Superescalares

Scheduling estático

Introducción

- La CPU ejecuta las instrucciones en orden.
- El compilador:
  - Puede cambiar el orden de ejecución.
  - Genera el paquete de emisión.
  - Trata de prevenir o reducir los peligros de datos y de control.
- Paquete de emisión (issue packet): el conjunto de instrucciones que se emiten juntas en un ciclo de reloj.

Introducción

- La mayoría de CPUs móviles son superescalares con scheduling estático.

Ejemplo

- Considerar una CPU MIPS superescalar con scheduling estático y doble emisión.
- Limitación: solo una instrucción puede ser de tipo R o brinco y la otra puede ser una carga o un store.
- El paquete de emisión tiene 8 bytes (2 instrucciones).
- La instrucción R o de brinco se pone primero y luego la instrucción de memoria.

Ejemplo

- Si el paquete no se puede completar, la instrucción faltante se reemplaza con una instrucción nop (no operation).

Instruction type	Pipe stages							
ALU or branch instruction	IF	ID	EX	MEM	WB			
Load or store instruction	IF	ID	EX	MEM	WB			
ALU or branch instruction		IF	ID	EX	MEM	WB		
Load or store instruction		IF	ID	EX	MEM	WB		
ALU or branch instruction			IF	ID	EX	MEM	WB	
Load or store instruction			IF	ID	EX	MEM	WB	
ALU or branch instruction				IF	ID	EX	MEM	WB
Load or store instruction				IF	ID	EX	MEM	WB

Fuente: COD 5, p. 335.

Ejemplo

N	C	CPI = C / N
2	5	2.5
4	6	1.5
6	7	1.17
8	8	1.0
10	9	0.9
…	…	
n	(n / 2) + 4	((n / 2) + 4) / n

- $\text{CPI} = \lim_{n \to \infty} \frac{\frac{n}{2}+4}{n} = 0.5$

Peligros

- La forma de tratar los peligros depende de cada procesador.
- En algunos procesadores:
  - El compilador quita todos los peligros insertando nops si es necesario.
  - No hay detección de peligros por hardware ni detenciones (stalls).

Peligros

- Por ejemplo, en una CPU con doble emisión el compilador convierte el siguiente código:

```
add $t0, $t1, $t2          # paquete i
sub $t3, $t0, $t4          # paquete i
```

- en:

```
add $t0, $t1, $t2          # paquete i
nop                        # paquete i
sub $t3, $t0, $t4          # paquete i + 1
```

Peligros

- En otros procesadores:
  - El hardware detecta peligros de datos verdaderos y genera detenciones entre dos paquetes de emisión.
  - El compilador elimina las dependencias entre las instrucciones del mismo paquete de emisión.

Conclusiones

- Scheduling estático:
- Las instrucciones corren en el orden dictado por el compilador.
- Si los peligros no se detectan por hardware, el compilador debe hacerlo.
- Es más fácil de implementar que el scheduling dinámico.
- Se usa principalmente en CPUs móviles.