Procesadores superescalares

* ¿Qué características definen un procesador como superescalar? Describa las políticas de emisión de instrucciones en un cauce segmentado.

# ¿Qué es un procesador superescalar?

Un procesador superescalar es aquel que usa múltiples cauces de instrucciones independientes. Cada cauce consta de múltiples etapas, de modo que puede tratar varias instrucciones a la vez. El hecho de que haya varios cauces introduce un nuevo nivel de paralelismo, permitiendo que varios flujos de instrucciones se procesen simultáneamente. Un procesador superescalar saca provecho de lo que se conoce como “paralelismo a nivel de instrucciones” que hace referencia al grado en que las instrucciones de un programa pueden ejecutarse en paralelo.

Un procesador superescalar capta varias instrucciones a la vez y a continuación, intenta encontrar instrucciones cercanas que sean independientes entre sí y puedan, por consiguiente, ejecutarse en paralelo. Si la entrada de una instrucción depende de la salida de una instrucción precedente, la segunda instrucción no puede completar su ejecución al mismo tiempo ni antes que la primera, por lo que al identificar estas dependencias, el procesador puede emitir y completar instrucciones en un orden diferente al del código máquina original.

El procesador puede eliminar algunas dependencias innecesarias mediante el uso de registros adicionales y el renombramiento de las referencias a registros en el código original. Para maximizar la utilización del cauce de instrucciones y aumentar el rendimiento no usa saltos retardados (como los RISC) sino que hacen predicción de saltos.

Las instrucciones comunes (aritmética entera y de coma flotante, cargas, almacenamientos y bifurcaciones condicionales) pueden iniciar su ejecución simultáneamente y ejecutarse de manera independiente (en diferentes cauces).

El enfoque superescalar conlleva a la duplicación de algunas o todas las partes de la CPU/ALU:

* Captar múltiples instrucciones al mismo tiempo
* Ejecutar sumas y multiplicaciones simultáneamente
* Ejecutar carga/almacenamiento, mientras se lleva a cabo una operación en ALU

El grado de paralelismo y la aceleración de la máquina aumentan, ya que se ejecutan más instrucciones en paralelo.

# Políticas de emisión de instrucciones:

La emisión de instrucciones es el proceso de iniciar la ejecución de instrucciones en las unidades funcionales del procesador. Y la política de emisión de instrucciones es el protocolo usado para emitir instrucciones.

El procesador intenta localizar instrucciones más allá del punto de ejecución en curso, que puedan introducirse en el cauce y ejecutarse. Son importantes 3 ordenaciones:

* Orden en que se captan las instrucciones
* Orden en que se ejecutan las instrucciones
* Orden en que las instrucciones actualizan los contenidos de los registros y de las posiciones de memoria

Para optimizar la utilización de los diversos elementos del cauce, el procesador tendrá que alterar uno o más de estos órdenes con respecto al orden que se encontraría en una ejecución secuencial estricta. La única restricción del procesador es que el resultado sea correcto.

Las políticas pueden ser de tres categorías:

* Emisión en orden y finalización en orden

La más sencilla. Emitir instrucciones en el orden exacto en que lo haría una ejecución secuencial y escribir los resultados en ese mismo orden.

* Emisión en orden y finalización desordenada

La finalización desordenada se usan en los procesadores RISC escalares para mejorar la velocidad de las instrucciones que necesitan muchos ciclos. Con la finalización desordenada puede haber cualquier número de instrucciones en la etapa de ejecución en un momento dado, hasta alcanzar el máximo grado de paralelismo de la máquina, ocupando todas las unidades funcionales.

La emisión de instrucciones se para cuando hay una pugna por un recurso, una dependencia de datos o una dependencia relativa al procedimiento.

Aparece la dependencia de salida o dependencia escritura-escritura.

La finalización desordenada necesita una lógica de emisión de instrucciones más compleja que la finalización en orden. Además, es más difícil ocuparse de las interrupciones y excepciones.

* Emisión desordenada y finalización desordenada

Con la emisión en orden, el procesador solo decodificara instrucciones hasta que haya conflicto o dependencia, no puede ir más allá del punto del conflicto buscando instrucciones que podrían ser independientes (hasta que el mismo no se resuelva) y que podrían introducirse provechosamente en el cauce

Para permitir la emisión desordenada, es necesario desacoplar las etapas del cauce de decodificación y ejecución, mediante un buffer llamado ventana de instrucciones. Cuando un procesador termina de decodificar una instrucción, coloca ésta en la ventana de instrucciones. Mientras el buffer no se llene, puede continuar captando y decodificando nuevas instrucciones. Cuando una unidad funcional de la etapa de ejecución queda disponible, se puede emitir una instrucción desde la ventana de instrucciones a la etapa de ejecución. Cualquier instrucción puede emitirse, siempre que: necesite la unidad funcional particular que esté disponible y que ningún conflicto ni dependencia la bloqueen.

Así el procesador tiene capacidad de anticipación, permitiendo identificar las instrucciones independientes que pueden introducirse en la etapa de ejecución. Las instrucciones se emiten desde la ventana de instrucciones sin que se tenga en cuenta su orden original en el programa. Aparece una nueva dependencia: la antidependencia o dependencia lectura-escritura, donde en lugar de que la 1ra instrucción produzca un valor que usa la 2da, la 2da instrucción DESTRUYE un valor que usa la 1ra

**Fuente**:

* “Clase 08”
* “Capítulo 13 - Procesadores superescalares” (Stalling 5ta ed. Pág 480 y 487)