

## Exámenes

### UT2B (2.4,2.5) - Prueba de seguimiento

[Volver a la Lista de Exámenes](#)

Parte 1 de 5 -

2.13/ 2.5 Puntos

Preguntas 1 de 13

0.25/ 0.5 Puntos

Indica cuales de las siguientes afirmaciones son correctas:

- ☐ Las instrucciones especulativas pueden generar excepciones durante su ejecución, aunque éstas se tratarán cuando las instrucciones sean confirmadas.
- ☐ A la fase Commit pueden llegar instrucciones especuladas incorrectamente, pero éstas se cancelan.
- ☒ Las instrucciones especulativas pueden ejecutarse fuera de orden.
- ☐ Una de las ventajas de la especulación hardware es que permite la confirmación de instrucciones fuera de orden.

Respuesta correcta: A, C

Preguntas 2 de 13

0.38/ 0.5 Puntos

En un procesador que aplica el algoritmo de Tomasulo con especulación hardware, indica cuáles de las siguientes afirmaciones son falsas:

- ☒ La planificación dinámica de instrucciones basada en el algoritmo de Tomasulo con especulación hardware sólo afecta a la ejecución de instrucciones multiciclo, como las de aritmética en coma flotante.
- ☐ Una instrucción dependiente solo podrá obtener sus operandos de los registros fuente que utilice, y si éstos estuvieran reservados por otra instrucción, anotarse la marca que éstos contengan.
- ☒ Las estaciones de reserva almacenan las instrucciones desde que se decodifican (etapa I) hasta que se confirman (etapa C).
- ☒ Cuando la instrucción llega a la etapa WB, difunde su resultado a través del bus común de datos para que todas las estaciones de reserva, buffers y registros que requieran de dicho valor puedan recuperarlo.

Respuesta correcta: A, B, C, D

¿Qué puedes decir de la instrucción que confirma su ejecución en un procesador que aplica planificación dinámica de instrucciones basada en el

algoritmo de Tomasulo con especulación ? Marca todas las opciones correctas.

- ☐ Que si es un salto incorrectamente predicho, se cancelarán todas las instrucciones existentes en el ROB, liberando todas las estaciones de reserva y buffers de escritura y lectura.
- ☒ Que si es una instrucción de almacenamiento, la operación pasará a realizarse en el siguiente ciclo de reloj.
- ☒ Que es la instrucción más antigua existente en el ROB.
- ☐ Que si es una instrucción aritmética, el resultado almacenado en el ROB se copiará en el registro destino y éste se liberará .

Respuesta correcta: B, C

Dada la siguiente secuencia de instrucciones que se ejecuta en procesador con ejecución fuera de orden:

1 2 3 4 5 6 7 8 9

add.d f0, f1, f2 IF I A1 A2 A3 WB ...

add.d f3, f4, f5 IF I

Indica en qué ciclo de reloj iniciará su ejecución la segunda instrucción para las opciones indicadas:

1. Hay sólo un operador de suma fp no segmentado. Ciclo ☒ 6
2. Hay dos operadores de suma fp. Ciclo ☒ 3
3. Hay sólo un operador de suma fp segmentado. Ciclo ☒ 4

Respuesta correcta: 6, 3, 4

Indica cuáles de las siguientes respuestas son correctas en el contexto de un procesador con gestión dinámica de instrucciones y especulación hardware:

- 
- ☐ Ninguna instrucción de almacenamiento contenida en el ROB debe cancelarse cuando una instrucción de salto que ha sido incorrectamente predicha llegue a la fase Commit.
  - ✓ ☒ Las instrucciones de almacenamiento escriben en memoria después de ser confirmadas.
  - ☐ Las instrucciones de almacenamiento (los stores) confirman su ejecución en la etapa WB.
  - ✓ ☒ Las instrucciones de almacenamiento (los stores) necesitan del valor de dos registros para poderse ejecutar.

**Respuesta correcta:** B, D

Parte 2 de 5 -

2.5/ 2.5 Puntos

Preguntas 6 de 13

0.5/ 0.5 Puntos

Las figuras siguientes muestran el diagrama instrucciones-tiempo correspondiente a la ejecución de dos aplicaciones A y B sobre un procesador superescalar de 4 vías.

Ciclo 0Ciclo 1Ciclo 2

A		
A	A	
A	A	A

Ciclo 0Ciclo 1Ciclo 2

		B
	B	B
B	B	B

Completa el siguiente diagrama instrucciones-tiempo, correspondiente a la ejecución de las dos aplicaciones A y B sobre un procesador multihilo con multithreading simultáneo (SMT). Si para un mismo ciclo se ejecutan las dos aplicaciones, ubica la aplicación A en las filas inferiores, y rellena las casillas de abajo a arriba. ¡¡No dejes ninguna celda en la tabla en blanco!! Utiliza el guión (-), y pon sólo un guión, en aquellas celdas en las que no se ejecute ni la tarea A, ni la B.

Ciclo 0Ciclo 1Ciclo 2Ciclo 3

✓B	✓B	✓B	✓-
✓A	✓B	✓B	✓-
✓A	✓A	✓B	✓-
✓A	✓A	✓A	✓-

Respuesta correcta: B, B, B, -, A, B, B, -, A, A, B, -, A, A, A, -

Preguntas 7 de 13

0.5/ 0.5 Puntos. Puntos descontados por fallo: 0.15

Los procesadores superescalares buscan reducir el tiempo de ejecución de los programas aumentando las instrucciones lanzadas a ejecución en cada ciclo de reloj.

✓ ☐ Verdadero  
☒ Falso

Respuesta correcta: Verdadero

## Preguntas 8 de 13

0.5/ 0.5 Puntos. Puntos descontados por fallo: 0.15

Los procesadores VLIW empaquetan varias operaciones en cada instrucción máquina, con el objetivo de reducir el número de instrucciones ejecutadas por los programas.

- ☒ Verdadero  
☐ Falso

**Respuesta correcta:** Verdadero

## Preguntas 9 de 13

0.5/ 0.5 Puntos

Las figuras siguientes muestran el diagrama instrucciones-tiempo correspondiente a la ejecución de dos aplicaciones A y B sobre un procesador superescalar de 4 vías.

Ciclo Ciclo Ciclo Ciclo

0 1 2 3

A

A

A A

A A

Ciclo 0

Ciclo 1

B

B B

B B

Completa el siguiente diagrama instrucciones-tiempo, correspondiente a la ejecución de las dos aplicaciones A y B sobre un procesador multihilo de grano grueso. Supón que un evento es de alta latencia cuando dura más de un ciclo, que empieza ejecutándose la aplicación A y que las casillas se rellenan de abajo a arriba. ¡¡No dejes ninguna celda en la tabla en blanco!! Utiliza el guión (-), y pon sólo un guión, en aquellas celdas en las que no se ejecute ni la tarea A, ni la B.

Ciclo 0Ciclo 1Ciclo 2Ciclo 3

✓A ✓- ✓- ✓-

✓A ✓- ✓B ✓-

✓A ✓B ✓B ✓A

✓A ✓B ✓B ✓A

Respuesta correcta: A, -, -, A, -, B, -, A, B, B, A, A, B, B, A

## Preguntas 10 de 13

0.5/ 0.5 Puntos

**Importante:** utiliza únicamente el punto como separador decimal.

Considera un procesador superescalar de 4 vías, que utiliza un predictor de saltos BTB que obtiene su predicción al final de la etapa IF. La cache de instrucciones ofrece un grupo de 4 instrucciones alineadas. Si en un grupo de 4 instrucciones aparece una instrucción de salto que se predice como que "salta", indica el número de instrucciones del grupo que se desecharán:

1. Si el salto es la última instrucción del grupo: ☒ 0
2. Si el salto es la primera instrucción del grupo: ☒ 3

**Respuesta correcta:** 0,3

Parte 3 de 5 -

1.0/ 1.5 Puntos

## Preguntas 11 de 13

1.0/ 1.5 Puntos

La ejecución de la primera iteración de un bucle en un procesador que aplica gestión dinámica de instrucciones con especulación hardware proporciona el siguiente diagrama instrucciones-tiempo:

PC	Instruc.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
loop	<code>l.d f2,%disp[x](r2)</code>	IF	I	AC	L1	L2	WB	C													
4100	<code>l.d f4,%disp[y](r2)</code>	IF	I	AC	L1	L2	WB	C													
4104	<code>mul.d f2,f2,f0</code>	IF	I					M1	M2	M3	M4	WB	C								
4108	<code>mul.d f4,f4,f1</code>	IF	I					M1	M2	M3	M4	WB	C								
4112	<code>add.d f6,f3,f2</code>		IF	I									A1	A2	WB	C					
4116	<code>add.d f6,f6,f4</code>		IF	I												A1	A2	WB	C		
4120	<code>s.d f6,%disp[y](r2)</code>		IF	I	AC														C	L1	L2
4124	<code>dadd r2,r2,8</code>		IF	I	E1	WB													C		
4128	<code>dsub r1,r1,1</code>			IF	I	E1	WB												C		
4132	<code>bnez r1,loop</code>			IF	I			E1	WB											C	
4136	<code>nop</code>			IF	X																
4140	<code>nop</code>			IF	X																
loop	<code>l.d f2,%disp[x](r2)</code>				IF	I	AC	L1	L2	WB										C	
4100	<code>l.d f4,%disp[y](r2)</code>				IF	I	AC	L1	L2	WB										C	
4104	<code>mul.d f2,f2,f0</code>				IF	I					M1	M2	M3	M4	WB					C	
4108	<code>mul.d f4,f4,f1</code>				IF	I					M1	M2	M3	M4	WB						C

El código asociado a cada iteración del bucle comienza en la posición *loop* y termina en la posición *4140*. A la vista del diagrama instrucciones-tiempo suministrado, responde a las siguientes preguntas:

1. ¿Cuántos ciclos tarda una iteración del bucle en ejecutarse si el predictor acierta? ☒ 3 ciclos
2. ¿Cuántos ciclos tarda una iteración del bucle en ejecutarse si el predictor falla? ☒ 19 ciclos
3. Asumiendo que se utiliza un BTB que asocia un predictor de 2 bits con histéresis a cada salto, y que antes de ejecutar el bucle, la instrucción de salto del mismo no está en el BTB, ¿cuál será el CPI promedio (expresado con 2 decimales y utilizando el punto decimal) resultante de procesar unos vectores X e Y de 64 elementos cada uno? El CPI promedio será de ~~X~~ 0.29 ciclos/instrucción.

Respuesta correcta: 3, 19, 0.35

Parte 4 de 5 -

2.0/ 2.0 Puntos

Preguntas 12 de 13

2.0/ 2.0 Puntos

Se ejecuta una aplicación en un procesador con gestión dinámica de instrucciones basada en el algoritmo de Tomasulo y especulación hardware. El resultado de los primeros 17 ciclos de la ejecución es el siguiente:

PC	Instruc.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
.text	l.d f0,a(r0)	IF	I	AC	L1	L2	WB	C													
salto	l.d f2,x(r1)		IF	I	AC		L1	L2	WB	C											
salto + 04	mul.d f4,f2,f0			IF	I					M1	M2	WB	C								
salto + 08	add.d f6,f4,f4				IF	I							A1	A2	WB	C					
salto + 12	sub.d f8,f4,f4					IF	I							A1	A2	WB	C				
salto + 16	dsub r1,r1,#8						IF	I	E1	WB								C			

Completa los ciclos 18, 19 y 20 para cada instrucción a partir de la primera instrucción que no ha realizado Commit. Supóngase que la instrucción bnez r1, salto es "efectiva". Rellena las instrucciones que se buscan en esos ciclo. Para resolver el ejercicio rellena la siguiente tabla. Si una instrucción no realiza ninguna actividad en un ciclo márcalo con un guión (-). En otras palabras, rellena toda las celdas de la tabla.



Instrucción	17	18	19	20
dsub r1,r1,#8	C			
add.d f10,f20,f0	✓ <u>C</u>	✓ <u>-</u>	✓ <u>-</u>	
bnez r1,salto	✓ <u>-</u>	✓ <u>C</u>	✓ <u>-</u>	
l.d f2,x(r1)	✓ <u>-</u>	✓ <u>-</u>	✓ <u>C</u>	
mul.d f4,f2,f0	M1	✓ <u>M2</u>	✓ <u>WB</u>	✓ <u>-</u>
add.d f6,f4,f4		✓ <u>-</u>	✓ <u>-</u>	✓ <u>A1</u>
sub.d f8,f4,f4		✓ <u>-</u>	✓ <u>-</u>	✓ <u>-</u>
dsub r1,r1,#8	WB	✓ <u>-</u>	✓ <u>-</u>	✓ <u>-</u>
add.d f10,f2,f0	A1	✓ <u>A2</u>	✓ <u>-</u>	✓ <u>WB</u>
bnez r1,salto		✓ <u>E1</u>	✓ <u>-</u>	✓ <u>-</u>
l.d f2,x(r1)	I	✓ <u>AC</u>	✓ <u>L1</u>	✓ <u>L2</u>
mul.d f4,f2,f0	IF	✓ <u>I</u>	✓ <u>-</u>	✓ <u>-</u>
✓ <u>add.d f6,f4,f4</u>		✓ <u>IF</u>	✓ <u>I</u>	✓ <u>-</u>
✓ <u>sub.d f8,f4,f4</u>		-	✓ <u>IF</u>	✓ <u>I</u>
✓ <u>dsub r1,r1,#8</u>		-	-	✓ <u>IF</u>

Respuesta correcta: C, -, -, -, C, -, -, -, C, M2, WB, -, -, -, A1, -, -, -, -, A2, -, WB, E1, -, -, AC, L1, L2, I, -, -, add.d f6,f4,f4, IF, I, -, sub.d f8,f4,f4, IF, I, dsub r1,r1,#8, IF

Parte 5 de 5 -

1.13/ 1.5 Puntos

## Preguntas 13 de 13

1.13/ 1.5 Puntos

El siguiente cronograma muestra la ejecución de un fragmento de código en un procesador con gestión dinámica de instrucciones y especulación:

PC	Instruc.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
salto	<code>l.d f0,x(r0)</code>	IF	I	AC	L1	L2	WB	C											
salto + 04	<code>s.d f0,y(r0)</code>		IF	I	AC				C	L1	L2								
salto + 08	<code>add.d f4,f0,f2</code>			IF	I			A1	A2	WB	C								
salto + 12	<code>s.d f4,z(r0)</code>				IF	I	AC					C	L1	L2					
salto + 16	<code>dsub r1,r1,#8</code>					IF	I	E1	WB				C						
salto + 20	<code>bnez r1,salto</code>						IF	I		E1	WB			C					
salto + 24	<code>trap 0</code>							IF	I					x					
.etext	<code>&lt;nop&gt;</code>								if	if	if	if	if	X					
salto	<code>l.d f0,x(r0)</code>														IF	I	AC	L1	L2
salto + 04	<code>s.d f0,y(r0)</code>															IF	I	AC	
salto + 08	<code>add.d f4,f0,f2</code>																IF	I	
salto + 12	<code>s.d f4,z(r0)</code>																	IF	I
salto + 16	<code>dsub r1,r1,#8</code>																		IF

responde a las siguientes preguntas sobre el estado del procesador al final del ciclo 12:

- ¿Cuántas entradas en el ROB están activas? ☒ 2
- ¿Cuántos buffers de lectura están ocupados? ☒ 0
- ¿Cuántos buffers de escritura están confirmados? ☒ 2
- ¿Cuántos registros de coma flotante tienen una marca activa? ☒ 0

Respuesta correcta: 2, 0, 1, 0

- [PoliformaT](#)
- [UPV](#)
- [Powered by Sakai](#)
- Copyright 2003-2020 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.