Perfil7
www.wuolah.com/student/Perfil7

ApuntesTEMA4AC.pdf

APUNTES con diapositivas y libro

2° Arquitectura de Computadores

Grado en Ingeniería Informática

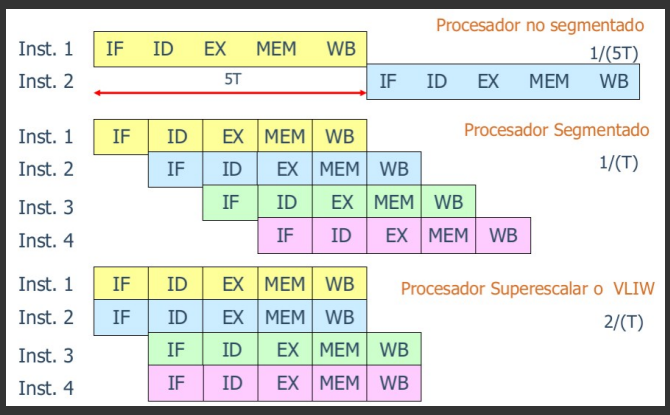
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

- ARQUITECTURA DE COMPUTADORES -

TEMA 4

Lección 11

PARALELISMO ENTRE INSTRUCCIONES



PROCESADORES SUPERESCALARES Y VLIW

Ambos disponen de múltiples unidades de ejecución, las cuales usan para ejecutar varias instrucciones de forma simultánea y emitir múltiples instrucciones en paralelo.

Sin embargo, en los procesadores superescalares es el propio hardware el que debe extraer el paralelismo de las instrucciones que ejecute, mientras que, los VLIW usan el paralelismo explícito en la palabra de ejecución, ya les viene en la propia palabra las instrucciones en paralelo.

Por lo tanto, podemos extraer otra diferencia, y es que el hardware de los superescalares es más complejo, dado que necesita unidades que extraigan paralelismo. La microarquitectura VLIW es más simple en sobre todo en etapas de emisión y finalización: buffers de renombrado, buffers de reordenamiento, etc.

MEJORA DE LAS PRESTACIONES DE LOS PROCESADORES

Sabiendo que la tecnología de la fabricación de circuitos integrados basada en el Silicio ha mejorado debido a la reducción del tamaño de los transistores y el aumento del tamaño del dado.

Y partiendo de que la V_{CPU} = IPC * F.

Al tener más transistores por circuito integrado, podemos implementar microarquitecturas más complejas como procesadores superescalares que posibilitan el paralelismo entre instrucciones. Lo que significa que : ⬆IPC.

Y por otro lado, se reduce la longitud de las puertas de los transistores, lo que se traduce en mayores frecuencias: ⬆F.

⬆IPC y ⬆F. → ⬆ V_{CPU}

PARALELISMO ENTRE INSTRUCCIONES (ILP) Y PARALELISMO DE LA MÁQUINA

El paralelismo entre instrucciones se debe a las dependencias de datos y de control de la operación, además del retardo de la misma.

El paralelismo de la máquina, sin embargo, está determinado por el número de instrucciones que pueden captarse y ejecutarse al mismo tiempo y la velocidad de la ejecución. En definitiva, el hardware de la máquina.

Podemos diferenciar 3 tipos de orden en secuencias de instrucciones:

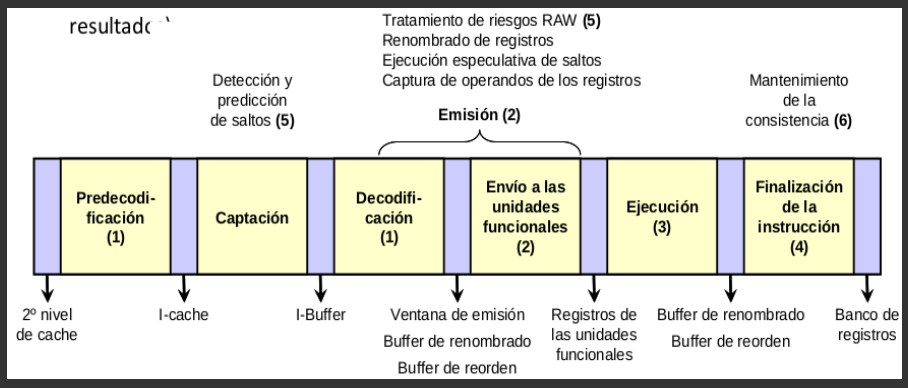
- El orden en el que se captan las instrucciones (orden en el código)
- El orden en que se ejecutan
- El orden en que modifican los registros

Un procesador superescalar debe encargarse él mismo de encontrar el paralelismo entre instrucciones para organizando la fase de captación, decodificación y ejecución en base a esto para aprovechar al máximo los recursos de la máquina. De manera que cuanto más sofisticado es el procesador superescalar, menos se ajusta a la ordenación de las instrucciones según se captan.

PROCESAMIENTO SUPERESCALAR Y ETAPAS DEL CAUCE

Compuesto por las fases:

1. Decodificación paralela → Aunque existe una decodificación más rápida anterior, la Precodificación.
2. Emisión paralela → Las instrucciones se emiten a las **unidades funcionales** según sus **dependencias**.
3. Ejecución paralela → Ejecutadas en las **unidades funcionales** (segmentadas).
4. Finalización del procesamiento de la instrucción
5. Detección y predicción de saltos
6. Mantenimiento de la consistencia secuencial (desacoplar la ejecución y escritura de resultados)



Un procesador segmentado decodifica una sola instrucción por ciclo, mientras que un superescalar lo hace con varias y además comprueba dependencias.

Gracias a la **precodificación**, podemos predecir que la instrucción será de salto o no, el tipo de unidad funcional que utilizará y si va a necesitar hacer referencia a memoria. Todo ello reflejado en unos bits de precodificación.

VENTANA DE INSTRUCCIONES

La ventana de instrucciones almacena las instrucciones pendientes, en caso de que sea una ventana centralizada, almacena todas las instrucciones, si es distribuida, solo almacena las instrucciones de un tipo determinado.

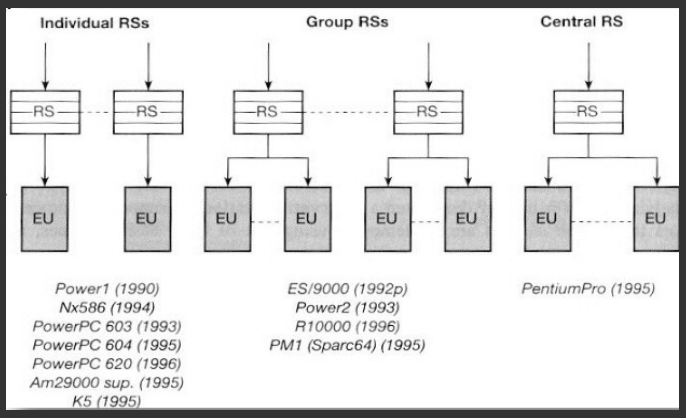
Dichas instrucciones solo se cargan en la ventana de instrucciones una vez decodificadas. Y para indicar si el operando está disponible, hay un bit específico, además se almacena el valor o la dirección donde se encuentra. Si el operando no está disponible, se almacena la unidad funcional de donde procederá el dato.

De manera que las instrucciones se emitirán solo cuando todos sus operandos estén disponibles al igual que la unidad funcional donde se debe procesar.

Estaciones de reserva de ventanas de instrucciones (Ventanas de instrucciones distribuidas) :

El decodificador tiene acceso a la cola de instrucciones, para poder emitirlas a la estación de reserva, desde donde serán enviadas a la Unidad funcional, junto con los números de registros fuente (al banco de registros) para servir como operandos.

Siempre que no hay limitaciones de hardware, las instrucciones se emite a las estaciones de reserva (buffers de shelving) independientemente de las dependencias. Y una vez están en la estación de reserva, esperan su turno limitadas por las dependencias, para acabar enviándose a las unidades funcionales cuando estén disponibles.



Existen distintas variables que afectan al envío de instrucciones a las unidades funcionales:

- Reglas de selección → Son las reglas que determinan qué instrucciones se envían a las UF
  Son las instrucciones ejecutables
- Reglas de arbitraje → Son las reglas que determinan qué instrucciones se envían cuando hay varias ejecutables.
  Son las más antiguas
- Orden de envío → Pueden darse envíos ordenados, desordenados y parcialmente ordenados.
- Velocidad de envío → Es el número de instrucciones que se pueden enviar por ciclo.
  Una sola instrucción o varias

Para poder llevar a cabo el envío, se debe comprobar la disponibilidad de los operandos mediante los bits de validez de la ventana o estación de reserva.

RENOMBRAMIENTO DE REGISTROS

Para evitar el efecto del las dependencias WAR (**emisión desordenada**) y WAW (**ejecucución desordenada**), implementamos el uso de registros físicos distintos.

La implementación puede darse durante la compilación (**estática**) o durante la ejecución (**dinámica**) con circuitos adicionales y registros extra.

Los buffers de renombrado pueden estar mezclados con los registros de la arquitectura o separados. Hay un número no estandarizado de buffers de renombrado y el mecanismo de acceso puede ser asociativo o indexado. La velocidad del renombrado se limita por el máximo número de nombres asignados por ciclo.

BUFFERS DE RENOMBRAMIENTO

Los buffers de renombramiento cumplen con este cometido y permiten varias escrituras pendientes a un mismo registro, utilizando el bit de último para marcar cual ha sido la más reciente.

ALGORITMO DE TOMASULO

Es un algoritmo de planificación dinámico para el renombramiento de registros.

Lección 12

CONSISTENCIA

En el procesamiento de una instrucción, se puede distinguir entre:

- Final de la ejecución de la operación codificada en las instrucciones
  - Cuando ya se obtienen los resultados generados por las UF pero no se han modificado los registros.
- El final del procesamiento de la instrucción (Complete o Commit)
  - Se escriben los resultados en los registros. Si se utilizan los buffers de reorden ROB, se utiliza el termino Retirar la instrucción, Retire, en lugar de completar.

Con consistencia de un programa, nos referimos al orden en que las instrucciones se completan y el orden en que se accede a memoria para leer (LOAD) y escribir (STORE).

Cuando nos referimos a consistencia débil del procesador, nos referimos a que las instrucciones se pueden ejecutar desordenadamente y por tanto deben detectarse y resolverse las dependencias de forma manual. Mientras que la consistencia es fuerte si hay un orden estricto que se implementa mediante uso de ROB.

La consistencia de memoria es débil cuando los accesos a memoria se pueden realizar de forma desordenada siempre que no afecten a las dependencias y también se deben detectar y solucionar de forma específica. La consistencia de memoria es fuerte cuando los accesos a memoria deben realizarse estríctamente en un orden implementado mediante ROB.

La consistencia fuerte se suele dar en procesadores y la débil en memorias.