RISC

(computador con conjuntos de instrucciones reducidas)

Son procesadores que están diseñados para ejecutar un número reducido de tipos de instrucciones que les permite operar a una velocidad más elevada. La arquitectura RISC principalmente requiere menos cantidad de hardware y una mayor flexibilidad de construcción.

¿Qué son las instrucciones?

Las instrucciones no son más que una serie de indicaciones de lo que debe hacer el procesador con los datos, las cuales permiten desarrollar los programas. Debemos además entender que el término «reducido» no implica que la cantidad de instrucciones usadas sean limitadas, sino que son más sencillas.

La arquitectura RISC lo que permite es que las unidades aritmético-lógicas y las de punto flotante sean más sencillas. Permite crear chips más compactos o aprovechar el espacio sobrante para integrar otros elementos. Adicionalmente estos procesadores suelen requerir menos tensiones de funcionamiento y generan menos calor.

**Origen**

El inicio del desarrollo de la arquitectura RISC empezó en la década de 1970. Las primeras soluciones desarrolladas basadas en esta arquitectura fueron desarrolladas por IBM y las universidades de Stanford y Berkeley.

Quien parece ser que acuñó el término RISC fue David Patterson, Profesor de la Universidad de California en Berkeley. El concepto se utilizó en los procesadores SPARC de Sun Microsystems y fue la base de la creación de MIPS Technologies, actualmente parte de Silicon Graphics.

**Características de RISC**

La arquitectura RISC se basa en cinco características de diseño:

* Ejecución en un único ciclo: La mayoría de los diseños de CPU convencionales tienen una tasa máxima de ejecución de una instrucción por ciclo de máquina básica. Para una determinada tecnología, el tiempo de ciclo puede tener un límite inferior. En las CPU complejas, la mayoría de las instrucciones generadas por el compilador suelen ser sencillas. El diseño RISC lo que hace es enfatizar solo un ciclo único de ejecución. Permite incluso sintetizar secuencias de múltiples instrucciones para operaciones menos frecuentes.
* Poco o ningún microcódigo: Los microcódigos lo que hacen es agregar capas de sobrecarga operativas. Esto lo que hace es aumentar el número de ciclos por instrucción. Algo que puede llevar a que instrucciones sencillas puedan requerir varios ciclos de cómputo.
* Pocos modos de direccionamiento: Se simplifican las instrucciones complejas y los modos de direccionamiento. Esto implica a las instrucciones del microcódigo o del multiciclo.
* Diseño de registro-registro: Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos. Estos realizan operaciones de registro-registro. Esto se deriva de los tres principios anteriores.
* Pipelining profundo y eficiente: Para hacer un uso conveniente de la paralelización del hardware sin complejo microcódigo horizontal, los procesadores utilizan canalizaciones (pipelining). Una canalización de ‘n’ etapas mantienen ‘z’ instrucciones activas de manera simultánea, terminando una en cada ciclo, idealmente. Las instrucciones deben de ajustarse cuidadosamente para permitir el pipelining.

**Ventajas**

* Ofrecen un mejor rendimiento gracias al número simple y limitado de instrucciones que soportan, La CPU trabaja mas rápido al utilizar menos ciclos de reloj para ejecutar instrucciones.
* Requiere de menor cantidad de hardware físico, lo cual reduce su coste
* Permite que el espacio libre en el encapsulado se puede usar para integrar otros circuitos
* Menos consumo energético y generación de calor, haciéndolos ideales para dispositivos móviles
* Son muy polivalentes, pudiéndose usar para todo tipo de aplicaciones

**Desventajas**

* Podemos encontrar rendimiento variable según el software ejecutado. Es posible que las instrucciones nuevas dependan de una instrucción anterior.
* Usan más instrucciones y registros para realizar las mismas tareas que los procesadores CISC y es posible que no admitan algunas instrucciones o modos que sí lo hacen los procesadores CISC.
* Estos procesadores necesitan de más memoria y muy rápida para guardar varias instrucciones, lo cual hace necesaria mucha memoria caché para poder ejecutar las instrucciones en muy poco tiempo

**ARM qué es?**

Lo primero que tenemos que saber de ARM es que es una empresa y al mismo tiempo es una arquitectura de procesadores o SoCs. ARM Holdings es la empresa que desarrolla los diseños de la arquitectura ARM de chips. Y es que la particularidad de ARM Holdings es que no fabrica chips o procesadores, sino licencias de arquitecturas (diseños).

**Usos de la arquitectura RISC**

Sistemas móviles y de gama baja

* La arquitectura ARM domina el mercado de los sistemas integrados de bajo consumo y bajo coste (normalmente de 200 a 1800 MHz en 2014). Se utiliza en varios sistemas, como la mayoría de los sistemas basados ​​en Android, el iPhone y iPad de Apple, Microsoft Windows Phone (antes Windows Mobile ), dispositivos RIM, Nintendo Game Boy Advance, DS, 3DS y Switch, Raspberry Pi, etc.
* El PowerPC de IBM se utilizó en las consolas de juegos GameCube, Wii, PlayStation 3, Xbox 360 y Wii U.
* La línea MIPS (en un punto utilizada en muchas computadoras SGI ) se usó en PlayStation, PlayStation 2, Nintendo 64, consolas de juegos PlayStation Portable y gateways residenciales como la serie Linksys WRT54G .

**Ordenadores de sobremesa y portátiles**

* Algunos Chromebook utilizan plataformas basadas en ARM desde 2012.
* Apple usa procesadores Apple Silicon de diseño propio basados ​​en la arquitectura ARM para su línea de computadoras de escritorio y portátiles desde su transición de los procesadores Intel, y las primeras computadoras de este tipo se lanzaron en noviembre de 2020.

**Estaciones de trabajo, servidores y supercomputadoras**

* MIPS, de Silicon Graphics (dejó de fabricar sistemas basados ​​en MIPS en 2006).
* SPARC , de Oracle (anteriormente Sun Microsystems ) y Fujitsu.
* IBM ‘s IBM POWER arquitectura del conjunto de instrucciones, PowerPC, y la ISA de energía fueron y son utilizados en muchos de los superordenadores de IBM, servidores de gama media y estaciones de trabajo.

**Diferencias con CISC**

* **Complejidad de instrucciones**:
* **RISC**: Se caracteriza por tener un conjunto de instrucciones reducido y simples, lo que facilita su ejecución en un solo ciclo de reloj.
* **CISC**: Presenta un conjunto de instrucciones más amplio y complejo, que incluye operaciones que requieren múltiples ciclos de reloj para ejecutarse.
* **Uso de microcódigo**:
* **RISC**: Minimiza o elimina el uso de microcódigo, lo que ayuda a reducir la sobrecarga operativa y optimiza la velocidad de ejecución.
* **CISC**: Suelen utilizar microcódigo para traducir instrucciones complejas en secuencias de operaciones simples, lo que puede ralentizar la ejecución.
* **Modos de direccionamiento:**
* **RISC**: Utiliza pocos modos de direccionamiento, lo que simplifica las instrucciones y agiliza su ejecución.
* **CISC**: Ofrece una variedad de modos de direccionamiento complejos para manejar diversas operaciones, lo que puede complicar el diseño y la ejecución eficiente.
* **Diseño de registro-registro**:
* **RISC**: Prioriza las operaciones entre registros de la CPU, minimizando el acceso a la memoria.
* **CISC**: Incluye instrucciones que pueden operar directamente con la memoria, aumentando la flexibilidad pero potencialmente ralentizando la ejecución.
* **Pipelining**:
* **RISC**: Suele tener un pipelining más profundo y eficiente, permitiendo la ejecución simultánea de múltiples instrucciones.
* **CISC**: El pipelining puede ser menos eficiente debido a la complejidad de las instrucciones y la necesidad de traducción a microinstrucciones.

**Conclusión**

En resumen, la arquitectura RISC, con su enfoque en instrucciones simples y eficiencia de hardware, mientras que CISC busca ofrecer instrucciones complejas y versátiles, ofrece ventajas significativas en términos de rendimiento, consumo de energía y versatilidad. Su evolución hacia RISC-V como una arquitectura de conjunto de instrucciones abierta promete democratizar aún más el acceso a procesadores eficientes y personalizables.