

Procesadores x86 vs ARM

Oscar Andrés Rosas Hernández *Instituto Politécnico Nacional, Escuela Superior de Cómputo, CDMX*

I. IDEAS DESPUES DE LEER EL ARTICULO

Si se puede resumir estas diferencias en un par de palabras. ARM es una arquitectura RISC (Computación de conjuntos de instrucciones reducidas) mientras que x86 es una arquitectura CISC (Computación de conjuntos de instrucciones complejas).

I-A. 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 del tipo RISC, llamadas generalmente microinstrucciones.

El nombre CISC apareció por contraposición a RISC cuando apareció esta nueva arquitectura de diseño (finales 1980).

I-B. RISC

Es una filosofía de diseño de CPU para computadora que está a favor de conjuntos de instrucciones pequeños y simples que se ejecutan más rápidamente y acceden a registros internos. Esta filosofía proviene del hecho de que muchas de las características de los diseños de las CPU estaban siendo ignoradas por los programas que eran ejecutados en ellas, principalmente por una simplificación de los compiladores que tienden a utilizar siempre las mismas instrucciones.

Además, la velocidad del procesador en relación con la memoria de la computadora que accedía era cada vez más alta, por lo que se decidió reducir los accesos a memoria. Esto llevó a la aparición de RISC que utiliza diversas técnicas para reducir el procesamiento dentro del CPU, así como de reducir el número total de accesos a memoria.

Entonces regresando a ARM vs x86.

La diferencia principal es que las instrucciones ARM operan solo en registros con algunas instrucciones para cargar y guardar datos desde / a la memoria, mientras que x86 también puede operar directamente en la memoria.

Hasta v8 ARM era una arquitectura nativa de 32 bits, que favorecía las operaciones de cuatro bytes sobre otras.

Por lo tanto, ARM es una arquitectura más simple, que conduce a un tamaño pequeño y muchas funciones de ahorro de energía, mientras que x86 se convierte en una bestia en términos de consumo de energía y producción.

El ARM es como un auto deportivo italiano:

- Un muy buen equilibrado, bien ajustado motor. Da buena aceleración y velocidad máxima.
- Excelentes chasis, frenos y suspensión. Puede detenerse rápidamente, puede girar sin disminuir la velocidad.

El x86 es como un muscle estadounidense:

- Gran motor, gran bomba de combustible. Proporciona excelente velocidad máxima y aceleración, pero usa mucho combustible.
- Frenos terribles, hay que poner una cita en su calendario, si deseamos reducir la velocidad.
- Terrible capacidad para girar, tienes que reducir la velocidad.

En resumen: el x86 se basa en un diseño de 1974 y es bueno en línea recta (pero consume mucho combustible). ARM usa poco combustible, no se ralentiza en los giros (branches).

Metáfora terminada, aquí hay algunas diferencias reales.

- ARM tiene más registros.
- ARM tiene pocos registros de propósito especial, en x86 todo es un registro de propósito especial
- ARM tiene pocos comandos de acceso a la memoria, solo carga / almacena el registro.
- ARM es simple y rápido.

- Las instrucciones de ARM son arquitectónicamente de ciclo único (excepto cargar / almacenar múltiples).
- Las instrucciones de ARM a menudo hacen más de una cosa (en un solo ciclo).
- Cuando se necesita más de una instrucción de ARM, como el almacenamiento en bucle del x86 y el incremento automático, ARM lo hace en menos ciclos de reloj.
- ARM tiene más instrucciones condicionales.
- El predictor de ramificación es trivialmente simple (si es incondicional o al revés, entonces asume ramificación, de lo contrario asume no ramificación), y funciona mejor que el muy muy complejo en el x86 en muchas ocasiones.
- ARM tiene un conjunto de instrucciones simple y consistente (puede compilar a mano y aprender el conjunto de instrucciones rápidamente).

REFERENCIAS

- [1] <https://stackoverflow.com/questions/14794460/how-does-the-arm-architecture-differ-from-x86>
- [2] A Detailed Analysis of Contemporary ARM and x86 Architectures
https://www.researchgate.net/publication/266457125_A_Detailed_Analysis_of_Contemporary_ARM_and_x86_Architectures