

Arquitectura de Computadores. Curso 2015/2016. Prueba de Teoría del Tema 4

Considere que en un mismo ciclo se decodifican las siguientes cuatro instrucciones (el índice i , $i+1, \dots$ 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 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 de memoria, y puede emitir CUATRO instrucciones en el mismo ciclo, con emisión DESORDENADA:

Nº inst	Instrucción	Significado	1	2	3	4	5	6	7	8	9
i	ld r2, 0(r3)	$r2 \leftarrow m(r3+0)$									
$i+1$	mul r4,r2,r1	$r4 \leftarrow r2 \times r1$									
$i+2$	add r1,r1,r2	$r1 \leftarrow r1+r2$									
$i+3$	add r4,r3,r5	$r4 \leftarrow r3+r5$									
Pregunta 5											
Pregunta 6											

1. Las instrucciones i e $i+3$ se podrían emitir en el mismo ciclo
2. Las instrucciones $i+1$ e $i+2$ no se podrían emitir en el mismo ciclo porque entre ellas hay riesgo de tipo WAR
3. El compilador puede evitar el riesgo WAW entre las instrucciones $i+1$ e $i+3$ utilizando el registro $r6$ en lugar de $r4$ en la instrucción $i+1$
4. En el procesador que se ha descrito al comienzo, aunque la emisión es desordenada, la finalización de instrucciones es ordenada.
5. Si en el procesador descrito la SUMA tiene un retardo de un ciclo, la CARGA de memoria un retardo de dos ciclos, y la MULTIPLICACIÓN un retardo de cuatro ciclos ¿Cuántos ciclos tardaría en EJECUTARSE la secuencia de cuatro instrucciones anterior?. (Contando desde el ciclo en que termina de ejecutarse la primera de las cuatro instrucciones)
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)?
7. Las instrucciones de movimiento condicional de datos permiten reducir el número de instrucciones de salto condicional en los códigos.
8. La predicción de saltos dinámica implícita para una instrucción de salto condicional, i , utiliza el resultado (salto o no salto) de la ejecución previa de dicha instrucción de salto condicional, i , para hacer la predicción
9. Los procesadores WLIW no tienen buffers de renombramiento porque la planificación de instrucciones la realiza el compilador.

10. Considere las dos instrucciones siguientes (la instrucción i precede a la $i+1$ en el código

(i) `sw 0(r5),r2 // $m(r5+0) \leftarrow r2$`

($i+1$) `ld r4,0(r6) // $r4 \leftarrow m(r6+0)$`

Un procesador que NO implemente adelantamiento ESPECULATIVO de LOADs a STORES podría adelantar la ejecución de $i+1$ a la de i