorias principales de semiconductores hace que la relación de velocidades de la memoria principal y la CPU sea parecida.
- 2)- La inclusión de la memoria cache mejora la velocidad de la memoria principal, de forma que el tiempo de acceso a esta memoria se hace equivalente al de la Memoria de Control.
- 3)- Convirtiendo la Memoria de Control en memoria rápida, tipo cache, o de almacenamiento temporal de datos, se puede reducir el coste del sistema y/o aumentar su velocidad.
- 4)- El diseño de compiladores es más fácil para juegos de instrucciones reducidos. Porque las nuevas técnicas de optimización de compiladores hacen que, difícilmente se justifiquen instrucciones complejas y, porque un juego de instrucciones sencillo y ortogonal simplifica el diseño, al limitar los casos a considerar.
- 5)- El coste de programar una función es mucho menor que de microprogramarla. Además su modificación es sencilla.

**RISC y CISC:** son dos tipos de tecnologías utilizadas para el diseño de microprocesadores.

### TECNOLOGIA RISC



### DIFERENCIAS ENTRE ARQUITECTURAS RISC Y CISC

La diferencia entre las dos es el control por microcódigo (CISC) y el control de cableado (RISC).

Los últimos lineamientos en el desarrollo de los microprocesador de la X86 (INTEL) abandonaron estos ejes para dar paso a un nuevo concepto:

"LA TECNOLOGÍA HÍBRIDA".

**CISC y RISC** : Pueden ser entendidos como dos filosofías totalmente diferentes y opuestas para la construcción de microprocesadores, aunque ambas tienen como objetivo el soporte de lenguajes de programación de alto nivel. Los últimos microprocesadores de la línea INTEL X86 no pueden ser inscriptos bajo ninguna de estas 2 consideraciones, sino que administran las ventajas que cada una de ellas brindan y logran un nuevo modelo de arquitectura. Desde el surgimiento de los procesadores Pentium y toda la gama de soluciones de quinta generación se reconocen bajo la denominación de "Procesadores Híbridos" desapareciendo por completo del horizonte del usuario el debate económico entre RISC y CISC.

**CONSIDERACIONES MAS IMPORTANTES DE LOS NUEVOS  
"HÍBRIDOS" QUE HACEN PROPIAS LAS TECNOLOGÍAS RISC Y CISC**

**CISC (Complex Instruction Set Computer).**

**⇒ MAS COMPLEJIDAD Y MEJOR PERFORMANCE.**

Es un enfoque tradicional en el diseño de procesadores. Con el mecanismo de microcódigo en el procesador (generalmente una ROM). Cada instrucción ASSEMBLER se traduce previamente a su ejecución en un conjunto de instrucciones más simple.

**Causa:** Porque los accesos a memoria eran mas lentos que los tiempos de ejecución internos de la C.P.U., por lo tanto se disminuía el número de instrucciones por programa.

**CARACTERÍSTICAS:**

Permite agregar nuevas instrucciones complejas sin perder las anteriores.

**RISC(Reduced Intrucciones Set Computer)**

**⇒ MENOR COMPLEJIDAD MEJOR PERFORMAN.**

Surgen por el avance tecnológico en la arquitectura de sistemas.

**CARACTERÍSTICAS:**

La disminución del tiempo de acceso a memoria externa de los procesadores, a través de los avances en circuitos semiconductores y la utilización de caches de alta velocidad y la mejora en el software de análisis en el comportamiento de los compiladores permite la construcción de circuitos más sencillos y veloces dejando el "trabajo" más pesado de codificación de instrucciones de alto nivel al software.

Por la sencillez de estos procesadores, su diseño se limita al análisis de las instrucciones más importantes de un lenguaje o aplicación por su cableado en el chips.

**Objetivo:** Lograr reducir al mínimo el número de instrucciones por ciclo reloj, aun a costo de tener que incrementar el número de instrucciones por unidad de programación.

**ESTO SE LOGRA POR LAS SIGUIENTES OPCIONES:**

- Conjunto reducido de instrucciones: Esto permite reducir la complejidad de chips.
- Instrucciones simples y de largo fijo: Permite que las instrucciones puedan ejecutarse en un solo ciclo de reloj.
- Minimización del número de operaciones que acceden a memoria externa: Se limitaba acceso a las operaciones load/store (carga / almacenamiento), realizando todas las demás sobre registro (tanto operando como resultados).
- Utilización de Pipelines: Para la ejecución de instrucciones en lugar del control por medio de un microcódigo típico de CISC.



## PIPELINES ¿RISC O CISC?

Técnica que surgió para poder acercar a un número de instrucciones ejecutadas por ciclo reloj permitiendo la ejecución concurrente de varias instrucciones (técnica utilizada por microprocesadores que usan microcodigo INTEL 486, PENTIUM, PENTIUM PRO Y X86)

En esta técnica, eminentemente RISC, cada instrucción se divide en Segmentos y cada uno de estos se conectan de forma de poder ejecutar concurrentemente diferentes de estas etapa de cada instrucción. En cada ciclo de reloj comienza una instrucción termina otra tanto, dependiendo del número de pasos, otras se están ejecutando. El tiempo de ejecución de un pipeline esta determinado por la mas lenta de sus etapas, y por lo tanto es necesario, en el diseño, balancear los diferentes pasos de ejecución de cada instrucción.

Por ejemplo en un pipeline de cuatro pasos se sigue el siguiente modelo para la ejecución de cada instrucción:

- 1)Carga y decodificación de la instrucción .
- 2)Lectura de los operandos del archivos de registros.
- 3)Ejecución de una operación de lógica aritmética, con los operandos.
- 4)Escritura del resultado en el archivo de registros.

Para la realización es necesario contar con, al menos, tres buses : dos para la lectura simultánea de los operandos y otro para la escritura, siempre sobre archivo de registros. En el modelo, pueden haber algunas variantes en la ejecución, como por ejemplo la necesidad de acceso a memoria en el caso de operaciones de carga o almacenamiento. En tal caso, la ejecución necesitará un ciclo adicional de reloj.

Con este esquema, inmediatamente surge el problema de que hacer en el caso de que dos instrucciones consecutivas actúen de forma tal que la segunda utilice el registro de resultado de la primera. Esto se soluciona generalmente realizando un “forwarding” interno o en el caso de MIPS, por ejemplo directamente no se tiene en cuenta esta dificultad (Scheduling estático), dejando el trabajo de organización al compilador (MIPS quiere decir, precisamente, Machine Without Interlocked Pipeline Stages), aunque en últimas implementaciones esto se ha dejado, en parte, de lado.

**CONCLUSIÓN:** Técnica RISC cuya instrucciones se dividen en segmentos conectados para que puedan ejecutar concurrentemente diferentes etapas en cada instrucción (en cada ciclo de reloj comienza una instrucción y termina otra, mientras que dependiendo del número de pasos se están ejecutando otras).

## OTROS RISC

**POWER PC:** Fue el más publicitado de los procesadores RISC de la actualidad, su mercado sigue muy fragmentado, con diversos procesadores y arquitecturas, así como distintas metas e implementaciones.

**SPARC (SCALABLE PROCESSOR ARCHITECTURE)** Arquitectura de procesadores escalares).Diseñado por SUN MICROSYTEM 1987 para su propio uso, debido a que los productores de chips se retrasan en la liberación al mercado de los “CHIPS RISC” que prometían una nueva época en velocidad.

**Características SPARC:** No es considerado como un procesador mas, sino mas bien una arquitectura.

Respeto en extremo las directivas del diseño basándose en los aspectos referidos a la limitación de los accesos a memoria a simples operaciones de carga, almacenamiento y eliminando las operaciones de multiplicación y división de ciclo múltiple.

**Característica importante:** Utilización de ventanas de registros para la fácil implementación de llamadas a funciones sin tener que acceder a memoria.

**MIPS:**La familia RX000 de procesadores MIPS (Microprocessor Without Interlocked Pipeline Stage). Hoy constituye una alternativa a la familia de procesadores basada en los proyectos RISC de Berkeley.

**R2000:** Fue el primero de la familia. Se caracteriza por la ausencia de control en el pipeline. Facilitando el diseño del microprocesador. Por lo tanto se deja al compilador la tarea de evitar instrucciones que por ejemplo utilicen como operando el resultado generado por la instrucción inmediatamente anterior. Por lo tanto solamente las instrucciones de ciclo único pueden acceder a los 32 registros y las operaciones de carga y almacenamiento deberán tomar un ciclo mas del tiempo de procesamiento.

El **R2000**(procesador) implementado debido a la frecuencia de ocurrencia, la multiplicación y la división de ciclo múltiple, debiendo existir controles de dependencia de instrucciones sucesivas desarrollados desde el hardware (la ausencia generaría problemas de reconocimiento).

**R3000** evidencio grandes mejoras en la cache de control. En lugar de utilizar ventanas de registros, solamente disponía de 32 registros para la multiplicación y división.

Su pipeline cuenta con 5 pasos:

- Carga de instrucción.
- Lectura de operandos de registros.
- operación en la unidad aritmético - lógica.
- Acceso a memoria.
- Escritura de registros.

En 1994 fue introducido **R8000** optimizada para la operación de punto fijo superescalar y con coprocesador independiente y concurrente con unidad entera.

**R10000** promete una unidad de punto fijo incorporado en un chip, así, como técnicas superescalares con cuatro operaciones concurrentes.

Los microprocesadores que utilizan estos procesadores MIPS es el Silicon Graphics Indy dedicados a sistemas de alta performance visual y desarrollos multimedia profesionales.

#### • **HEWLETT PACKARD PA-RISC.**

**PA-RISC:** Con un número bastante grande de instrucciones fue definido por HEWLETT PACKARD para sus estaciones de trabajo y minicomputadoras HP/UX UNIX. Sus inicios fueron anteriores a la popularidad de RISC.

Su diseño es extremadamente sencillo y cuenta con un pipeline de 5 pasos, con controles de dependencia de instrucciones implementados vía hardware. También posee una arquitectura Harvard (con buses de datos y de instrucciones implementados por separado) característica que permite la realización de búsqueda de instrucciones, lectura y escritura en memoria de manera concurrente.

**PA-RISC 7200:** Introducido en 1994 incluye segunda unidad entera.

**PA-RISC 8000:** Expande la arquitectura a 64 bits eliminando la segmentación de memoria.

Con sus características técnicas intentara competir HEWLETT PACKARD junto con INTEL planean diseñar chips en la tecnología VLIW (instrucciones extremadamente cortas). Las máquinas de la serie PA-RISC como por ejemplo la HEWLETT PACKARD H90.000 modelo 712 son equipos con una fuerte orientación hacia los gráficos y una muy buena performance en tareas de propósito general.

## **BIBLIOGRAFÍA**

“Introducción a la Arquitectura de Computadores” Pedro de Miguel Anasagasti - José Mario Angulo

Usategui. Editorial: Paraminfo

“Revista: “CompuMagazine”. Año IX / N°90.