



Tema 7. Introducción a la ejecución multiciclo

Organización de Computadores

LUIS ENRIQUE MORENO LORENTE
RAÚL PÉRULA MARTÍNEZ
ALBERTO BRUNETE GONZALEZ
DOMINGO MIGUEL GUINEA GARCIA ALEGRE
CESAR AUGUSTO ARISMENDI GUTIERREZ
JOSÉ CARLOS CASTILLO MONTOYA

Departamento de Ingeniería de Sistemas y Automática





MÁS COMPLICACIONES: OPERACIONES MULTICICLO

- Existen operaciones demasiado largas para ser acomodadas en un cauce segmentado de longitud fija:
 - La división entera.
 - La mayor parte de las operaciones en coma flotante.
 - Modos de direccionamiento complejos en procesadores CISC.
- El número de ciclos de reloj varía, puede depender de los datos que se ejecutan, esto afecta a:
 - Latencia
 - Ciclo de repetición
 - En una unidad segmentada completamente es de 1 ciclo.
- **No es posible utilizar un cauce estático** como el estudiado hasta este momento, donde todas las instrucciones pasan por todas las etapas del cauce.
 - Deben utilizarse **cauces dinámicos** con etapas que opcionalmente se ejecutan o no.





SOLUCIONES: UNA PRIMERA IDEA

- Indudablemente, una posibilidad sería exigir que todas las instrucciones completen su fase de ejecución en un ciclo.
 - Esto origina que el ciclo de reloj sea innecesariamente largo, y ralentizaría enormemente el procesador.
 - Otra posibilidad sería utilizar una enorme cantidad de HW para conseguir que el tiempo de la operación más larga se acorte todo lo posible, esto resulta muy caro.
- La única posibilidad práctica consiste en permitir que la etapa EX finalice la ejecución de la operación en los ciclos que necesite.
 - Esto genera, como consecuencia un aumento considerable de los riesgos estructurales, ya que mientras la etapa EX esta ejecutando no podría ser utilizada por una instrucción posterior.





SOLUCIONES POSIBLES

- Interbloqueo (detener el cauce)
 - Está solución implementa la idea anterior. Es la forma más sencilla de resolver el problema del riesgo estructural que plantean estas operaciones multiciclo.
 - Es altamente ineficiente, genera un gran número de burbujas en el cauce y degrada las prestaciones del mismo.
- Segmentación de la etapa de ejecución.
 - Si se segmenta la etapa de ejecución, se elimina el riesgo estructural (al menos en una parte) ya que la etapa de ejecución sería capaz de aceptar una ejecución de instrucción por ciclo.
 - Esta idea no es nueva, la instrucción load en la arquitectura MIPS esta segmentada siendo ejecutada en dos etapas (en la EX se calcula la dirección del operando en memoria y en la MEM se lee de la memoria) en vez de en una única etapa.
- Múltiples unidades funcionales.
 - Una posibilidad es utilizar múltiples unidades funcionales, de forma que mientras que se ejecuta una instrucción en una se pueda proceder a ejecutar otras instrucciones en otras unidades funcionales, de esta forma se solapan las ejecuciones de estas instrucciones.
 - Estas unidades funcionales pueden estar a su vez segmentadas.





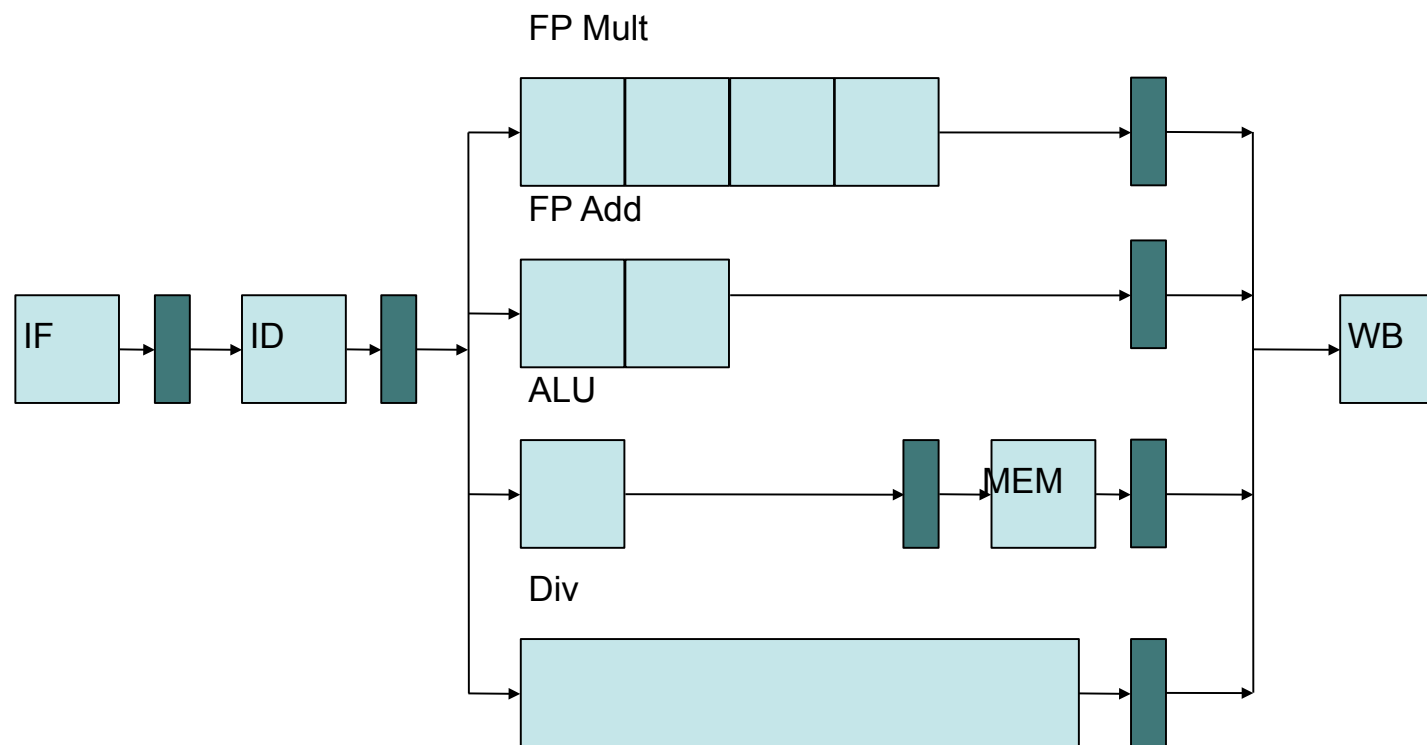
SOLUCIÓN MÁS EFICAZ

- La solución **más eficaz**, pero también la **más compleja** consiste en la utilización de múltiples unidades funcionales.
 - Esta solución implica que las instrucciones no se completan en el orden del programa original.
 - La finalización fuera de orden puede dar lugar a riesgos WAW cuando existen dependencias de salida entre instrucciones.
 - Existen dos posibilidades para resolver este riesgo WAW:
 - Una primera posibilidad consiste en retrasar el write back de la operación que se ejecuta antes (en el tiempo) para que se cumpla el orden de programa.
 - Otra posibilidad es desarrollar algún tipo de hardware que permita la finalización fuera de orden.
 - Entre las técnicas que permiten ejecutar y finalizar out-of-order, y que son denominadas planificación dinámica de instrucciones, tenemos:
 - » Técnica de Scoreboard.
 - » Algoritmo de Tomasulo.
 - » Ambas requieren hardware adicional de cierta complejidad.





ESQUEMA SIMPLE





PLANIFICACIÓN DINÁMICA DE INSTRUCCIONES: SCOREBOARD

- Esta técnica permite la ejecución fuera de orden cuando hay suficientes recursos y no existen dependencias de datos.
- Trata de mantener una tasa de ejecución de 1 instrucción por ciclo en ausencia de riesgos estructurales (permite la ejecución de las instrucciones tan pronto como es posible).
- El scoreboard, o marcador es un dispositivo HW que permite seguir el estado de la ejecución de las instrucciones, de las unidades funcionales que las ejecutan y de los registros que almacenarán el resultado.
- Utiliza ciertas estructuras de datos y se comunica con las unidades funcionales.
- Realiza la detección de riesgos, la resolución de estos y controla la progresión de las instrucciones de una etapa a otra.





PLANIFICACIÓN DINÁMICA DE INSTRUCCIONES: TOMASULO

- Esta técnica permite la ejecución fuera de orden cuando hay suficientes recursos y no existen dependencias de datos (verdaderas).
- Trata de mantener una tasa de ejecución de 1 instrucción por ciclo en ausencia de riesgos estructurales (permite la ejecución de las instrucciones tan pronto como es posible).
- Elimina los riesgos de datos producidos por dependencias de nombre, es decir elimina los riesgos WAR y WAW.
- Para ello asocia a cada UF una o más estaciones de reserva (registros internos).
- Utiliza un bus para la difusión rápida de los resultados (bus común de datos) y técnicas de sondeo de bus para anticipar los datos.
- El control se realiza por HW de forma distribuida, y sigue el estado de la ejecución de las instrucciones, de las unidades funcionales que las ejecutan y de los registros que almacenarán el resultado.
- Utiliza ciertas estructuras de datos y se comunica con las unidades funcionales.
- Realiza la detección de riesgos RAW, la resolución de estos y controla la progresión de las instrucciones de una etapa a otra.

