

**UNIVERSIDAD DE PANAMÁ
CENTRO REGIONAL UNIVERSITARIO DE VERAGUAS
FACULTAD DE INFORMÁTICA, ELECTRÓNICA Y
COMUNICACIÓN**

MONOGRAFÍA:

RISC VS. CISC

PRESENTA:

RAÚL ENRIQUE DUTARI DUTARI

2009

TABLA DE CONTENIDOS

1.	Observaciones Preliminares.	1
2.	CISC Ante RISC.....	1
3.	RISC Y CISC: Diseños Opuestos.	4
3.1.	Caso De Estudio: Multiplicación De Números En Memoria.	4
3.1.1.	La filosofía CISC Para El Caso De Estudio.....	5
3.1.2.	La filosofía RISC Para El Caso De Estudio.....	6
4.	Arquitecturas CISC Y RISC: Origen Y Características Fundamentales.	7
4.1.	La Arquitectura Tradicional (CISC).	7
4.2.	La Arquitectura Evolutiva (RISC).	10
5.	El Rol Que Desempeñan Los Compiladores En Los Sistemas RISC Y CISC.....	15
6.	Capacidad Real De Procesamiento De Los Sistemas Desde El Punto De Vista Del Usuario.	16
7.	Campo De Aplicación De Los Procesadores CISC Y RISC.....	17
8.	Y El Ganador Es: RISC Ó CISC.....	19

9.	Conclusiones.....	20
10.	Referencias Bibliográficas.....	21

1. OBSERVACIONES PRELIMINARES.

En la actualidad, el software es cada vez más grande y complejo, demandando mayor capacidad de procesamiento de información del hardware, que se refleja en la necesidad de implementar microprocesadores más rápidos y eficientes.

Por otro lado, 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.

Estos factores se encuentran en franca controversia, al momento de decidir que arquitectura de procesadores se debe utilizar para una gama de aplicaciones particulares:

- La arquitectura RISC (Reduced Instruction Set Computer o conjunto reducido de instrucciones de cómputo).
- La arquitectura CISC (Complex Instruction Set Computer o conjunto reducido de instrucciones de cómputo).

Esta monografía pretende ofrecer al lector una panorámica de estas arquitecturas, presentando sus características fundamentales, así como sus ventajas y desventajas y sus campos de aplicación más comunes.

2. CISC ANTE RISC.

En principio, se debe analizar el significado de los conceptos CISC y RISC:

- CISC (Complex Instruction Set Computer) Computadoras con un conjunto de instrucciones complejo.
- RISC (Reduced Instruction Set Computer) Computadoras con un conjunto de instrucciones reducido.

Los atributos complejo y reducido describen las diferencias entre los dos modelos de arquitectura para microprocesadores solo de forma superficial. *Se requieren otras características esenciales para definir a los procesadores RISC y los CISC típicos.* Aun más, existen diversos procesadores de actualidad que no se pueden asignar con facilidad a ninguna categoría determinada.

Así, los términos complejo y reducido, expresan muy bien una importante característica definitiva, siempre que no se tomen solo como referencia las instrucciones, sino que se considere también la complejidad del hardware del procesador.

Con tecnologías de semiconductores comparables e igual frecuencia de reloj, un procesador RISC típico tiene una capacidad de procesamiento de dos a cuatro veces mayor que la de un CISC, pero su estructura de hardware es tan simple, que se puede implementar en una fracción de la superficie ocupada por el circuito integrado de un procesador CISC.

Esto hace suponer que RISC reemplazará al CISC, pero la respuesta a esta cuestión no es tan simple ya que:

1. Para aplicar una determinada arquitectura de microprocesador son decisivas las condiciones de realización técnica y sobre todo la rentabilidad, incluyendo los costos de software.

2. Existían y existen razones de compatibilidad para desarrollar y utilizar procesadores de estructura compleja así como un extenso conjunto de instrucciones.

La meta principal es incrementar el rendimiento del procesador, ya sea optimizando alguno existente o se desee crear uno nuevo. Para esto se deben considerar tres áreas principales a cubrir en el diseño del procesador, que son:

- La Arquitectura: Aunque la tecnología de proceso y de encapsulado son vitales en la elaboración de procesadores más rápidos, es la arquitectura del procesador lo que hace la diferencia entre el rendimiento de una CPU (Control Process Unit) y otra.
- La Tecnología De Proceso: Se refiere a los materiales y técnicas utilizadas en la fabricación del circuito integrado
- El Encapsulado: Se refiere a cómo se integra un procesador con lo que lo rodea en un sistema funcional, que de alguna manera determina la velocidad total del sistema.

Dependiendo de cómo el procesador almacena los operandos de las instrucciones de la CPU, existen tres tipos de juegos de instrucciones:

- Juego de instrucciones para arquitecturas basadas en pilas.
- Juego de instrucciones para arquitecturas basadas en acumulador.
- Juego de instrucciones para arquitecturas basadas en registros.

Las arquitecturas **RISC** y **CISC** son ejemplos de CPU con un conjunto de instrucciones para arquitecturas basadas en registros.

3. RISC Y CISC: DISEÑOS OPUESTOS.

La manera más sencilla de ver las ventajas y desventajas de las arquitecturas RISC, es contrastarlas con sus predecesoras: las arquitecturas CISC, vía un caso de estudio.

3.1. CASO DE ESTUDIO: MULTIPLICACIÓN DE NÚMEROS EN MEMORIA.

A continuación, se presenta un diagrama que representa el esquema de almacenamiento de un computador genérico.

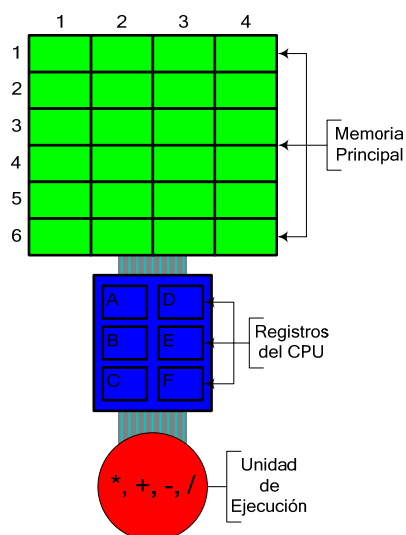


Ilustración 1: Esquema de almacenamiento en un computador genérico.

La memoria principal está dividida en emplazamientos numerados desde Fila 1, Columna 1 (1:1) hasta Fila 6, Columna 4 (6:4). La unidad de ejecución es

responsable de llevar a cabo todos los cálculos. Sin embargo, la unidad de ejecución sólo puede operar con datos que han sido cargados en alguno de los 6 registros de propósito general existentes (A, B, C, D, E, ó F). Digamos que queremos hallar el producto de 2 números: el primero está almacenado en la posición (2:3) y el otro está en la posición (5:2). Después guardaremos el producto hallado en la posición (2:3) de memoria.

3.1.1. LA FILOSOFÍA CISC PARA EL CASO DE ESTUDIO.

El objetivo principal de la arquitectura CISC es completar una tarea en el menor número de líneas de código ensamblador posibles. Este objetivo es conseguido mediante la construcción de un microprocesador capaz de comprender y ejecutar una serie de operaciones complejas. Para esta tarea en particular, un procesador CISC vendría preparado con una instrucción específica (la llamaremos "MULT"). Cuando se ejecuta, esta instrucción lee los dos valores de memoria, multiplica los operandos en la unidad de ejecución, y después almacena el resultado en la posición de memoria adecuada. De esta manera, la tarea completa de multiplicar dos números puede ser llevada a cabo con una única instrucción:

MULT (2:3), (5:2)

MULT es lo que se conoce como una "instrucción compleja". Opera directamente sobre los bancos de memoria del computador y no requiere al programador para llamar explícitamente a una carga de operandos o funciones de almacenamiento. Se parece mucho a un comando en un lenguaje de programación de alto nivel. Por ejemplo, si permitimos que "a" represente el valor de (2:3) y "b" represente el valor de (5:2), entonces este comando es idéntico a la expresión en C "a = a * b".

Una de las ventajas principales de esta filosofía es que el compilador tiene que hacer muy poco trabajo para traducir un lenguaje de alto nivel a ensamblador. Además, debido a que la longitud del código es relativamente corta, hace falta poca RAM para almacenar las instrucciones. Pero la dificultad está en construir instrucciones complejas directamente en hardware.

3.1.2. LA FILOSOFÍA RISC PARA EL CASO DE ESTUDIO.

Los procesadores RISC sólo usan instrucciones sencillas que se puedan ejecutar rápidamente. Por este motivo suelen ser arquitecturas basadas en registros de propósito general que operan siempre sobre operandos que se encuentran almacenados en el procesador, cerca de la unidad de ejecución. De esta forma, el comando "MULT" descrito más arriba podría ser dividido en tres comandos por separado: "LOAD", que mueve datos del banco de memoria a un registro; "PROD", que halla el producto de dos operandos situados en los registros; y "STORE", que mueve datos de un registro al banco de memoria. Para realizar la serie de pasos descritos en la filosofía CISC, un programador debería codificar la instrucción en 4 líneas de código ensamblador:

```
LOAD A, (2:3)
LOAD B, (5:2)
PROD A, B
STORE (2:3), A
```

A primera vista, puede parecer una manera mucho menos eficiente de completar la operación. Debido a que hay más líneas de código, hace falta más RAM para almacenar las instrucciones en ensamblador. El compilador debe realizar también más trabajo para convertir un lenguaje de alto nivel en código.

Sin embargo, la estrategia RISC también tiene grandes ventajas. Debido a que cada instrucción realiza una operación muy simple, el código se ejecutará en aproximadamente el mismo tiempo que el comando "MULT" de la arquitectura CISC. Estas "instrucciones reducidas" RISC requieren menos hardware y más sencillo que las instrucciones complejas, dejando más espacio para registros de propósito general. Además, las optimizaciones sobre un hardware más sencillo son mucho más fáciles de realizar.

4. ARQUITECTURAS CISC Y RISC: ORIGEN Y CARACTERÍSTICAS FUNDAMENTALES.

A continuación, se procede a exponer las características más importantes de las arquitecturas RISC y CISC.

4.1. LA ARQUITECTURA TRADICIONAL (CISC).

La tecnología CISC vio la luz pública de dentro de la empresa Intel, creador 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 números y letras nacería la cuarta generación de ordenadores, la de los conocidos como PC u ordenadores personales.

No obstante, el origen de la arquitectura CISC se remonta a los inicios de la programación ubicada en los años 60 y 70. Para contrarrestar la crisis del software de ese entonces, empresas electrónicas fabricantes de hardware pensaron que una buena solución era crear una CPU con un amplio y detallado manejo de instrucciones, a fin de que los programas fueran más sencillos. Los

programadores en consecuencia crearon multitud de programas para esa arquitectura. La posterior masificación de los PCs, permitió que el mercado fuera luego copado de software creado para procesadores CISC.

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 implementaciones de procesadores 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 compleja.

En la década de los sesentas (60) la microprogramación, por sus características, era la técnica más apropiada para las tecnologías de memorias existentes en esa época y permitía desarrollar también procesadores con compatibilidad ascendente. En consecuencia, los procesadores se dotaron de poderosos conjuntos de instrucciones.

Cuando el sistema operativo o una aplicación requieren 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 procesador 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. Ambos procesos ralentizan el rendimiento del sistema.

Posteriormente, el procesador envía el comando solicitado a una unidad que lo descodifica en instrucciones más pequeñas que podrán ser ejecutadas por un nano-procesador, 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, lo que bloquea la posibilidad de realizar procesamiento paralelo.

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

Ahora, todas las dificultades que pasa el procesador CISC al ejecutar el código, tienen su contrapartida en una serie de ventajas que ofrecen, entre otras:

- 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.

Como ejemplo de microprocesadores basados en la tecnología CISC, representativos, tenemos, entre otros:

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

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).

4.2. LA ARQUITECTURA EVOLUTIVA (RISC).

Un procesador **RISC** es el resultado de las investigaciones hechas por Paterson y Hennessy en Berkeley y Stanford respectivamente, a inicios y mediados de la década de 1980. Estas investigaciones fueron los primeros trabajos que se hicieron en forma sistemática y analítica sobre los procesadores y como se puede mejorar sustancialmente el rendimiento de los mismos.

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.

La idea estuvo inspirada también por el hecho de que muchas de las características que eran incluidas en los diseños tradicionales de CPU para aumentar la velocidad estaban siendo ignoradas por los programas que eran ejecutados en ellas. Además, la velocidad del procesador en relación con la memoria de la computadora que accedía era cada vez más alta.

Debido a que, bajo el nuevo paradigma, 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.

En consecuencia, 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 compiladores 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 decodificadas en instrucciones menores como en el caso de los chips CISC, pues ya constituyen en sí unidades decodificadas. Por ello, el procesador RISC no gasta tiempo verificando el tamaño del comando, en decodificarlo 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.

En el diseño de un computador RISC se deben respetar cinco pasos que se enuncian a continuación:

1. Analizar las aplicaciones para encontrar las operaciones clave.
2. Diseñar un bus de datos que sea óptimo para las operaciones clave.
3. Diseñar instrucciones que realicen las operaciones clave utilizando el bus de datos.

4. Agregar nuevas instrucciones sólo si no hacen más lenta a la máquina.
5. Repetir este proceso para otros recursos.

El primer punto se refiere a que el diseñador deberá encontrar qué es lo que hacen en realidad los programas que se pretenden ejecutar. Ya sea que los programas a ejecutar sean del tipo algorítmicos tradicionales, o estén dirigidos a robótica o al diseño asistido por computadora.

La parte medular de cualquier sistema es la que contiene los registros, el ALU y los buses que los conectan. Se debe optimar este circuito para el lenguaje o aplicación en cuestión. El tiempo requerido, (denominado tiempo del ciclo del bus de datos) para extraer los operandos de sus registros, mover los datos a través del ALU y almacenar el resultado de nuevo en un registro, deberá ser mínimo.

Esto quiere decir que para lograr un óptimo rendimiento deben crearse rutinas de programación eficientes, dejando atrás las complejas y largas rutinas que logran entorpecer el óptimo desempeño de los procesadores. De esta manera tendremos mayor cantidad de operaciones tanto aritméticas como lógicas desarrolladas de manera precisa en menos tiempo logrando un mayor beneficio tanto para el equipo como para el usuario.

El siguiente punto a cubrir es diseñar instrucciones de máquina que hagan un buen uso del bus de datos. Por lo general se necesitan solo unas cuantas instrucciones y modos de direccionamiento; sólo se deben colocar instrucciones adicionales si serán usadas con frecuencia y no reducen el desempeño de las más importantes, buscando de esta manera mejorar el rendimiento del bus.

Siempre que aparezca una nueva y atractiva característica, deberá analizarse y ver la forma en que se afecta al ciclo de bus. Si se incrementa el tiempo del ciclo, probablemente no vale la pena tenerla.

Por último, el proceso anterior debe repetirse para otros recursos dentro del sistema, tales como memoria cache, administración de memoria, coprocesadores de punto flotante, etcétera.

En consecuencia, entre las principales ventajas de la tecnología RISC se pueden mencionar, entre otras:

- 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.
- Cada instrucción puede ser ejecutada en un solo ciclo del CPU

Entre muchos ejemplos de microprocesadores basados en la tecnología RISC, se pueden mencionar a:

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

5. EL ROL QUE DESEMPEÑAN LOS COMPILADORES EN LOS SISTEMAS RISC Y CISC.

El compilador juega un papel clave para un sistema RISC equilibrado.

Todas las operaciones complejas se trasladan al microprocesador para ser gestionadas por medio de conexiones fijas en el circuito integrado para agilizar las instrucciones básicas más importantes. De esta manera, el compilador asume la función de un mediador inteligente entre el programa de aplicación y el microprocesador.

Es decir, se hace un gran esfuerzo para mantener al hardware tan simple como sea posible, aún a costa de hacer al compilador considerablemente más complicado. Esta estrategia se encuentra en clara contra posición con las máquinas CISC que tienen modos de direccionamiento muy complicados. En la práctica, la existencia en algunos modos de direccionamiento complicados en los microprocesadores CISC, hacen que tanto el compilador como la micro-código sean muy complicados.

No obstante, las máquinas CISC no tienen características complicadas como carga, almacenamiento y salto que consumen mucho tiempo, las cuales en efecto aumentan la complejidad del compilador.

Para suministrar datos al microprocesador de tal forma que siempre esté trabajando en forma eficiente, se aplican diferentes técnicas de optimización en distintos niveles jerárquicos del software.

Los diseñadores de RISC en la empresa MIP y en Hewlett Packard trabajan según la siguiente regla:

Una instrucción ingresa en forma fija en el circuito integrado del procesador (es decir, se alambra físicamente en el procesador) si se ha demostrado que la capacidad total del sistema se incrementa en por lo menos un 1%.

En cambio, los procesadores CISC, han sido desarrollados por equipos especializados de las empresas productoras de semiconductores y con frecuencia el desarrollo de compiladores se sigue por separado. Por consiguiente, los diseñadores de los compiladores se encuentran con una interfaz hacia el procesador ya definido y no pueden influir sobre la distribución óptima de las funciones entre el procesador y compilador.

Las empresas de software que desarrollan compiladores y programas de aplicación, tienden por razones de rentabilidad, a utilizar diferentes procesadores como usuarios de su software en lugar de realizar una optimización completa, y aprovechar así las respectivas características de cada uno. Esta situación genera otros factores negativos de eficiencia. Esta limitación de las posibilidades de optimización del sistema, que viene dada a menudo por una obligada compatibilidad, se superó con los modernos desarrollos RISC.

6. CAPACIDAD REAL DE PROCESAMIENTO DE LOS SISTEMAS DESDE EL PUNTO DE VISTA DEL USUARIO.

Aparte de la base conceptual para el desarrollo de un sistema de computación de alta calidad, se requieren técnicas especiales para optimizar cada uno de los factores que determinan la capacidad de procesamiento, que sólo pueden definirse con el programa de aplicación.

La información suministrada por un fabricante, sobre la velocidad en MIPS (Millones de Instrucciones por Segundo) que una arquitectura es capaz de realizar, carece de relevancia hasta que el usuario sepa cuantas instrucciones genera el respectivo compilador, al traducir su programa de aplicación y cuánto tiempo tarda la ejecución de estas instrucciones.

Solo el análisis de diferentes pruebas y comparaciones de rendimiento (benchmarks, en inglés) ofrecerán una idea aproximada, que el usuario puede aplicar para delimitar las arquitecturas adecuadas.

7. CAMPO DE APLICACIÓN DE LOS PROCESADORES CISC Y RISC.

Las arquitecturas CISC utilizadas desde hace 15 años han permitido desarrollar un gran número de productos de software. Ello representa una considerable inversión y asegura a estas familias de procesadores un mercado creciente.

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, lo cual no solo es válido en los subsistemas de alta capacidad en el campo de los llamados sistemas empotrados o "embedded", en los que siempre dominaron las soluciones especiales de alta capacidad de procesamiento sino también para las estaciones de trabajo "workstations".

Esta clase de equipos se han introducido poco a poco en oficinas, en la medicina y en bancos, debido a los cada vez más voluminosos y complejos paquetes de software que con sus crecientes requerimientos de reproducción

visual, que antes se encontraban solo en el campo técnico de la investigación y desarrollo.

En este tipo de equipos, el software de aplicación, se ejecuta bajo el sistema operativo UNIX, el cual es escrito en lenguaje C, por lo que las arquitecturas RISC actuales están adaptadas y optimizadas para este lenguaje de alto nivel. 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, (los procesadores RISC de 32 bits han visto crecer su mercado hasta en un 150% anual). En pocos años, el RISC conquistará de 25 al 30% del mercado de los 32 bits, pese al aparentemente abrumador volumen de software basado en procesadores con el estándar CISC que se ha comercializado en todo el mundo.

La arquitectura MIPS-RISC ha encontrado, en el sector de estaciones de trabajo, la mayor aceptación. Los procesadores MIPS son fabricados y comercializados por cinco empresas productoras de semiconductores, entre las que figuran NEC y Siemens. Los procesadores de los cinco proveedores son compatibles en cuanto a las terminales, las funciones y los bits.

Por otro lado, las empresas fabricantes de procesadores CISC, con Intel a la cabeza, paulatinamente han incluido conceptos de la tecnología RISC en sus diseños de procesadores, con la meta de mantener su competitividad en el mercado.

8. Y EL GANADOR ES: RISC Ó CISC.

Con base en la exposición previa, se deben evaluar las ventajas de ambas arquitecturas para tomar decisiones sobre la escogencia de una u otra a la hora de diseñar un sistema.

RISC es una tecnología más rápida, pero de mayor costo. Hablando en términos de costo hay que pensar que RISC utiliza más la circuiteria (comandos hardware o circuitos electrónicos) para ejecutar operaciones directas (el microprocesador tiene menor carga de trabajo), en tanto que CISC utiliza micro-código ejecutado por el microprocesador lo que la hace mas económica y mas lenta también (debido a la carga adicional de instrucciones que debe procesar el microprocesador).

Por un lado, hay más software de uso general para la plataforma CISC, que para RISC. Pero la exigencia de la informática demanda periódicamente mayor velocidad y administración de espacio en RAM y discos duros, área en la que ambas arquitecturas deben seguir innovando. Dado que CISC es mas popular a nivel de PC, las innovaciones en esta categoría son mas numerosas (nuevas interfaces, puertos, nuevos buses y velocidades de transmisión).

Técnicamente hablando, el rendimiento en RISC basado en la menor cantidad de carga de instrucciones en el microprocesador compensa a la mayor cantidad de código en software que es necesario utilizar, por lo que su arquitectura se considera más potente que CISC, siempre que se pueda costear.

9. CONCLUSIONES.

- Existen grandes diferencias entre los procesadores RISC y CISC, en la actualidad el mercado es ampliamente dominado por los procesadores CISC, debido en gran parte al desconocimiento de las personas de la funcionalidad y grandes ventajas que se pueden obtener de los procesadores RISC y en otro aspecto al costo de estos (RISC) que se hacen más difíciles de obtener por usuarios comunes.
- Los procesadores RISC manejan el concepto de que entre más práctico y preciso sea el código mayor provecho se le podrá sacar al procesador ya que no se ocupara espacio de memoria ni tiempo innecesario en tareas que no son importantes no como lo desarrollan los procesadores CISC los cuales manejan instrucciones más complejas para realizar las mismas tareas.
- El futuro de los procesadores RISC como procesadores comerciales masivos está distante, mientras que el rendimiento que se puede obtener de estos es más alto del que estamos obteniendo en la actualidad, con los procesadores CISC.
- Los procesadores CISC están incluyendo paulatinamente más y más conceptos de la tecnología RISC, para mantener su competitividad en el mercado informático.

10. REFERENCIAS BIBLIOGRÁFICAS.

- [HEPA93] HENNESSY, John; PATTERSON, David. Arquitectura de Computadoras. Un enfoque Cuantitativo. Primera Edición, McGraw-Hill, España, 1993.
- [STAL06] STALLINGS, William. Organización y Arquitectura de Computadoras. Principios de Estructura y de Funcionamiento. Séptima Edición, Pearson Prentice-Hall, España, 2006.
- [TANE92] TANENBAUM, Andrew S. Organización De Computadoras: Un Enfoque Estructurado. Tercera Edición, Prentice-Hall, México, 1992.
- [WIKIA09] FUNDACIÓN WIKIMEDIA. Complex Instruction Set Computing. Wikimedia Foundation, Inc. Fecha de Actualización: 2009-mayo-28. Fecha de Consulta: 2009-junio-10. Disponible en: <http://es.wikipedia.org/wiki/CISC>.
- [WIKIB09] FUNDACIÓN WIKIMEDIA. RISC. Wikimedia Foundation, Inc. Fecha de Actualización: 2009-junio-02. Fecha de Consulta: 2009-junio-10. Disponible en: <http://es.wikipedia.org/wiki/RISC>.