

Arquitectura Risc y Cisc

Claudio Acuña, Guillermo Rojas, Cristian Garrido, José Acuña

08/10/13

1 Introducción

Una de las primeras decisiones a la hora de diseñar un microprocesador es decidir cual será su juego de instrucciones. La decisión por dos razones; primero, el juego de instrucciones decide el diseño físico del conjunto; segundo, 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.

2 Arquitectura Risc

En la arquitectura computacional, RISC (del inglés reduced instruction set computer) es un tipo de microprocesador con las siguientes características fundamentales:

- Instrucciones de tamaño fijo y presentadas en un reducido número de formatos.
- Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

El objetivo de diseñar máquinas con esta arquitectura es posibilitar la segmentación y el paralelismo en la ejecución de instrucciones y reducir los accesos a memoria.

Las máquinas RISC protagonizan la tendencia actual de construcción de microprocesadores. PowerPC, DEC Alpha, MIPS, ARM, SPARC... son ejemplos de algunos de ellos. RISC es una filosofía de diseño de CPU para computadora que está a favor de conjuntos de instrucciones pequeñas y simples que toman menor tiempo para ejecutarse.

El tipo de procesador más comúnmente utilizado en equipos de escritorio, el x86, está basado en CISC en lugar de RISC, aunque las versiones más nuevas traducen instrucciones basadas en CISC x86 a instrucciones más simples basadas

en RISC para uso interno antes de su ejecución.

La idea fue inspirada 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. Esto con llevó la aparición de numerosas técnicas para reducir el procesamiento dentro del CPU, así como de reducir el número total de accesos a memoria.

2.1 Características:

En pocas palabras esto significa que para cualquier nivel de desempeño dado, un chip RISC típicamente tendrá menos transistores dedicados a la lógica principal. Esto permite a los diseñadores una flexibilidad considerable; así pueden, por ejemplo:

- Incrementar el tamaño del conjunto de registros.
- Mayor velocidad en la ejecución de instrucciones.
- Implementar medidas para aumentar el paralelismo interno.
- Añadir cachés enormes. Añadir otras funcionalidades, como E/S y relojes para minicontroladores.
- Construir los chips en líneas de producción antiguas que de otra manera no serían utilizables.
- No ampliar las funcionalidades, y por lo tanto ofrecer el chip para aplicaciones de bajo consumo de energía o de tamaño limitado.

2.1.1 Las características que generalmente son encontradas en los diseños RISC son:

- Codificación uniforme de instrucciones, lo que permite una de codificación más rápida.
- Un conjunto de registros homogéneo, permitiendo que cualquier registro sea utilizado en cualquier contexto y así simplificar el diseño del compilador.
- Modos de direccionamiento simple con modos más complejos reemplazados por secuencias de instrucciones aritméticas simples.
- Los tipos de datos soportados en el hardware no se encuentran en una máquina RISC.
- Los diseños RISC también prefieren utilizar como característica un modelo de memoria Harvard, donde los conjuntos de instrucciones y los conjuntos de datos están conceptualmente separados.

3 Arquitectura Cisc

En la arquitectura computacional, CISC (complex instruction set computer) es un modelo de arquitectura de computadora.

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, en contraposición a la arquitectura RISC.

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 del tipo RISC, llamadas generalmente microinstrucciones. Los CISC pertenecen a la primera corriente de construcción de procesadores, antes del desarrollo de los RISC. Ejemplos de ellos son: Motorola 68000, Zilog Z80 y toda la familia Intel x86 usada en la mayoría de las computadoras personales actuales.

Para realizar una sola instrucción un chip CISC requiere de cuatro a diez ciclos de reloj.

3.1 Ejemplo de microprocesadores basados en la tecnología CISC:

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

3.2 Ejemplo de microprocesadores basados en la tecnología CISC:

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

4 Diferencias Entre Cisc y Risc

4.1 Cisc

- 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.

4.2 Risc

- La CPU trabaja mas 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.

4.3 Cuadro Comparativo

Figure 1: Cisc v/s Risc

CISC	RISC
Instrucciones multiciclo	Instrucciones de único ciclo
Carga/almacenamiento incorporadas en otras instrucciones	Carga/almacenamiento son instrucciones separadas
Arquitectura memoria-memoria	Arquitectura registro-registro
Instrucciones largas, Código con menos líneas	Instrucciones cortas, Código con más líneas
Utiliza memoria de microprograma	Implementa las instrucciones directamente en hardware
Se enfatiza la versatilidad del repertorio de instrucciones	Se añaden instrucciones nuevas sólo si son de uso frecuente y no reducen el rendimiento de las más importantes
Reduce la dificultad de implementar compiladores	Compiladores complejos
	Elimina microcódigo y la decodificación de instrucciones complejas