



Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

ARQUITECTURA DE COMPUTADORAS

Grupo: 004

Hora: N4

ACTIVIDAD FUNDAMENTAL 3

createElement

- 5. Arquitectura SISC, RISC y Pipeline**
- 6. Taxonomía de Flynn, multiprocesadores y multicomputadoras**

Alumno: IRMA RAQUEL REYES GUTIERREZ

Matrícula: 2107318

Carrera: ITS

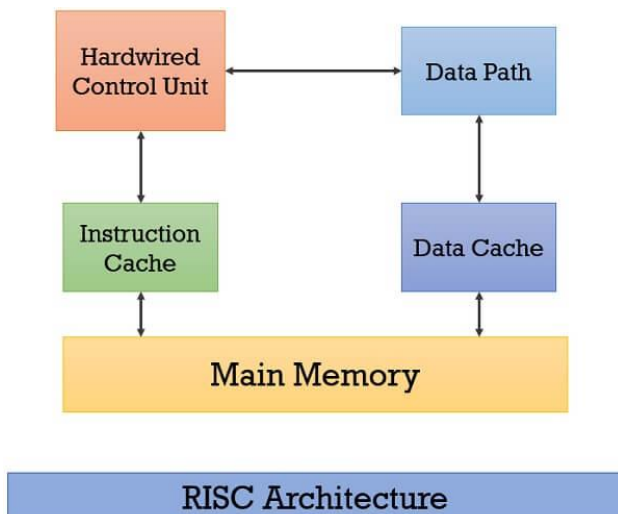
Cesar Yael Santillan Marroquin

04 DE MAYO DE 2025

Arquitectura CISC, RISC y Pipeline

Arquitectura CISC (Complex Instruction Set Computer)

La arquitectura CISC, siglas en inglés de *Complex Instruction Set Computer*, se caracteriza por contar con un conjunto de instrucciones amplio y complejo. Esto significa que los procesadores diseñados bajo este modelo pueden ejecutar una gran variedad de instrucciones, algunas de las cuales son bastante sofisticadas y permiten realizar tareas complejas con una sola línea de código. Esta arquitectura fue concebida en una época donde el costo de la memoria era elevado y los lenguajes de programación estaban menos desarrollados, por lo que simplificar el código fuente era una prioridad. La idea era que el compilador o el programador pudiera hacer más con menos líneas de código, lo que hacía que la programación fuese más directa y sencilla, especialmente en lenguajes de bajo nivel como el ensamblador. Cada instrucción podía realizar múltiples operaciones, como acceder a la memoria y realizar cálculos al mismo tiempo, lo cual facilitaba la escritura de software.



Un ejemplo clásico y ampliamente utilizado de procesadores que emplean arquitectura CISC es la familia Intel x86. Estos procesadores se encuentran en la mayoría de las computadoras personales y laptops desde hace décadas. También algunos procesadores de AMD comparten esta misma arquitectura. Uno de los aspectos distintivos de CISC es su complejidad interna: para poder interpretar y ejecutar esas instrucciones complejas, el hardware debe ser más elaborado, lo que incrementa la cantidad de circuitos y lógica interna. Esto a su vez puede afectar negativamente la eficiencia energética y la velocidad en ciertas operaciones, ya que decodificar y ejecutar instrucciones más pesadas toma más tiempo que instrucciones más simples.

-Ventajas

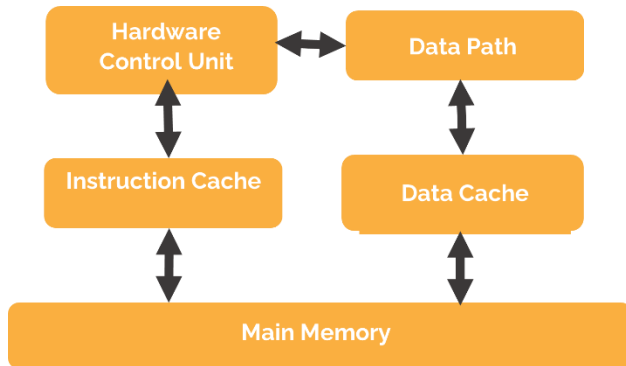
Entre las ventajas de esta arquitectura se destaca la simplicidad en la programación, ya que una sola instrucción puede traducirse en múltiples acciones, reduciendo la carga para el programador y haciendo que el código fuente sea más compacto.

-Desventaja

Sin embargo, entre las desventajas se encuentra su mayor complejidad a nivel de hardware, lo cual puede dificultar su diseño, incrementar el costo de fabricación y consumir más energía. Además, en muchos casos, los procesadores CISC no logran la misma velocidad de procesamiento que otras arquitecturas más simples cuando se

trata de ejecutar instrucciones de forma continua y eficiente.

Arquitectura RISC (Reduced Instruction Set Computer)



La arquitectura RISC, o *Reduced Instruction Set Computer*, sigue un enfoque completamente diferente al de CISC. En lugar de contar con un conjunto extenso de instrucciones complejas, RISC utiliza un conjunto reducido de instrucciones simples que pueden ejecutarse muy rápidamente, generalmente en un solo ciclo de reloj. Este diseño favorece la simplicidad tanto en el hardware como en la ejecución de instrucciones, y está optimizado para permitir que el procesador realice tareas repetitivas de forma muy eficiente. Cada instrucción está diseñada para realizar una tarea específica de manera rápida y directa, lo cual facilita su ejecución y también su decodificación por parte del procesador.

Procesadores conocidos que utilizan arquitectura RISC son los ARM, que se encuentran en la mayoría de los



dispositivos móviles como teléfonos inteligentes y tabletas, y los PowerPC, que fueron utilizados en algunas computadoras Apple antiguas, consolas de videojuegos y sistemas embebidos. La arquitectura RISC

se ha vuelto cada vez más popular en entornos donde la eficiencia energética es clave, como en dispositivos portátiles, donde la duración de la batería es una prioridad. Debido a su diseño simplificado, estos procesadores suelen consumir menos energía y generar menos calor, lo que permite construir dispositivos más pequeños y eficientes.

-Ventajas

Entre las ventajas principales de RISC se encuentra su mayor eficiencia energética, así como su capacidad para lograr una ejecución más rápida de instrucciones, lo que mejora el rendimiento general del sistema. También, al tener instrucciones más simples, el diseño del procesador es menos costoso y más fácil de optimizar.

-Desventaja

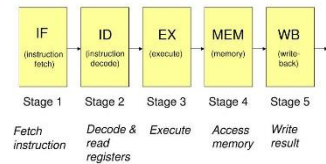
Sin embargo, una desventaja notable es que para realizar tareas complejas, el programa puede necesitar muchas más instrucciones que en una arquitectura CISC, lo cual puede incrementar el tamaño del código fuente y requerir más ciclos de procesamiento en algunos casos, aunque esta desventaja puede ser compensada con la velocidad y la eficiencia del hardware RISC.

Pipeline

El *pipeline* es una técnica fundamental en el diseño de procesadores modernos que permite mejorar el rendimiento general del sistema. Consiste en **dividir la ejecución de una instrucción en varias etapas** que se realizan de forma paralela. Las etapas típicas incluyen la **búsqueda de la instrucción** en la memoria, su **decodificación**, la **ejecución** de la operación, el **acceso a la memoria** si es necesario y finalmente la **escritura del resultado**. En lugar de esperar a que una instrucción se complete totalmente antes de empezar la siguiente, el pipeline permite que, mientras una instrucción está siendo

ejecutada, la siguiente ya esté siendo decodificada y otra más esté siendo buscada en memoria, como una línea de producción en una fábrica.

Simple Processor Pipeline



© 2000 Cowart F. Gehring ECE 465/521 Lecture Notes, Spring 2008

significativamente la velocidad de procesamiento, ya que se pueden tener múltiples instrucciones en diferentes fases de ejecución al mismo tiempo. El resultado es una mejora considerable del **rendimiento** general del procesador, ya que se aprovecha mejor el tiempo de cada unidad funcional. El uso de pipeline es especialmente efectivo en arquitecturas RISC, donde las instrucciones son uniformes y simples, lo que permite una segmentación clara y fluida.

-Desventaja

No obstante, el pipeline no está exento de **problemas**. Uno de los más comunes son los **conflictos de datos**, que ocurren cuando una instrucción necesita un dato que aún está siendo calculado por una instrucción anterior. También hay **conflictos de control**, que suceden por ejemplo cuando hay saltos condicionales en el código, y el procesador no puede saber con certeza cuál será la próxima instrucción hasta que se evalúe la condición. Para resolver estos problemas, los procesadores modernos incorporan técnicas adicionales como el reordenamiento de instrucciones, la predicción de saltos y el uso de buffers. A pesar de estas complicaciones, el pipeline sigue siendo una técnica esencial para lograr un procesamiento eficiente en la mayoría de los procesadores actuales.

-Ventajas

Esta técnica tiene la ventaja de **incrementar**

Taxonomía de Flynn

La **Taxonomía de Flynn** es una clasificación propuesta por el científico informático Michael J. Flynn en 1966, que categoriza las arquitecturas de computadoras según la cantidad de **flujos de instrucciones** y **flujos de datos** que procesan de manera simultánea. Esta clasificación permite entender cómo se

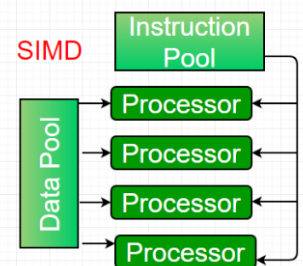
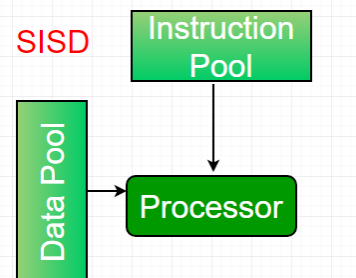
organizan y funcionan los sistemas de procesamiento en términos de paralelismo.

Flynn estableció cuatro categorías

principales: **SISD**, **SIMD**, **MISD** y **MIMD**.

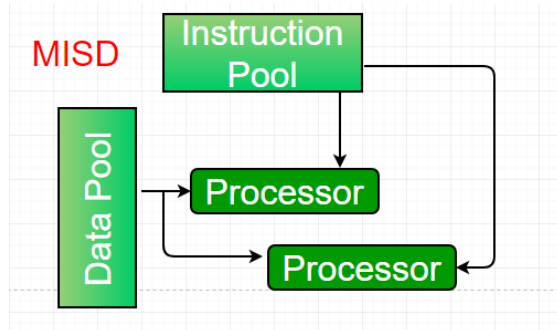
- La primera categoría, **SISD (Single Instruction, Single Data)**, se refiere a las arquitecturas tradicionales, donde un único procesador ejecuta una sola instrucción sobre un único conjunto de datos a la vez. Es el modelo clásico de las computadoras secuenciales, como las computadoras personales más básicas o los primeros procesadores. Aunque son simples y fáciles de programar, tienen un rendimiento limitado ya que no aprovechan el procesamiento paralelo.

- La segunda categoría, **SIMD (Single Instruction, Multiple Data)**, es utilizada en sistemas donde **una sola instrucción se aplica simultáneamente a múltiples datos**. Este tipo de arquitectura es

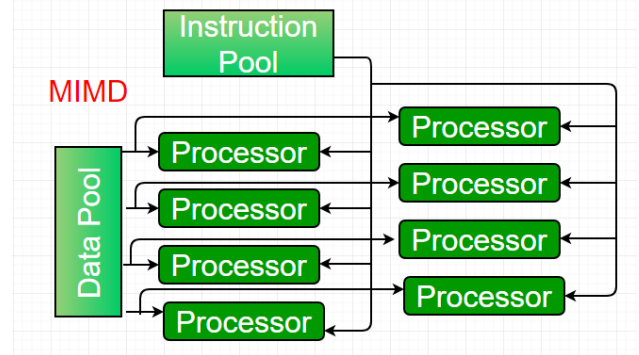


ideal para operaciones repetitivas sobre grandes volúmenes de datos, como el procesamiento de imágenes, gráficos y señales. SIMD se encuentra comúnmente en tarjetas gráficas (GPU) y en ciertas extensiones de procesadores modernos (como SSE y AVX en Intel). Su principal ventaja es la rapidez en tareas que requieren aplicar la misma operación sobre diferentes elementos de forma paralela.

- La tercera categoría, **MISD (Multiple Instruction, Single Data)**, es bastante inusual en la práctica. En teoría, describe sistemas donde múltiples unidades ejecutan diferentes instrucciones sobre el mismo dato. No se utiliza en la mayoría de los sistemas computacionales debido a su complejidad y limitada aplicabilidad. Sin embargo, se ha mencionado en contextos muy específicos como sistemas de control redundante o tolerancia a fallos.



- Finalmente, **MIMD (Multiple Instruction, Multiple Data)** representa la forma más común y poderosa de procesamiento paralelo en los sistemas modernos. En esta arquitectura, diferentes procesadores ejecutan diferentes instrucciones sobre distintos conjuntos de datos, lo cual permite realizar múltiples tareas completamente independientes al



mismo tiempo. Esta categoría abarca los sistemas **multiprocesadores**, **multicomputadoras** y muchos entornos de **computación distribuida**, siendo una base fundamental para los centros de datos, servidores, supercomputadoras y computación en la nube.

Multiprocesadores

Los **multiprocesadores** son sistemas que integran **dos o más unidades centrales de procesamiento (CPUs)** dentro de un mismo equipo o sistema. Estas CPUs comparten la misma memoria principal y recursos del sistema, lo que permite que trabajen en colaboración para ejecutar distintas tareas al mismo tiempo. Gracias a este diseño, los multiprocesadores pueden manejar cargas de trabajo intensivas de manera más eficiente que un sistema con un solo procesador, ya que permiten la **ejecución simultánea de procesos** o hilos de ejecución (*threads*), lo cual mejora considerablemente el **rendimiento** general del sistema.



Ventajas

Una ventaja fundamental de los multiprocesadores es la **capacidad de realizar procesamiento paralelo**, lo que reduce los tiempos de espera y mejora la eficiencia, especialmente en aplicaciones que requieren alto poder de cómputo como la simulación científica, el modelado 3D o los servidores web. También ofrecen **mayor tolerancia a fallos**, ya que si un procesador se detiene, los otros pueden continuar funcionando. Un ejemplo típico de uso de multiprocesadores son los **servidores empresariales**, estaciones de trabajo avanzadas para diseño gráfico o ingeniería, y algunas supercomputadoras pequeñas que necesitan ejecutar múltiples tareas complejas al mismo tiempo.

Multicomputadoras



A diferencia de los multiprocesadores, las **multicomputadoras** están formadas por múltiples **computadoras independientes** que se encuentran **conectadas entre sí a través de una red**, y que colaboran para resolver problemas comunes. Cada computadora tiene su propio procesador y memoria, y la comunicación entre ellas se realiza a través de mensajes, en lugar de compartir memoria directamente. Este modelo es fundamental en el diseño de **sistemas distribuidos** y se emplea ampliamente en aplicaciones que requieren **alto rendimiento** o **gran capacidad de procesamiento**.

Los sistemas de multicomputadoras se

utilizan en estructuras como los **clústeres de computación**,

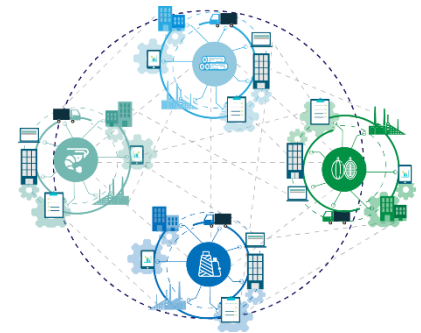
donde varias máquinas trabajan en conjunto para simular fenómenos científicos,



procesar grandes volúmenes de datos o renderizar animaciones complejas. También son la base de la **computación en la nube**, donde los recursos se distribuyen entre múltiples servidores remotos que trabajan en paralelo. Su principal ventaja es la escalabilidad: se pueden añadir más computadoras a la red para incrementar el poder de procesamiento. Además, al no depender de un solo sistema físico, ofrecen **mayor flexibilidad** y pueden funcionar de forma colaborativa en entornos geográficamente dispersos.

Aplicaciones prácticas

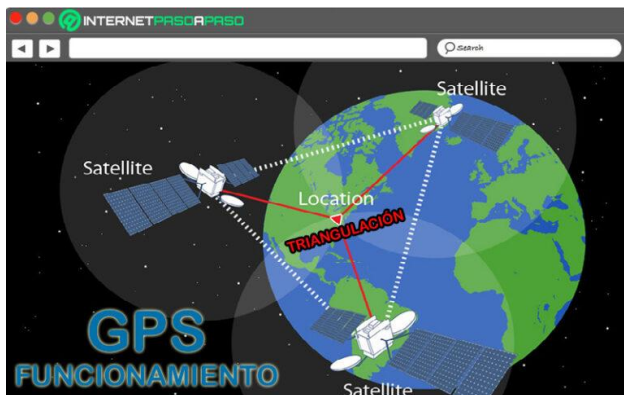
Las arquitecturas y tecnologías mencionadas se aplican en **diversos campos tecnológicos**, muchas veces de forma



combinada, como ocurre en sistemas avanzados de navegación, reconocimiento y seguridad. Un ejemplo práctico de estas aplicaciones son los **clústeres PDA y sistemas GPS**. En el caso de los clústeres PDA, se utilizan arquitecturas paralelas como MIMD o entornos multicomputadora para procesar información en tiempo real sobre ubicación, mapas, sensores y conexión a internet, permitiendo una

navegación eficiente y rápida. Por su parte, los sistemas GPS modernos integran múltiples procesadores para calcular la ubicación precisa del usuario y combinarla con mapas digitales y datos adicionales, todo en cuestión de segundos.

Otra área destacada donde se aplican estas tecnologías es en los **sistemas biométricos**, que utilizan características únicas del cuerpo humano para identificar o autenticar a las personas. Entre los ejemplos más conocidos están el **reconocimiento de huellas digitales, voz, rostro o retina**. Estos sistemas requieren procesar imágenes, analizar patrones y



comparar resultados con bases de datos en fracciones de segundo. Para lograrlo de manera eficiente y segura, se emplean procesadores RISC o multiprocesadores, además de arquitecturas paralelas para gestionar el volumen de información y tomar decisiones con rapidez. En el ámbito de la **seguridad informática**, los sistemas biométricos están cada vez más integrados en dispositivos como smartphones, computadoras y sistemas de control de acceso, y permiten reemplazar contraseñas tradicionales por métodos de autenticación más seguros y personalizados.

Conclusion

Las arquitecturas CISC y RISC representan enfoques distintos para diseñar procesadores. Mientras CISC busca simplificar la programación mediante

instrucciones complejas, RISC apuesta por instrucciones simples que se ejecutan rápidamente, permitiendo el uso más eficiente de técnicas como el Pipeline, que incrementa el rendimiento dividiendo las instrucciones en etapas. Esta mejora en el nivel del procesador individual es fundamental cuando se aplican estos procesadores dentro de entornos paralelos o distribuidos, como los descritos por la Taxonomía de Flynn.

Por ejemplo, los sistemas MIMD, que procesan múltiples instrucciones y múltiples datos simultáneamente, suelen estar compuestos por varios procesadores (ya sean RISC o CISC), organizados en multiprocesadores o multicomputadoras. El uso de arquitecturas eficientes y pipelines en cada procesador contribuye al rendimiento global del sistema. Así, la capacidad de ejecutar muchas tareas al mismo tiempo a gran escala depende directamente de cómo están diseñados los procesadores a nivel interno y cómo se comunican entre sí.

En conjunto, estos temas nos permiten comprender cómo el diseño interno del procesador afecta el rendimiento individual, y cómo la organización de múltiples procesadores define el rendimiento colectivo. En la era de la inteligencia artificial, la seguridad biométrica, los sistemas móviles y la computación en la nube, estas arquitecturas se combinan para dar vida a aplicaciones cada vez más rápidas, seguras y capaces. La relación entre microarquitectura (como RISC y Pipeline) y macroarquitectura (como MIMD o multiprocesadores) es hoy más relevante que nunca, ya que de su integración depende el avance tecnológico en todos los campos de la informática moderna.