

El Microprocesador

A quien no le ha pasado que cuando uno está acostumbrado al uso de nuestro smartphone y probamos otro más “potente” a veces solemos sentirlo más “lento”, es decir, que algunas aplicaciones se “traban” o se siente una sensación de lentitud en algunas funciones aun cuando el teléfono más potente tiene una velocidad mayor y hasta más núcleos.

Realmente no hay una manera fácil de decirlo sin entrar a muchos tecnicismos que aburrirían hasta el que crea los procesadores, pero vamos a tratar de explicarlo de una manera que sea lo *suficientemente técnica para aclararlo a la vez que fácil de entender para los usuarios que no les interesan datos técnicos*.

Cada procesador es diferente, y no solo a diferencias en cómo se hacen, si no a diferencias en personalizaciones como el pipeline, tamaño del bus, métodos de fabricación, materiales de fabricación, cache, set de instrucciones y todas esas cosas que el usuario final rara vez ve ya que usualmente, el marketing solo se enfoca en la velocidad de reloj (los miles de Megahertz) y el número de núcleos (single core, dual-core, quad-core, octa-core y así sucesivamente).

Ahora imagínense todas las combinaciones posibles de entre esas diferencias, cada una generando un rendimiento diferente, por mínimo que sea, es diferente y muchas veces no siempre es para bien. Usualmente, lo que importa en el número de núcleos y la velocidad de los núcleos, pero a veces el hecho de tener 2 núcleos no resulta mejor que tener un núcleo, ya que la manera en que el chip fue fabricado no fue de la mejor manera para que la arquitectura aproveche la ventaja de tener 2 núcleos. Lo mismo pasa comparando un doble núcleo contra un cuádruple núcleo.

Esto se ve más comúnmente en dispositivos de gama baja en donde los procesadores no son bien optimizados para ahorrar gastos, así que podemos ver un dual-core en un smartphone barato con un desempeño de risa, mientras que un smartphone de un solo núcleo hacer maravillas.

Pero lo anterior solo es una cara de la moneda, ya que también el software (es decir el S.O. y las apps) tiene que ser optimizado en sus líneas de programación para poder usar las capacidades de 2, 4, 8, 16, 32 o más núcleos según sea el caso. Muchas veces y sobre todo en el ambiente libre que tiene Android, vemos aplicaciones que no están bien optimizadas para el uso de varios núcleos y no es hasta que el desarrollador actualiza la aplicación con nuevas líneas de códigos de optimización para varios núcleos que cuando esa app se “siente” más rápida en nuestro recién adquirido smartphone quad-core.

Esto fue más común en el ambiente Android en donde realmente Google no tenía un control muy definido de como las apps son desarrolladas y subidas al Play Store de Android, caso contrario pasa con Apple y su iPhone en donde en la App Store, todas las apps que se suben pasan por un riguroso proceso de análisis, las cuales tienen que estar optimizadas para trabajar aprovechando la potencia del más reciente iPhone, iPad o iPod.

Como verán, es un conjunto de muchas variables el hecho de que un procesador pueda desempeñar eficazmente el rendimiento que esperamos de éste, por eso siempre es bueno comparar, comparar y comparar hasta que estemos convencidos de que lo que vamos a comprar se ajusta realmente a nuestras necesidades y expectativas. Los famosos Benchmarks o pruebas de rendimiento que vemos en videos de Youtube no son realmente un indicador si el teléfono es bueno o no. Ayudan a ver solo la potencia de un procesador, pero recordemos que el verdadero rendimiento se ven cuando todos los factores que influyen en un dispositivo se ven todos unidos.

Analizar la siguiente comparación entre Ryzen 9 9950 y Intel core i9 14900

<https://versus.com/en/amd-ryzen-9-9950x-vs-intel-core-i9-14900k>

ISA (Instruction Set Architecture, Arquitectura del Conjunto de Instrucciones)

Un conjunto de instrucciones es un grupo de comandos para una unidad central de procesamiento (CPU) en lenguaje máquina. El término puede referirse a todas las instrucciones posibles para una CPU o a un subconjunto de instrucciones para mejorar su rendimiento en determinadas situaciones.

Todas las CPU tienen conjuntos de instrucciones que permiten ejecutar comandos que dirigen a la CPU a conmutar los transistores correspondientes. Estas instrucciones le indican a la CPU qué tareas realizar. Algunas instrucciones son simples comandos de lectura , escritura y movimiento que dirigen datos a diferentes elementos de hardware.

Las instrucciones se componen de un número específico de bits . Por ejemplo, las instrucciones de la CPU pueden tener 8 bits, donde los primeros 4 bits conforman el código de operación que indica al ordenador qué hacer. Los siguientes 4 bits son el operando, que indica al ordenador los datos que deben utilizarse.

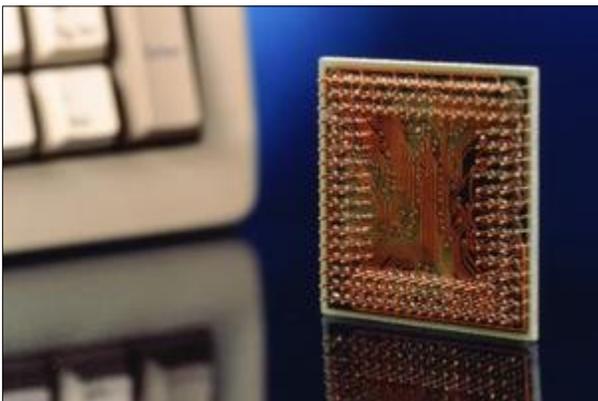
La longitud de un conjunto de instrucciones puede variar desde tan solo 4 bits hasta varios cientos. En algunas arquitecturas de conjuntos de instrucciones (ISA), las instrucciones tienen longitudes diferentes. Otras ISA tienen instrucciones de longitud fija.

https://en.wikipedia.org/wiki/Comparison_of_instruction_set_architectures

Los Núcleos

Una de las primeras decisiones a la hora de diseñar un microprocesador es decidir cuál será su juego de instrucciones. La decisión es trascendente por dos razones;

- El juego de instrucciones decide el diseño físico del conjunto;
- Cualquier operación que deba ejecutarse en el microprocesador deberá poder ser descrita en términos de un lenguaje de estas instrucciones. Frente a esta cuestión caben dos filosofías de diseño; máquinas denominadas **CISC** y máquinas denominadas **RISC**.



Cuando hablamos de microprocesadores **CISC**, *computadoras con un conjunto de instrucciones complejo*, del inglés *complex instruction set computer*, y procesadores **RISC**, *computadoras con un conjunto de instrucciones reducido*, del inglés *reduced instruction set computer*, se piensa que los atributos complejo y reducido describen las diferencias entre los dos modelos de arquitectura para microprocesadores. Esto es cierto solo de forma superficial, pues se requiere de muchas otras características esenciales para definir los **RISC** y los **CISC**. Aún más, existen diversos procesadores que no se pueden asignar con facilidad a ninguna categoría

determinada.

Hasta hace solo algunos años, la división era tajante: **RISC** se utilizaba para entornos de red, mientras que **CISC** se aplicaba en ordenadores domésticos. Pero en la actualidad se alzan voces que afirman que **CISC** está agotando sus posibilidades, mientras otras defienden fervientemente que **CISC** ya ha alcanzado a **RISC**, adoptando algunas de sus principales características.

Un Nucleo - Single Core

Hablemos un poco de los Single-Core o procesadores sencillos de 1 núcleo de procesamientos. Son buenos mientras que no sean de aplicaciones de alto performance... ¿Como así? si estos procesadores son excelentes para: navegar en internet mientras escuchas música y editas Word, power point, etc... Más o menos con estos 3 programas simultáneos el PC funciona fluido claro que ya depende del procesador y la velocidad de este.

Juegos clásicos que funcionan con un núcleo:

1. **Half-Life 2** (2004) - Un clásico de Valve que funciona bien en sistemas antiguos.
2. **The Elder Scrolls III: Morrowind** (2002) - Un RPG de mundo abierto que no requiere múltiples núcleos.
3. **Diablo II** (2000) - Un juego de acción y rol que funciona en sistemas de un solo núcleo.
4. **StarCraft** (1998) - Un juego de estrategia en tiempo real que no necesita múltiples núcleos.
5. **Age of Empires II** (1999) - Otro clásico de estrategia que funciona en sistemas antiguos.

Juegos indie o menos exigentes que funcionan con un nucleo:

1. **Undertale** (2015) - Un RPG indie que no requiere mucha potencia.
2. **Stardew Valley** (2016) - Un juego de simulación de granja que funciona bien en sistemas de un solo núcleo.
3. **Terraria** (2011) - Un juego de aventuras y construcción 2D que no necesita múltiples núcleos.
4. **Hotline Miami** (2012) - Un juego de acción rápido y frenético que no es exigente con el hardware.
5. **Cave Story+** (2011) - Un plataformero clásico que funciona en sistemas antiguos.

Juegos retro o emuladores que funcionan con un nucleo:

1. **Emuladores de consolas antiguas** (NES, SNES, Sega Genesis, etc.) - Los emuladores de consolas clásicas suelen funcionar bien con un solo núcleo.
2. **Doom** (1993) - Un clásico de los shooters en primera persona que funciona en casi cualquier sistema.
3. **Quake** (1996) - Otro clásico de los FPS que no requiere múltiples núcleos.

Dos Nucleos - Dual Core

Muy bueno para algunas aplicaciones o para pocas tareas, se destaca por poder utilizar multiproceso jugar escuchar música y navegar en internet a la vez, estos procesadores de 2 cores pueden hacer muchísimos procesos al mismo tiempo!

Mas de 4 Cores

Estos últimos ofrecen un óptimo desempeño en tareas de muchísimos sub procesos es decir con

estos puedes oír música jugar navegar en Firefox, Chrome, Editar videos e imagenes, usar Word PowerPoint y más con una fluidez espectacular...

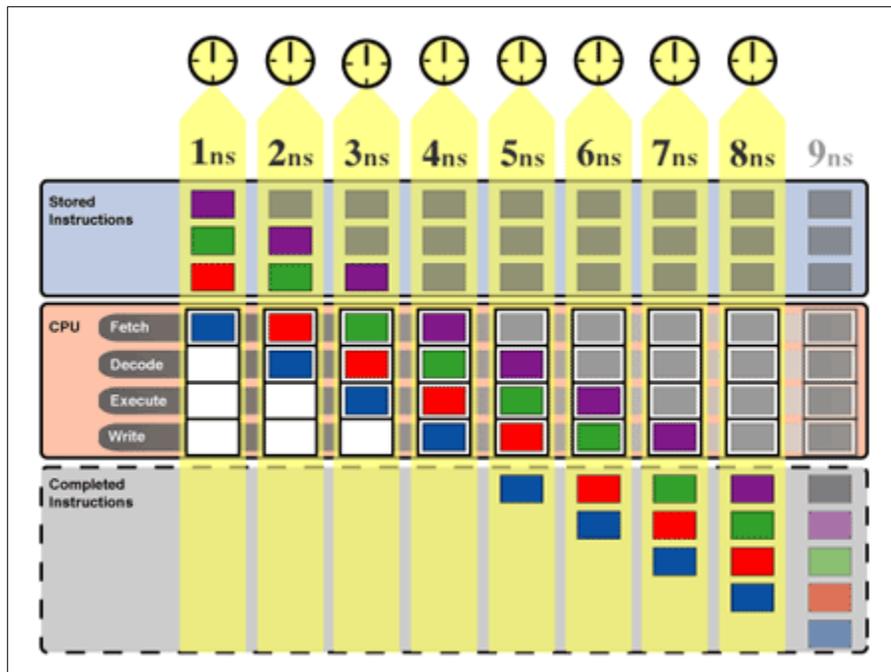
No todas las aplicaciones de software están diseñadas para tomar ventaja de las capacidades del procesador de cuatro núcleos o mas. Si la mayor parte del software que se utiliza ha sido diseñado para funcionar con procesadores de núcleo único, no puedes observar un gran aumento de rendimiento al cambiar a un procesador de cuatro núcleos. Las aplicaciones que están diseñadas para ejecutar tareas al mismo tiempo serán capaces de ejecutarlas incluso con núcleos individuales en el procesador.

¿Que es el Pipeline?

Pipeline es un término perteneciente a la ingeniería de software, y consiste en una cadena de elementos de procesamiento ordenados de tal manera que la salida de cada elemento es la entrada del siguiente, con almacenamiento temporal de datos o buffering entre los procesos.

Suena complicado pero no lo es; el nombre quiere decir en español "tuberías", y el sistema es básicamente como el agua que circula por cañerías o tubos. En este caso el agua vendría a ser la información o los procesos.

El pipeline es común verlo en sistemas operativos multitarea, como los que empleamos hoy en día en nuestras computadoras; se ejecutan una serie de procesos de manera simultánea, que son ejecutados luego de manera secuencial mediante un administrador de tareas dándoles diferente prioridad y capacidad de procesamiento, alternando entre un "pipeline" y los demás.

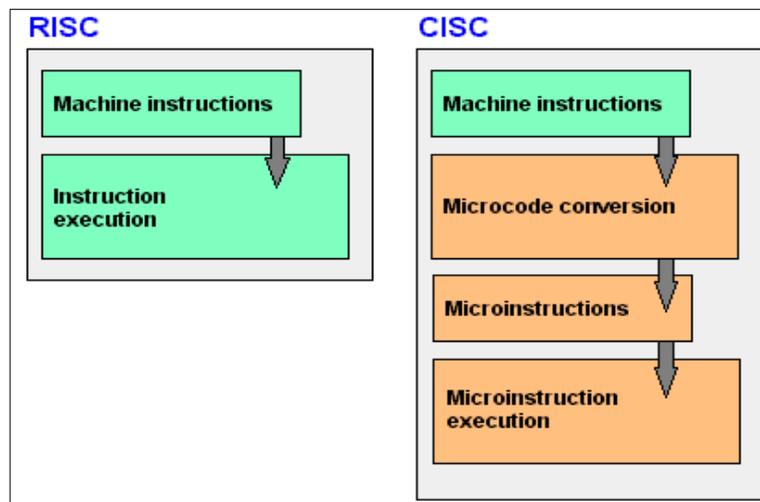


Al comienzo del primer nanosegundo, la instrucción azul entra en la etapa de fetch (traer). Comienza el segundo nanosegundo y la instrucción azul se mueve a la etapa de decodificación mientras la siguiente instrucción, la roja, se abre camino desde el almacén de código al procesador (esto es, entra a la etapa de fetch). Al comienzo del tercer nanosegundo, la instrucción azul avanza a la etapa de ejecución, la instrucción roja a la etapa de decodificación, y la instrucción verde entra a la etapa de fetch. En el

cuarto, la instrucción azul avanza a la etapa de escritura, la roja a la de ejecución, la verde a la decodificación, y la violeta a la de fetch. Cuando se ha completado el cuarto nanosegundo y comienza el quinto, la instrucción azul ha pasado a través del pipeline y ha terminado su ejecución. Así podemos decir que al final de cuatro nanosegundos (= a cuatro ciclos de reloj) el procesador con pipeline ha completado una instrucción.

Al comienzo del quinto nanosegundo, el pipeline está completo y el procesador puede empezar a completar instrucciones al ritmo de una instrucción por nanosegundo. Esta tasa de ejecución de 1 instrucción/ns es un incremento de 4x sobre la tasa del procesador de ciclo simple, que era 0.25 instrucciones/ns (o 4 instrucciones cada 16 nanosegundos)

Tecnología RISC y CISC



Tecnología CISC

La tecnología CISC (Complex Instruction Set Computer) nació de la mano de Intel, creador en 1971 del primer microchip que permitiría el nacimiento de la informática personal. Más concretamente, sería en 1972 cuando aparecería el 8080, primer chip capaz de procesar 8 bits, suficiente para representar números y letras. Con la posibilidad de colocar todos los circuitos en un solo chip y la capacidad de manejar número y letras nacería la cuarta generación de ordenadores, la de los conocidos como PC u ordenadores personales.

Los microprocesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y permitir operaciones complejas entre operandos situados en la memoria o en los registros internos.

Este tipo de arquitectura dificulta el paralelismo entre instrucciones, por lo que en la actualidad la mayoría de los sistemas CISC de alto rendimiento implementan un sistema que convierte dichas instrucciones complejas en varias instrucciones simples, llamadas generalmente microinstrucciones.

La microprogramación es una característica importante y esencial de casi todas las arquitecturas CISC. La microprogramación significa que cada instrucción de máquina es interpretada por un microprograma localizado en una memoria en el circuito integrado del procesador. Las instrucciones compuestas son decodificadas internamente y ejecutadas con una serie de microinstrucciones almacenadas en una ROM interna. Para esto se requieren de varios ciclos de reloj, al menos uno por microinstrucción. Es así entonces como los chips CISC utilizan comandos que incorporan una gran diversidad de pequeñas instrucciones para realizar una única operación.

Cuando el sistema operativo o una aplicación requiere de una de estas acciones, envía al procesador el nombre del comando para realizarla junto con el resto de información complementaria que se necesite. Pero cada uno de estos comandos de la ROM del CISC varían de tamaño y, por lo tanto, el chip debe en primer lugar verificar cuanto espacio requiere el comando para ejecutarse y poder así reservárselo en la memoria interna. Además, el procesador debe determinar la forma correcta de cargar y almacenar el comando, procesos ambos que ralentizan el rendimiento del sistema.

El procesador envía entonces el comando solicitado a una unidad que lo descodifica en instrucciones más pequeñas que podrán ser ejecutadas por un nanoprocesador, una especie de procesador dentro del procesador. Y al no ser las instrucciones independientes, pues son instrucciones menores procedentes de la descodificación de una instrucción mayor, sólo puede realizarse una instrucción cada vez.

A través de la compleja circuitería del chip, el nanoprocesador ejecuta cada una de las instrucciones del comando. El desplazamiento por esta circuitería también ralentiza el proceso. Para realizar una sola instrucción un chip CISC requiere de cuatro a diez ciclos de reloj.

Entre las bondades de CISC destacan las siguientes:

- Reduce la dificultad de crear compiladores.
- Permite reducir el costo total del sistema.
- Reduce los costos de creación de software.
- Mejora la compactación de código.
- Facilita la depuración de errores.

Ejemplo de microprocesadores basados en la tecnología CISC:

- Intel 8086, 8088, 80286, 80386, 80486.
- Motorola 68000, 68010, 68020, 68030, 6840.

Set de instrucciones de un X86:

<https://learn.microsoft.com/es-es/windows-hardware/drivers/debugger/x86-instructions>

Tecnología RISC

Buscando aumentar la velocidad del procesamiento se descubrió en base a experimentos que, con una determinada arquitectura de base, la ejecución de programas compilados directamente con microinstrucciones y residentes en memoria externa al circuito integrado resultaban ser más eficientes, gracias a que el tiempo de acceso de las memorias se fue decrementando conforme se mejoraba su tecnología de encapsulado.

Debido a que se tiene un conjunto de instrucciones simplificado, éstas se pueden implantar por hardware directamente en la CPU, lo cual elimina el microcódigo y la necesidad de decodificar instrucciones complejas.

La arquitectura RISC funciona de modo muy diferente a la CISC, su objetivo no es ahorrar esfuerzos externos por parte del software con sus accesos a la RAM, sino facilitar que las instrucciones sean ejecutadas lo más rápidamente posible. La forma de conseguirlo es simplificando el tipo de instrucciones que ejecuta el procesador. Así, las instrucciones más breves y sencillas de un procesador RISC son capaces de ejecutarse mucho más aprisa que las instrucciones más largas y complejas de un chip CISC. Sin embargo, este diseño requiere de mucha más RAM y de una tecnología de compilador más avanzada.

La relativa sencillez de la arquitectura de los procesadores RISC conduce a ciclos de diseño más cortos cuando se desarrollan nuevas versiones, lo que posibilita siempre la aplicación de las más recientes tecnologías de semiconductores. Por ello, los procesadores RISC no solo tienden a ofrecer una capacidad de procesamiento del sistema de 2 a 4 veces mayor, sino que los saltos de capacidad que se producen de generación en generación son mucho mayores que en los CISC.

Los comandos que incorpora el chip RISC en su ROM constan de varias instrucciones pequeñas que realizan una sola tarea. Las aplicaciones son aquí las encargadas de indicar al procesador qué combinación de estas instrucciones debe ejecutar para completar una operación mayor.

Además, los comandos de RISC son todos del mismo tamaño y se cargan y almacenan del mismo modo. Al ser estas instrucciones pequeñas y sencillas, no necesitan ser descodificadas en instrucciones menores como en el caso de los chips CISC, pues ya constituyen en sí unidades descodificadas. Por ello, el procesador RISC no gasta tiempo verificando el tamaño del comando, en descodificarlo ni en averiguar cómo cargarlo y guardarlo.

El procesador RISC puede además ejecutar hasta 10 comandos a la vez pues el compilador del software es el que determina qué comandos son independientes y por ello es posible ejecutar varios a la vez. Y al ser los comandos del RISC más sencillos, la circuitería por la que pasan también es más sencilla. Estos comandos pasan por menos transistores, de forma que se ejecutan con más rapidez. Para ejecutar una sola instrucción normalmente les basta con un ciclo de reloj.

Entre las ventajas de RISC tenemos las siguientes:

- La CPU trabaja más rápido al utilizar menos ciclos de reloj para ejecutar instrucciones.
- Utiliza un sistema de direcciones no destructivas en RAM. Eso significa que, a diferencia de CISC, RISC conserva después de realizar sus operaciones en memoria los dos operandos y su resultado, reduciendo la ejecución de nuevas operaciones.
- Estos microprocesadores siguen tomando como base el esquema moderno de Von Neumann.
- Las instrucciones, aunque con otras características, siguen divididas en tres grupos:
 - Transferencia.
 - Operaciones.
 - Control de flujo.
- Reducción del conjunto de instrucciones a instrucciones básicas simples, con la que pueden implantarse todas las operaciones complejas.
- Arquitectura del tipo load-store (carga y almacena). Las únicas instrucciones que tienen acceso a la memoria son 'load' y 'store'; registro a registro, con un menor número de acceso a memoria.
- Casi todas las instrucciones pueden ejecutarse dentro de un ciclo de reloj. Con un control implantado por hardware (con un diseño del tipo load-store), casi todas las instrucciones se pueden ejecutar cada ciclo de reloj, base importante para la reorganización de la ejecución de instrucciones por medio de un compilador.
- Pipeline (ejecución simultánea de varias instrucciones). Posibilidad de reducir el número de ciclos de máquina necesarios para la ejecución de la instrucción, ya que esta técnica permite que una instrucción puede empezar a ejecutarse antes de que haya terminado la anterior.

Ejemplo de microprocesadores basados en la tecnología RISC:

- MIPS, Millions Instruction Per Second.
- PA-RISC, Hewlett Packard.
- SPARC, Scalable Processor Architecture, Sun Microsystems.
- POWER PC, Apple, Motorola e IBM.

RISC V :

<https://www.youtube.com/watch?v=wg3j35gQ71k>

<https://www.youtube.com/watch?v=wg3j35gQ71k>

Set de instrucciones de un procesador MIPS

Ing. Carlos Eduardo Molina C.

www.redtauros.com

cemolina@redtauros.com

AMD vs INTEL

Característica	Arquitectura CISC (Intel x86)	Arquitectura RISC (AMD e Intel modernas)
Definición	Usa instrucciones complejas que realizan múltiples operaciones en una sola instrucción.	Usa instrucciones simples y reducidas que se ejecutan en un solo ciclo de reloj.
Historia en Intel	Intel ha sido tradicionalmente asociado con CISC (arquitectura x86).	Intel ha incorporado elementos RISC en sus procesadores modernos (ej: microoperaciones en x86).
Historia en AMD	AMD también se basa en la arquitectura x86 (CISC) para compatibilidad con software.	AMD ha adoptado enfoques RISC en sus diseños, como en la arquitectura Zen
Ventajas	<ul style="list-style-type: none"> - Compatibilidad con software antiguo. - Menor necesidad de compilar código frecuentemente. 	<ul style="list-style-type: none"> - Mayor eficiencia energética. - Mejor escalabilidad en multi-núcleos.
Desventajas	<ul style="list-style-type: none"> - Mayor consumo de energía. - Mayor complejidad en el diseño. 	<ul style="list-style-type: none"> - Requiere más instrucciones para tareas complejas. - Dependencia de compiladores optimizados.
Ejemplos en Intel	Procesadores Intel Core (basados en x86)	Uso de microoperaciones RISC en procesadores Intel Core modernos.
Ejemplos en AMD	Procesadores AMD Ryzen (compatibles con x86).	Arquitectura Zen, que utiliza un diseño basado en RISC para mayor eficiencia.
Enfoque moderno	Intel y AMD han fusionado conceptos de RISC y CISC para mejorar el rendimiento y la eficiencia.	Ambos fabricantes usan diseños híbridos, combinando lo mejor de ambas arquitecturas.
Uso en aplicaciones	Ideal para aplicaciones generales y compatibilidad con software antiguo.	Ideal para aplicaciones que requieren alto rendimiento y eficiencia energética.

Notas adicionales:

- **Intel:** Aunque Intel ha sido históricamente asociado con CISC (x86), sus procesadores modernos traducen instrucciones complejas (CISC) en microoperaciones más simples (RISC) para mejorar la eficiencia.
- **AMD:** AMD ha adoptado un enfoque similar en sus procesadores Ryzen, utilizando una arquitectura

basada en RISC (Zen) para ejecutar instrucciones x86 (CISC) de manera más eficiente.

- **Convergencia:** En la actualidad, la distinción entre RISC y CISC es menos clara, ya que ambos fabricantes utilizan técnicas híbridas para optimizar el rendimiento y la eficiencia energética.

Practica:

Programa en C++ que suma 5 numeros aleatorios:

```
#include <iostream>
#include <cstdlib> // Para usar rand() y srand()
#include <ctime>   // Para usar time()

int main() {
    // Semilla para generar números aleatorios
    std::srand(std::time(0));

    int suma = 0;

    std::cout << "Los numeros generados son: ";

    // Bucle for para generar y sumar 5 números aleatorios
    for (int i = 0; i < 5; ++i) {
        int numero = std::rand() % 100; // Genera un número aleatorio entre 0 y 99
        std::cout << numero << " ";
        suma += numero; // Suma el número generado
    }

    std::cout << "\nLa suma de los numeros es: " << suma << std::endl;

    return 0;
}
```

Utilizar:

<https://godbolt.org/>

Pegar el código y compilarlo para:

Ing. Carlos Eduardo Molina C.

www.redtauros.com

cemolina@redtauros.com

- x86 msvc v19.latest

Y en otra pestana para:

- mips clang 20.1.0

https://www.cs.tufts.edu/comp/140/lectures/Day_3/mips_summary.pdf

<https://learn.microsoft.com/es-es/windows-hardware/drivers/debugger/x86-instructions>

<https://bellard.org/jslinux/>

Conclusiones

- Hoy en día, los programas cada vez más grandes y complejos demandan mayor velocidad en el procesamiento de información, lo que implica la búsqueda de microprocesadores más rápidos y eficientes.
- Los avances y progresos en la tecnología de semiconductores han reducido las diferencias en las velocidades de procesamiento de los microprocesadores con las velocidades de las memorias, lo que ha repercutido en nuevas tecnologías en el desarrollo de microprocesadores. Hay quienes consideran que en breve los microprocesadores RISC sustituirán a los CISC, pero existe el hecho que los microprocesadores CISC tienen un mercado de software muy difundido.
- En la década de los sesenta, la microprogramación era la técnica más apropiada para la tecnología de memorias existentes. En consecuencia, los procesadores se dotaron de poderosos conjuntos de instrucciones, dando surgimiento a la arquitectura CISC.
- Las arquitecturas CISC utilizadas desde hace 15 años han permitido desarrollar un gran número de productos de software. Sin embargo, simultáneamente aumentan las aplicaciones en las cuales la capacidad de procesamiento que se pueda obtener del sistema es más importante que la compatibilidad con el hardware y el software anteriores. Por ello, todos los productores de estaciones de trabajo de renombre, han pasado en pocos años, de los procesadores CISC a los RISC, lo cual se refleja en el fuerte incremento anual del número de procesadores RISC.
- Cada usuario debe decidirse a favor o en contra de determinada arquitectura de procesador en función de la aplicación concreta que quiera realizar. Nunca será decisiva únicamente la capacidad de procesamiento del microprocesador; se debe considerar por igual la capacidad real que puede alcanzar el sistema en su conjunto.
- Si bien el campo de aplicaciones de la arquitectura RISC crece con fuerza, esto no equivale al fin de la arquitectura CISC, que también seguirá perfeccionándose adoptando técnicas típicas de los procesadores RISC, a fin de encontrar nuevas rutas para el incremento de sus capacidades.
- Cuando en 1990 Apple lanza su primer Power Macintosh y anuncia que se pasa al RISC, muchos auguraron que la tecnología de instrucciones complejas tenía los días contados. Pero desde entonces han pasado 17 años y los fabricantes de procesadores CISC han seguido aumentando el rendimiento, y lo que es más importante, han conseguido mantener los precios de los chips muy bajos.
- Intel ha mantenido hasta ahora que la arquitectura CISC puede estar perfectamente a la altura de la RISC. Pero la progresiva asimilación de técnicas propias del RISC en sus chips parece revelar un abandono progresivo de la arquitectura de instrucciones complejas por la de instrucciones más reducidas
- AMD e Intel siguen caminos distintos en el rendimiento de sus microprocesadores. Intel se decidió por aumentar los MHz para aumentar el rendimiento mientras que AMD se decantó por aumentar el número de instrucciones que se pueden realizar por ciclo de reloj.

https://triton.astroscu.unam.mx/fruiz/introduccion/introduccion_computacion/Arquitectura%20RISC%20vs%20CISC.pdf