

Exámenes

UT2 (2.4,2.5) - Prueba de seguimiento

[Tabla de Contenidos](#)

Tiempo restante: 0:20:25

[Ocultar/Mostrar el tiempo restante](#)

Parte 1 de 4 -

Preguntas 1 de 8

1 Puntos

Cuando aplicamos Tomasulo con especulación hardware a las instrucciones, éstas terminan su ejecución en la etapa , liberando la estación de reserva correspondiente y escribiendo el resultado de la operación en .

Preguntas 2 de 8

1 Puntos

Indica cuales de las siguientes afirmaciones son correctas:

☐

A. Las instrucciones especulativas pueden generar excepciones durante su ejecución, aunque éstas se tratarán cuando las instrucciones sean confirmadas.

☐

B. Las instrucciones especulativas pueden ejecutarse fuera de orden.

☐

C. A la fase Commit pueden llegar instrucciones especuladas incorrectamente, pero éstas se cancelan.

☐

D. Una de las ventajas de la especulación hardware es que permite la confirmación de instrucciones fuera de orden.

Preguntas 3 de 8

1 Puntos

Gestionar dinámicamente las instrucciones a ejecutar conlleva el mantenimiento de un grafo de dependencias. Se añaden nuevas entradas al grafo cuando las instrucciones pasan por la etapa , mientras que se eliminan cuando pasan por la etapa .

Parte 2 de 4 -

Preguntas 4 de 8

1 Puntos

Indica cuales de las siguientes afirmaciones son correctas:

☐

A. Los procesadores con un formato de instrucción muy largo codifican varias operaciones en una sola instrucción y su hardware se encarga de gestionarlas dinámicamente.

☐

B. Los procesadores con un formato de instrucción muy largo son compatibles a nivel binario con las máquinas de partida de tipo escalar.

☐

C. Los procesadores con un formato de instrucción muy largo ofrecen una mejora de prestaciones que está íntimamente relacionada con la calidad del compilador utilizado.

☐

D. Los procesadores con un formato de instrucción muy largo requieren del uso de técnicas de compilación especiales que eviten empaquetar en una misma instrucción varios NOPs.

Preguntas 5 de 8

1 Puntos. Puntos descontados por fallo: 0.3

Los procesadores superescalares buscan reducir el tiempo de ejecución de los programas aumentando la instrucciones lanzadas a ejecución por ciclo.

☒ Verdadero☐ Falso[Borra selección](#)

Preguntas 6 de 8

1 Puntos. Puntos descontados por fallo: 0.3

Los procesadores VLIW aumentan la complejidad de las instrucciones máquina y el trabajo que éstas realizan con el objetivo de reducir el tiempo de ejecución de los programas.

☒ Verdadero☐ Falso[Borra selección](#)

Parte 3 de 4 -

Preguntas 7 de 8

2 Puntos

El siguiente código ensamblador para un procesador MIPS implementa la operación $Z = aX + bY$ condicionada al contenido del vector de mascara M:

```
loop:  LD R3, M(R1)
      BEQZ R3, endif    ; Salta si M[i] == 0
      L.D F2, X(R1)     ; Valor inicial de R1 = 0
```

L.D F3, Y(R1)

MULT.D F3, F1, F3 ; F1 contiene b = 5.0

MULT.D F2, F0, F2 ; F0 contiene a = 2.5

ADD.D F4, F2, F3

S.D F4, Z(R1)

endif: DADD R1, R1, #8

BNE R1, R2, loop ; R2 contiene 64 * 8

Su ejecución en un procesador que aplica planificación dinámica de instrucciones con especulación hardware genera el siguiente diagrama instrucciones tiempo:

Instruc.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
ld r3,m(r1)	IF	I	AC	L1	L2	WB	C															
beqz r3,endif		IF	I				E1	WB	C													
dadd r1,r1,#8			IF	I	E1	-	WB		x													
bne r1,r2,loop				IF	I			E1	X													
ld r3,m(r1)					IF	I		AC	X													
beqz r3,endif						IF	I		x													
dadd r1,r1,#8							IF	I	X													
bne r1,r2,loop								IF	X													
ld r3,m(r1)									X													
l.d f2,x(r1)										IF	I	AC	L1	L2	WB	C						
l.d f3,y(r1)											IF	I	AC	L1	L2	WB	C					
mul.d f3,f1,f3												IF	I				M1	M2	M3	M4		
mul.d f2,f0,f2													IF	I			M1	M2	M3	M4	WB	
add.d f4,f2,f3														IF	I							
s.d f4,z(r1)															IF	I	AC					
dadd r1,r1,#8																IF	I	E1	WB			
bne r1,r2,loop																	IF	I		E1		
ld r3,m(r1)																		IF	I	AC		
beqz r3,endif																			IF	I		
l.d f2,x(r1)																				IF		

Rellena la siguiente tabla para indicar cuál sería la evolución durante los ciclos 21 y 22 de las instrucciones no confirmadas del diagrama instrucciones-tiempo mostrado. Indica las etapas que atraviesa cada instrucción. Recuerda que según la instrucción a ejecutar sus etapas recibirán una denominación distinta: L1, L2 para las instrucciones de carga/almacenamiento; M1, M2, ... para las multiplicaciones/divisiones; A1, A2, ... para las sumas y restas en coma flotante; E1 para las instrucciones de aritmética entera y los saltos. **No dejes en blanco un ciclo cuando la instrucción no haga nada útil en el mismo. Utiliza en ese caso el guión (-).**

Instrucciones	21	22
---------------	----	----

mul.d f3, f1, f3	WB		C
mul.d f2, f0, f2	-		-
add.d f4, f2, f3	E1		WB
s.d f4, z(r1)	-		-
dadd r1, r1, #8	-		-
bne r1, r2, loop	X		-
ld r3, m(r1)	X		-
beqz r3, endif	X		-
l.d f2, x(r1)	X		-

Parte 4 de 4 -

Preguntas 8 de 8

2 Puntos

Se ha ejecutado un bucle en un procesador superescalar de 2 vías que aplica planificación dinámica de instrucciones basada en el algoritmo de Tomasulo con especulación. El lanzamiento de las instrucciones se realiza en orden y alineado. El bucle ejecutado comienza en la posición de memoria *loop*. El diagrama instrucciones-tiempo resultante de la ejecución de la primera iteración del bucle es el siguiente:

PC	Instruc.	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>
.text	dsub r1,r1,#8	IF	I	E1	WB	C		
.text + 04	ld r1,n(r0)	IF	I	AC	L1	L2		
.text + 08	l.d f0,a(r0)		IF	I	AC			
loop	l.d f2,x(r1)		IF	I				
loop + 04	mul.d f2,f2,f0			IF	I			
loop + 08	add.d f2,f2,f0			IF	I			
loop + 12	s.d f2,y(r1)				IF	I		
loop + 16	dsub r1,r1,#8				IF	I		
loop + 20	bnez r1,loop					IF		
loop + 24	trap 0					IF		

Las características de las unidades funcionales de la unidad de ejecución de instrucciones son las siguientes:

Unidad	Nº Operadores	Latencia	Segmentado	Características	Etapas
Carga/Almacenamiento	1	2	No	3 TL, 3 TE	L1, L2
Sumador/Restador c.f.	1	2	No	3 ER	A1, A2
Multiplicador c.f.	1	4	Sí	2 ER	M1, M2, M3, M4
Enteros/Saltos	1	1	No	5 ER	E1

Se pide rellenar el diagrama instrucciones-tiempo que se presenta a continuación para los ciclos 6 y 7. En el diagrama aparecen las instrucciones que ya hay en ejecución, pero se deben incluir aquellas que se buscan en memoria durante los dos ciclos considerados. En caso de no tener que hacer nada en un ciclo para una instrucción **NO DEJES NINGUNA CELDA DE LA TABLA EN BLANCO utiliza un guión (-) para señalar dicha situación.**

PC	Instrucciones	6	7
.text	dsub r1, r1, #8	-	-
.text+04	ld r1, n(r0)	WB	C

.text+08	l.d f0, a(r0)	L1	L2
loop	l.d f2, x(r1)	-	AC
loop+04	mul.d f2, f2, f0	-	-
loop+08	add.d f2, f2, f0	-	-
loop+12	s.d f2, y(r1)	-	-
loop+16	dsub r1, r1, #8	E1	WB
loop+20	bnez r1, loop	I	-
loop+24	trap 0	I	E1
loop	l.d f2, x(r1)	IF	I
loop+04	mul.d f2, f2, f0	IF	I
loop+08	add.d f2, f2, f0	-	IF
loop+12	s.d f2, y(r1)	-	IF

- [PoliformaT](#)
- [UPV](#)
- [Powered by Sakai](#)
- Copyright 2003-2018 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.