



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

MATERIA:

Sistemas Operativos

ALUMNO:

**Francisco Javier García Pilar
108002**

TAREA:

**CONCEPTOS DE SOS'S
"PROCESADORES CISC VS RISC"**

FEBRERO 25 DEL 2008



PROCESADORES CISC VS RISC

RESUMEN

Desde hace algunos años, el diseño de los microprocesadores han tenido cambios, y se distinguen dos filosofías: CISC y RISC. Por Complex Instruction Set Computer se entiende un

procesador tradicional, dotado de complejas operaciones que, sin embargo, tienden a ser poco utilizadas porque son especializados. La tendencia opuesta Reduced Instruction Set Computer, especifica diseños donde se reduce la potencia y complejidad de las instrucciones, pero aumentando su velocidad de proceso como compensación. No se puede decir que un diseño RISC sea inherentemente mejor que uno CISC porque, siendo estos aparatos tan enormemente elaborados y complejos, cualquier análisis simplista fracasará, como suele suceder. Además, los nuevos microprocesadores incluyen características de ambas filosofías en su diseño.

PRIMERA CLASIFICACIÓN DEL CPU EN BASE AL MANEJO DE INSTRUCCIONES.

Los Microprocesadores o CPU administran juegos de instrucciones basadas en pilas, acumuladores y registros. Las instrucciones basadas en registros han recibido la mayor atención por parte de los programadores, hecho que a su vez ha propiciado que los fabricantes de semiconductores, diseñen arquitecturas de microprocesadores SEGUN la forma en que se administran los registros.

Partiendo de esa base, han surgido dos grandes arquitecturas de microprocesadores para PCs: los diseñados con instrucciones avanzadas o complejas llamados CISC (Complex Instruction Set Computer) y los diseñados con instrucciones simples o reducidas llamados RISC (Reduced Instruction Set Computer).

LA ARQUITECTURA CISC (Complex Instruction Set Computer).

Fue la primera tecnología de CPUs con la que la maquina PC se dio a conocer mundialmente. Adoptada por Intel, se colocó en las primitivas PCs (procesador 8088) que fueron lanzadas bajo la marca IBM el 12 de Agosto de 1981. Su sistema de trabajo se basa en la Microprogramación. Dicha técnica consiste en hacer que cada instrucción sea interpretada por un microprograma localizado en una sección de memoria en el circuito integrado del Microprocesador. A su vez, las instrucciones compuestas se decodifican para ser ejecutadas por microinstrucciones almacenadas en una ROM interna. Las operaciones se realizan al ritmo de los ciclos de un reloj.

Considerando la extraordinaria cantidad de instrucciones que la CPU puede manejar, la construcción de una CPU con arquitectura CISC es realmente compleja. A este grupo pertenecen los microprocesadores de INTEL (celeron, Pentium II, Pentium III, Pentium IV) y AMD (Duron, Athlon).

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 mas 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.

Entre las bondades de CISC destacan las siguientes:

1. Reduce la dificultad de crear compiladores.
2. Permite reducir el costo total del sistema.
3. Reduce los costos de creación de Software.
4. Mejora la compactación de código.
5. Facilita la depuración de errores (debugging).

LA ARQUITECTURA RISC (RISC = Reduced Instruction Set Computer).

Ha sido la consecuencia evolutiva de las CPU. Como su nombre lo indica, se trata de microprocesadores con un conjunto de instrucciones muy reducidas en contraposición a CISC. ¿Que ventaja se deriva de esta tecnología? Veamos: 1. La CPU trabaja mas rápido al utilizar menos ciclos de reloj para cumplir sus funciones (ejecutar instrucciones). 2. 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 operando y su resultado (total tres direcciones), lo que facilita a los compiladores conservar llenos los 'pipelines' (conductos) de la CPU para utilizarlos concurrentemente y reducir la ejecución de nuevas operaciones. 3. Cada instrucción puede ser ejecutada en un solo ciclo de la CPU (máxima velocidad y eficiencia).

Considerada como una innovación tecnológica creada a partir del análisis de la primitiva arquitectura CISC, RISC ha dado origen a la aparición de Microprocesadores poderosos cuya principal aplicación a la fecha (Octubre 2003), ha sido el trabajo en las grandes máquinas (servidores), aunque también han llegado a posicionarse en ciertas maquinas desktop (Apple), computadoras de mano, maquinas de juegos, y otros artefactos electrónicos domésticos.

Ejemplos de tecnología RISC son los sistemas MIPS (Millions Instruction Per Second), 1992, SPARC = Scalable Processor ARCHitecture de la empresa Sun (utiliza Solaris, sistema operativo de ambiente Unix), POWER PC, 1993, diseñado por Apple, Motorola e IBM, son utilizados en PCs de Apple, Macintosh y mainframes de IBM (RS/6000 y AS/400) con sistemas operativos AIX y Windows NT. El PowerPC se conoce también como G3, G4, G5 (alcanza un billon de operaciones de punto flotante por segundo).

LAS ARQUITECTURAS ENFRENTADAS: RISC VS CISC.

Partiendo de lo expuesto, habría que 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 más rápida, pero más costosa. Hablando en términos de costo hay que pensar que RISC utiliza mas la circuiteria (comandos hardware o circuitos electrónicos) para ejecutar operaciones directas (el microprocesador esta mas libre de carga), 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 que soporta el microprocesador). -

Hay más software de uso general para la plataforma CISC. 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 más popular a nivel de PCs, las innovaciones en esta categoría son más 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.

FUNDAMENTOS Y ORIGEN DEL RISC

Los titulares que definen la arquitectura RISC, podría resumirse, con la suficiente flexibilidad, en varios puntos:

- Reducción del número de instrucciones (ensamblador).
- Uso intensivo de registros, disminuyendo los accesos a memoria.
- Simplificación de la CPU en aras de una mayor velocidad de proceso.
- Empleo de memorias caché.
- Utilización de "compiladores optimizados", generadores de código objeto adaptado a los requerimientos de la CPU.

CORAZÓN DEL RISC

La CPU o procesador es el centro, el cerebro del ordenador. Será ella quien dicte las acciones a tomar tras leer la información contenida en memoria. Se alimenta de instrucciones en lenguaje ensamblador para, a partir de cada una de estas, generar una pleyade de ordenes (microordenes) que, semiocultas al programador, recorreran las interioridades del propio procesador, activando registros, etc.,... de su complejo hardware, a la vez que escapan hacia el exterior de la CPU con la misión de gobernar el comportamiento de la memoria buses de E/S, etc. En resumen son estas microordenes las que constituyen el sistema nervioso del ordenador. El problema se plantea al implementar en la CPU un dispositivo que "dispare" una serie de respuestas ante el estímulo de los "LOAD", "STORE", "JUMP", "MOVE"...

Las CPU's cuentan con una pequeña memoria de control de naturaleza ROM en la que se graba el conjunto de microordenes que corresponden a cada instrucción. Cuando una de estas es introducida en el procesador, lo que se hace realmente es utilizarla para direccionar adecuadamente la memoria de control y obtener así las microordenes correspondientes. ¿Ventajas? Un cambio en las instrucciones solo exige regrabar esta ROM.

¿COMO SURGE EL RISC?

Los ordenadores etiquetados como CISC gozan de los privilegios y defectos del micro código. La CPU es considerablemente más rápida que la memoria principal. Esto significa que conviene manejar un amplio abanico de instrucciones complejas cuyo significado equivalga al de varias instrucciones simples, disminuyendo así los accesos a memoria. A esto se le añade la tangencia de aumentar el grado de complejidad de las instrucciones para acercarlas a los lenguajes de alto nivel.

Sin embargo, como resultado de ciertos estudios en los que se examinó la frecuencia de utilización de las diferentes instrucciones, se observó que el 80 % del tiempo era consumido por solo el 20 % de las instrucciones, con prioridad de los almacenamientos (STORE), cargas (LOAD) y bifurcaciones (BRANCH).

Instrucciones más usadas

Esto significa que se poseían soberbias memorias de control cuyo contenido era muy poco utilizado. Se estaba penalizando la velocidad de respuesta en aras de tener información poco útil.

La alternativa RISC se basa en esto y propone procesadores cableados con un repertorio simple de instrucciones sencillas y frecuentes; todo código complejo puede descomponerse en varios congénitos más elementales en los que, para evitar los terribles efectos sobre los retardos de la memoria principal (MP), se recurre a numerosos registros y a memorias caché. Un registro es una unidad de almacenamiento enclavada en la CPU y, por tanto, tan rápida como esta. Las memorias caché son pequeñas memorias de alta velocidad, se alimentan de la MP, de la que toman la información que está siendo más frecuentemente utilizada. Otro de los objetivos del RISC fue lograr que a cada instrucción correspondiera un solo ciclo de reloj, a excepción, de aquellos casos que hay que mover datos.

- Disminuye la probabilidad de fallo.
- Reduce el tamaño de la CPU, que puede entonces albergar más recursos (registros).
- Facilita el diseño.
- Permite máquinas más compactas y con menor consumo.
- A menor complejidad... menor coste.

Aun así con todos los datos obtenidos a favor y en contra del CISC y del RISC, hay que tener en cuenta otro factor importante: la eficacia del software.

COMPILADORES OPTIMIZADOS DEL RISC

Es cierto que un procesador RISC es más veloz que uno CISC, pero también lo es que, al ser más simples las instrucciones, necesita más de estas para emular funciones complejas, por lo que los programas son más largos y voluminosos. Es decir, el código objeto generado, ocupa más memoria y, al ser más extenso, emplea más tiempo

En ser procesado. Los partidarios argumentan que el factor volumen de memoria incide poco en el precio, además estiman que el aumento de código no toma dimensiones importantes por el uso de coprocesadores y compiladores optimizados.

Los segundos destacan dos aspectos:

1.- Al existir menor variedad en el código generado, el proceso de compilación es más rápido. El motivo es que hay menor número de reglas y posibilidades entre las que elegir (no existirá la disyuntiva de construir la misma acción por diferentes caminos, solo habrá una forma única) evitando la exploración de grandes tablas de instrucciones en busca del sujeto correcto.

“En un ordenador convencional, la misma instrucción de lenguaje de alto nivel puede ejecutarse de diversa formas, cada una con sus inconvenientes y ventajas, pero en el RISC solo hay una forma de hacer las cosas ”

2.- Al traducir los lenguajes de alto nivel mediante unidades de extrema simplicidad, se incrementa la eficiencia. Si se emplean instrucciones potentes se corre el riesgo de no aprovecharlas en su totalidad y potencia, menor es la adaptación a los diferentes casos particulares

¿RISC O CISC?

El conflicto surge al evaluar las ventajas netas ¿que es más apropiado, usar muchas instrucciones de un solo ciclo aprovechadas al máximo, o pocas de múltiples pasos de reloj en las que existe infratilización?

La cuestión, es que hasta el momento, el estudio de prestaciones de ambas tecnologías, nos ha llevado a concluir que hoy en día los RISC obtienen más prestaciones, es decir, son más potentes y rápidos que los CISC. Sin embargo, el mercado se ha decantado por la tecnología CISC en cuanto a volumen de ventas. ¿Por que?

1.- Por experiencia propia, podemos comprobar que un CISC tiene un coste "razonable", que es alcanzado a nivel de usuario. Esto mismo, no ocurre con los RISC, que por el contrario tienen un coste elevado, por esto mismo esta tecnología ha sido enfocada a ventas a nivel de empresa y equipos de gama alta.

2.- La utilidad que se le da a la maquina es muy importante, ya que el usuario debe encontrar un nivel optimo en cuanto a calidad - precio. Y por que pagar más si realmente no se le va a sacar partido al cien por cien.

3.- El software utilizado es otro de los factores importantes, dado que un RISC no utiliza el mismo software que un CISC. Estos últimos, por lo general tienen un software más asequible.

4.- Dada la compatibilidad hacia atrás de la familia CISC x86, los usuarios han podido renovar sus equipos sin por ello tener que abandonar software que ya conocían, y reutilizar sus datos. Así mismo, los fabricantes han tenido en cuenta este factor, puesto que seguir con otra línea de procesadores suponía no solo un cambio muy radical, sino que además podía llevar un riesgo en cuanto a ventas.

Estos son algunos de los motivos. Sin embargo, también hay que tener en cuenta el conflicto de intereses de algunos fabricantes, así como la opinión de distintas revistas, algunas de ellas asociadas a diferentes marcas.

Se están estudiando las tendencias futuras, como pueden ser los híbridos, mejoras en los microprocesadores CISC, mejoras en los RISC,...