Arquitectura de Computadores. Curso 2016/2017. Prueba de Teoría del Tema 4

Considere que en un mismo ciclo se decodifican las siguientes cuatro instrucciones (el índice i, i+1,... indica el orden en el que están en el código) y pasan a una ventana de instrucciones centralizada desde la que se emiten a las unidades funcionales. El procesador es de 32 bits y dispone de un buffer de reordenamiento (ROB) que también implementa el renombramiento. Además, tiene DOS unidades de suma, UN multiplicador, y UNA unidad de CARGA/ALMACENAMIENTO de memoria, y puede emitir CUATRO instrucciones en el mismo ciclo, con emisión DESORDENADA y adelantamiento (NO ESPECULATIVO) de Loads a Stores:

Nº ins	Instrucción	Significado	1	2	3	4	5	6	7	8	9
i	mul r4,r2,r1	r4←r2×r1	EX	EX	EX	ROB	WB				
i+1	sw 0(r3), r4	m(r3+0)←r4				EX	WB				
i+2	ld r1,32(r3)	r1←m(r3+32)	EX	EX	ROB		WB				
i+3	add r4,r2,r1	r4←r2+r1			EX	ROB		WB			

9 ——— 2

1. Las instrucciones i e i+1 se pueden emitir en el mismo ciclo

(F)

2. Las instrucciones i+1 e i+2 no se podrían emitir en el mismo ciclo porque entre ellas hay riesgo de tipo RAW

(F)

3. El compilador puede evitar el riesgo WAR entre las instrucciones i+1 e i+3 cambiando el registro r4 en la instrucción i+1 (y SOLO en esa instrucción) por el registro r6

(F)

4. La instrucción i+2 se podría ejecutar antes que la i+1 aunque el procesador no implemente adelantamiento ESPECULATIVO de LOADs a STOREs

(V)

5. Si en el procesador descrito la SUMA tiene un retardo de un ciclo, la CARGA (LD) de memoria dos ciclos, el ALMACENAMIENTO (SW) un ciclo y la MULTIPLICACIÓN tres ciclos ¿Cuántos ciclos tardaría en EJECUTARSE la secuencia de cuatro instrucciones anterior?. (Contando desde el ciclo en que empieza a ejecutarse la primera de las instrucciones)

4 ciclos

6. Suponiendo que el ROB puede retirar tres instrucciones por ciclo. ¿En cuántos ciclos se retirarían las cuatro instrucciones anteriores (contando desde el ciclo en el que se retira la primera de las cuatro instrucciones)?

2 ciclos

7. La segmentación software es una técnica de planificación estática local que, a diferencia del desenrollado, no suele aumentar el tamaño de los bloques básicos

(V)

8. En la predicción dinámica de dos bits, para que cambie el sentido de la predicción (Saltar a No Saltar o viceversa) el predictor SIEMPRE debe fallar dos veces consecutivas

(F)

9. Los procesadores VLIW tienen Buffers de Reordenamiento (ROB) porque la planificación de instrucciones no la realiza el compilador.

(F)

10. En un procesador todas las instrucciones pueden predicarse, y los valores de los predicados se asignan con instrucciones con formato **p1,p2 cmp.cnd r1,r2** donde **cnd** es la condición que se comprueba entre **r1** e **r2 (lt, le, gt, ge, eq, ne)**. ¿Cómo escribiría la sentencia **if ((r1==0) and (r2==0)) then r3=0 else r3=1;** sin instrucciones de salto?

```
      p3 cmp.ne r1,r1
   p1,p2 cmp.eq r1,0
(p1) p3,p2 cmp.eq r2,0
(p3)       add r3,r0,r0
(p2)       add r3,r0,1
```

(Hay otras alternativas, incluso con menos predicados)

Arquitectura de Computadores. Curso 2016/2017. Prueba de Teoría del Tema 4

Considere que en un mismo ciclo se decodifican las siguientes cuatro instrucciones (el índice i, i+1,... indica el orden en el que están en el código) y pasan a una ventana de instrucciones centralizada desde la que se emiten a las unidades funcionales. El procesador es de 32 bits y dispone de un buffer de reordenamiento (ROB) que también implementa el renombramiento. Además, tiene DOS unidades de suma, UN multiplicador, y UNA unidad de CARGA/ALMACENAMIENTO de memoria, y puede emitir CUATRO instrucciones en el mismo ciclo, con emisión DESORDENADA y adelantamiento (NO ESPECULATIVO) de Loads a Stores:

Nº ins	Instrucción	Significado	1	2	3	4	5	6	7	8	9
i	add r4,r2,r1	r4←r2+r1	EX	ROB	WB						
i+1	sw 0(r3), r4	m(r3+0)←r4			EX	WB					
i+2	ld r1,16(r3)	r1←m(r3+16)	EX	EX	ROB	WB					
i+3	mul r4,r2,r1	r4←r2×r1			EX	EX	EX	ROB	WB		

(5) (6)

1. Las instrucciones i+1 e i+2 se pueden emitir en el mismo ciclo

(F)

2. Las instrucciones i+1 e i+3 no se podrían emitir en el mismo ciclo porque entre ellas hay riesgo de tipo WAR

(F)

3. El compilador puede evitar el riesgo WAR entre las instrucciones i+1 e i+3 cambiando el registro r4 en la instrucción i+1 (y SOLO en esa instrucción) por el registro r6

(F)

4. La instrucción i+2 se podría ejecutar antes que la i+1 SOLO si el procesador implementa adelantamiento ESPECULATIVO de LOADs a STOREs,

(F)

5. Si en el procesador descrito la SUMA tiene un retardo de un ciclo, la CARGA (LD) de memoria dos ciclos, el ALMACENAMIENTO (SW) un ciclo y la MULTIPLICACIÓN tres ciclos ¿Cuántos ciclos tardaría en EJECUTARSE la secuencia de cuatro instrucciones anterior?. (Contando desde el ciclo en que empieza a ejecutarse la primera de las instrucciones)

5 ciclos

6. Suponiendo que el ROB puede retirar dos instrucciones por ciclo. ¿En cuántos ciclos se retirarían las cuatro instrucciones anteriores (contando desde el ciclo en el que se retira la primera de las cuatro instrucciones)?

5 ciclos

7. El desenrollado es una técnica de planificación estática local que permite aumentar el tamaño de un bloque básico

(V)

8. En la predicción dinámica de un bit cambia el sentido de la predicción (Saltar a No Saltar o viceversa) si el predictor falla.

(V)

9. Los procesadores VLIW no tienen Buffers de Renombramiento porque la planificación de instrucciones la realiza el compilador.

(V)

10. En un procesador todas las instrucciones pueden predicarse, y los valores de los predicados se asignan con instrucciones con formato **p1,p2 cmp.cnd r1,r2** donde **cnd** es la condición que se comprueba entre **r1** e **r2 (lt, le, gt, ge, eq, ne)**. ¿Cómo escribiría la sentencia **if ((r1==1) and (r2==1)) then r3=0 else r3=1;** sin instrucciones de salto?

```
p3 cmp.ne r1,r1
p1,p2 cmp.eq r1,1
(p1) p3,p2 cmp.eq r2,1
(p3) add r3,r0,r0
(p2) add r3,r0,#1
```

(Hay otras alternativas, incluso con menos predicados)