Arquitectura de Computadores

SEGMENTACIÓN DE CAUCE
PLANIFICACIÓN DINÁMICA

Reading Hennessy-Patterson:

1996 1ª Ed cap.6.7: Advanced Pipelining:Dynamic Scheduling

2017 6ª Ed cap.3.4: Overcoming Data Hazards With Dynamic Scheduling

Método

- Repasar definiciones de Riesgos de datos proporcionando ejemplos con el ISA del DLX.
  - RAW
  - WAW
  - WAR
- Casar con los conceptos de dependencias verdaderas, anti-dependencias, de salida y dependencias de nombre
- Leer la parte introductoria de la sección del HP indicada y responder a las preguntas siguientes
  - Todas las ediciones de HP tratan planificación dinámica pero evolucionan y abandonan marcadores (scoreboard centrándose en Tomasulo
- El propósito de las preguntas es ayudar a una lectura comprensiva en profundidad

PLANIFICACIÓN DINÁMICA

- Define la P.D con tus propias palabras
- Explica contrastando con la P. estática,
- ¿Cuál es mejor y porqué si cambia la configuración del cauce de un procesador (P.E. o P.D.)?
- ¿Cuál es el inconveniente respecto a la utilización de las U.F. de la emisión+ejecución en orden?
- ¿Por qué los riesgos WAR no ocurren sin pl. dinámica ? -proporciona ejemplo ilustrativo
- Explica cómo la pl. dinámica puede lograr un encauzamiento sin paradas
- ¿Por qué deben existir varias UF o UF segmentadas para lograr esta aproximación a CPI=1?
- ¿Se necesita algún repositorio o cola de fecth para varias instrucciones? ¿Por qué?

Pl. din.:la idea (Scoreboard)

- Fase ID partida: ¿qué hace cada parte? qué parte trata R. estructurales y cuál R.datos ahora
- ¿Hay emisión en orden o fuera de orden?
- ¿Hay ejecución en orden o fuera de orden?
- El problema con Scoreboard es su uso de los registros:
  - No hay circuito de adelantamiento: en caso de RAW la instrucción consumidora ha de esperar a leer el valor del banco de registros
- El alg. de TOmasulo usa la técnica de renombrado de registros la cual elimina dependencias de nombre (WAR y WAW). Es efectiva a lo largo de iteraciones de bucles tb.
  - nos centraremos en este método

Tomasulo

- https://www.youtube.com/watch?v=y-N0Dsc9LmU
- Video recomendado acerca del algoritmo de Tomasulo
- También contiene una buena introducción sobre planificación dinámica (hasta min. 2:40)
- Em este vídeo se va un pasa más allá con la planificación dinámica en bucles https://www.youtube.com/watch?v=eI-xrVbUIdY
- Sobre desenrrollado de bucles: https://www.youtube.com/watch?v=zXg5gvlxJkI
-

Tomasulo

- En qué fase se resuelven riesgos
  - estructurales
  - WAW
  - WAR
  - RAW
- ¿Qué se usa para reemplazar registros para lo cuales no hay un valor aún disponible? (renombrado)
- ¿Cuántas Estaciones de Reserva hay en una unidad de procesamiento en CF que use Tomasulo?
  - Qué contienen las ER exactamente? ¿Pueden tener más de una entrada?
- ¿El control es centralizado o descentralizado?¿Por qué?
- ¿Qué es el CDB y para qué se usa?

Completa una descripción de los elementos de una ER

- OP:
- Para opernados j y k
  - Vj, Vk :
  - Qj, Qk:
  - Son ambos válidos simultáneamente?
- UF status:

- Additionalmente:
  - ¿dónde se guarda la identidad del registro destino?
  - ¿Cómo se casan resultado y registro al que corresponde?

RAW & WAR & WAW

- Sigue el ejemplo del libro (atención al “bug” del código)
- Ejemplo del libro HP :
  - ld f6, 32(r2)
  - ld **f2**, 44(r2)
  - multd f0, f2, f4
  - subd **f8**, f2, f6
  - divd f10, f0, f6 (*)
  - **addd f6, f2, f8**

  - (*) esta instrucción especifica incorrectamente f0← f0/f6 en algunas ediciones de llibro
  - Desarrolla el ejemplo observando cómo se resuelven los riesgos

	UF							
ld f6, 32(r2)	Load1	Emitida						
ld **f2**, 44(r2)	Load2	Emitida						
multd f0, f2, f4	Mul1	Emitida						
subd**f8**, f2, f6	Add1	Emitida						
divd f10, f0, f6	Mul2	Emitida						
addd f6, f2, f8		en fetch						

UF	Ocupada	Op	Vj	Vk	Qj	Qk		
Mul1	si	mul		(f4)	Load2			
Mul2	si	div			Mul1	Load1		
Add1	si	sub			Load2	Load1		
Add2	no							
Add3	no							

	f0-1	f2-3	f4-5	f6-7	f8-9	f10-11	…	f30-31
UF productora	Mul1	Load2		Load1	Add1	Mul2		

http://nathantypanski.github.io/tomasulo-simulator/
En este enlace se puede emitir estas instrucciones en el orden deseado y ver su evolución

	UF							
ld f6, 32(r2)	Load1	Emitida	Ejecución					
ld **f2**, 44(r2)	Load2	Emitida						
multd f0, f2, f4	Mul1	Emitida						
subd **f8**, f2, f6	Add1	Emitida						
divd f10, f0, f6	Mul2	Emitida						
addd f6, f2, f8	Add2	Emitida						

UF	Ocupada	Op	Vj	Vk	Qj	Qk		
Mul1	si	mul		(f4)	Load2			
Mul2	si	div			Mul1	Load1		
Add1	si	sub			Load2	Load1		
Add2	si	add			Load2	Add1		
Add3	no							

	f0-1	f2-3	f4-5	f6-7	f8-9	f10-11	...	f30-31
UF productora	Mul1	Load2		~~Load1~~ Add2	Add1	Mul2		

	UF							
ld f6, 32(r2)	Load1	Emitida	Ejecución	Completada		Load1 difunde	M(32+r2)	
ld **f2**, 44(r2)	Load2	Emitida	Ejecución					
multd f0, f2, f4	Mul1	Emitida						
subd **f8**, f2, f6	Add1	Emitida						
divd f10, f0, f6	Mul2	Emitida						
addd f6, f2, f8	Add2	Emitida						

UF	Ocupada	Op	Vj	Vk	Qj	Qk		
Mul1	si	mul		(f4)	Load2			
Mul2	si	div		M(32+r2)	Mul1			
Add1	si	sub		M(32+r2)	Load2			
Add2	si	add			Load2	Add1		
Add3	no							

	f0-1	f2-3	f4-5	f6-7	f8-9	f10-11	...	f30-31
UF productora	Mul1	Load2		Add2	Add1	Mul2		

	UF							
ld f6, 32(r2)	Load1	Emitida	Ejecución	Completada		Load1 ha difundido M(32+r2)		
ld **f2**, 44(r2)	Load2	Emitida	Ejecución	Completada		Load2 difunde M(44+r2)		
multd f0, f2, f4	Mul1	Emitida	Ejecución					
subd **f8**, f2, f6	Add1	Emitida	Ejecución					
divd f10, f0, f6	Mul2	Emitida						
addd f6, f2, f8	Add2	Emitida						

UF	Ocupada	Op	Vj	Vk	Qj	Qk
Mul1	si	mul	*M(44+r2)*	*(f4)*		
Mul2	si	div		M(32+r2)	Mul1	
Add1	si	sub	*M(44+r2)*	*M(32+r2)*		
Add2	si	add	M(44+r2)			Add1
Add3	no					

	f0-1	f2-3	f4-5	f6-7	f8-9	f10-11	...	f30-31
UF productora	Mul1	~~Load2~~		Add2	Add1	Mul2		

	UF							
ld f6, 32(r2)	Load1	Emitida	Ejecución	Completada		Load1 ha difundido M(32+r2)		
ld **f2**, 44(r2)	Load2	Emitida	Ejecución	Completada		Load2 ha difundido M(44+r2)		
multd f0, f2, f4	Mul1	Emitida	Ejecución					
subd **f8**, f2, f6	Add1	Emitida	Ejecución	Completada		Add1 difunde result subd		
divd f10, f0, f6	Mul2	Emitida						
addd f6, f2, f8	Add2	Emitida						
UF	Ocupada	Op	Vj	Vk	Qj	Qk		
Mul1	si	mul	*M(44+r2)*	*(f4)*				
Mul2	si	div		M(32+r2)	Mul1			
Add1	no							
Add2	si	add	*M(44+r2)*	*resultsubd*		~~Add1~~		
Add3	no							
	f0-1	f2-3	f4-5	f6-7	f8-9	f10-11	…	f30-31
UF productora	Mul1			Add2	~~Add1~~	Mul2		

Evolucionar el diagrama con el siguiente orden de completado:

..
addd
multd
divd

Contrastar con esta variante:

..
multd
addd
divd