

## Conjunto 1 Evaluación Continua Tema 4

1 pregunta

1  
VF

Un procesador de 32 bits, puede captar CUATRO instrucciones por ciclo, DECODIFICAR CUATRO instrucciones por ciclo, dispone de una ventana de instrucciones centralizada desde donde se produce la EMISIÓN DESORDENADA de hasta DOS instrucciones por ciclo. También dispone de un buffer de reordenamiento (ROB) donde implementa el renombramiento y la finalización ordenada, pudiendo retirarse desde el ROB, CUATRO instrucciones por ciclo para escribir sus resultados en la etapa WB del cauce. El procesador incluye además DOS unidades de Suma/Resta de UN ciclo; UN multiplicador de TRES ciclos, y DOS unidades de Carga/Almacenamiento de memoria de DOS ciclos. Por tanto, las etapas del cauce del procesador son: IF (captación de instrucciones), ID (Decodificación de instrucciones), EX (ejecución en unidad funcional de la operación codificada por la instrucción), y WB (retirada de la instrucción del ROB y escritura de resultado en los registros del procesador). No se considera etapa ROB explícita en el cauce porque se supone que en el último ciclo de ejecución de las unidades funcionales, el resultado de la operación queda almacenado en el ROB. Las instrucciones de almacenamiento (escritura en memoria) no utilizan etapa WB.

Para este procesador y la secuencia de instrucciones siguiente

```
(1) ld r1,0(r3)    // r1=M(r3)
(2) ld r2,8(r3)    // r2 = M(r3+8)
(3) add r2, r1, r2  // r2=r1+r2
(4) mul r4, r1, r2  // r4= r1*r2
(5) sub r4, r1, r4  // r4=r1-r4
(6) add r6, r1, r3  // r6=r1+r3
```

La instrucción (6) empieza a ejecutarse en el ciclo 5 (los ciclos se numeran desde 1) y la secuencia de instrucciones tarda en procesarse 12 ciclos.

1-v

2-v

3-v

4-v

5-v

6-f

7-f

8-v

9-v

10-f

## Conjunto 2 Evaluación Continua Tema 4

1 pregunta

2  
VF

Suponga que en un mismo ciclo se decodifican las cuatro instrucciones siguientes

```
(i)  add r2, r2, r1    // r2 = r2+r1
(i+1) mul r3, r4, r1   // r3 = r4*r1
(i+2) add r4, r4, r1   // r4 = r4+r1
(i+3) sub r5, r2, r1   // r5 = r2-r1
```

(entre paréntesis se indica el orden en el que están en el código), y pasan a una ventana de instrucciones única desde la que se emiten a las distintas unidades funcionales. El procesador puede emitir (con emisión DESORDENADA) CUATRO instrucciones por ciclo y tiene TRES unidades de suma/resta (con un retardo de 1 ciclo) y UN multiplicador (con un retardo de 3 ciclos).

Las instrucción (i+1) se puede emitir en el mismo ciclo que la (i)

## Conjunto 3 Evaluación Continua Tema 4

1 pregunta

3  
VF

Es posible implementar el renombramiento en un buffer de reorden (ROB)

## Conjunto 4 Evaluación Continua Tema 4

1 pregunta

4  
VF

Las instrucciones de tipo centinela facilitan la aplicación del adelantamiento especulativo de loads a stores en procesadores VLIW

## Conjunto 5 Evaluación Continua Tema 4

1 pregunta

5  
VF

Suponga un procesador donde todas las instrucciones pueden predicarse, y los valores de los predicados se asignan con instrucciones con los formatos p1,p2 cmp.cnd r1,r2, o también p1 cmp.cnd r1,r2 donde cnd es la condición que se comprueba entre r1 y r2 (lt, le, gt, ge, eq, ne, respectivamente, menor que, menor o igual que, mayor que, mayor o igual que, igual, distinto). En dicho procesador, la sentencia if ((r1==0) and (r2==0)) then r3=0 else r3=1 puede escribirse sin instrucciones de salto como

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

donde r0=0 siempre y los predicados, p1, p2, y p3 pueden tener cualquier valor (0 ó 1) al iniciarse la ejecución del código

## Conjunto 6 Evaluación Continua Tema 4

1 pregunta

6  
VF

En la predicción dinámica de dos bits, el sentido de la predicción siempre cambia (de saltar a no saltar o de no saltar a saltar) cuando el predictor falla

## Conjunto 7 Evaluación Continua Tema 4

1 pregunta

7  
VF

Para implementar un procedimiento de predicción dinámica de saltos es preciso utilizar bits de historia.

## Conjunto 8 Evaluación Continua Tema 4

1 pregunta

8  
VF

Un procesador de 32 bits (4 bytes) que implementa adelantamiento de loads a stores, aunque no implementa el adelantamiento ESPECULATIVO, podría adelantar la instrucción i+1 a la i (i precede a i+1 en el código)

```
(i)  sw 0(r6), r2      // M(r6)<-- r2
(i+1) lw r4, 8(r6)     // r4 <-- M(r6+8)
```

## Conjunto 9 Evaluación Continua Tema 4

1 pregunta

9  
VF

Un procesador superescalar de 2 GHz cuyo ROB es capaz de retirar dos instrucciones por ciclo tiene una velocidad pico de 4 GIPS (Giga-instrucciones por segundo)

## Conjunto 10 Evaluación Continua Tema 4

1 pregunta

10  
VF

El principal responsable del aprovechamiento eficiente del paralelismo entre instrucciones (ILP) en un procesador superescalar es el compilador.

1	Un procesador de 32 bits, puede captar CUATRO instrucciones por ciclo, DECODIFICAR CUATRO instrucciones por ciclo, y dispone de una ventana de instrucciones centralizada desde donde se produce la EMISIÓN DESORDENADA de hasta DOS instrucciones por ciclo. También dispone de un buffer de reordenamiento (ROB) donde implementa el renombramiento y la finalización ordenada, pudiendo retirarse desde el ROB, CUATRO instrucciones por ciclo para escribir sus resultados en la etapa WB del curso. El procesador incluye además DOS unidades de Suma/Resta de 1IN ciclo; UN multiplicador de TRES ciclos, y DOS unidades de Carga/Almacenamiento de memoria de DOS ciclos. Por tanto, las etapas del cauce del procesador son: IF (captación de instrucciones), ID (Decodificación de instrucciones), EX (ejecución en unidad funcional de la operación codificada por la instrucción), y WB (retirada de la instrucción del ROB y escritura de resultado en los registros del procesador). No se considera etapa ROB explícita en el cauce porque se supone que en el último ciclo de ejecución de las unidades funcionales, el resultado de la operación queda almacenado en el ROB. Las instrucciones de almacenamiento (escritura en memoria) no utilizan etapa WB.
V/F	<p>Para este procesador y la secuencia de instrucciones siguiente</p> <pre>(1) ld r1,0(r3) // r1=M(r3) (2) ld r2,8(r3) // r2 = M(r3+8) (3) add r2, r1, r2 // r2=r1+r2 (4) mul r4, r1, r2 // r4= r1*r2 (5) sub r4, r1, r4 // r4=r1-r4 (6) add r6, r1, r3 // r6=r1+r3</pre> <p>La instrucción (6) empieza a ejecutarse en el ciclo 5 (los ciclos se numeran desde 1) y la secuencia de instrucciones tarda en procesarse 12 ciclos.</p> <p>Usuario Profesores</p> <p><b>F</b> </p>
2	Suponga que en un mismo ciclo se decodifican las cuatro instrucciones siguientes
V/F	<pre>(i) add r2, r2, r1 // r2 = r2+r1 (i+1) mul r3, r4, r1 // r3 = r4*r1 (i+2) add r4, r4, r1 // r4 = r4+r1 (i+3) sub r5, r2, r1 // r5 = r2-r1</pre> <p>(entre paréntesis se indica el orden en el que están en el código), y pasan a una ventana de instrucciones única desde la que se emiten a las distintas unidades funcionales. El procesador puede emitir (con emisión DESORDENADA) CUATRO instrucciones por ciclo y tiene TRES unidades de suma/resta (con un retardo de 1 ciclo) y UN multiplicador (con un retardo de 3 ciclos).</p> <p>Las instrucción (i) NO se puede emitir en el mismo ciclo que la (i+3)</p> <p>Usuario Profesores</p> <p><b>V</b> </p>
3	El aprovechamiento eficiente del paralelismo entre instrucciones (ILP) en un procesador VLIW necesita el uso del ROB
V/F	<p>Usuario Profesores</p> <p><b>F</b> </p>
4	El compilador de un procesador VLIW no puede aplicar ninguna técnica de planificación estática global porque las condiciones de las instrucciones de salto que separan los bloques básicos solo se pueden conocer durante la ejecución del programa.
V/F	<p>Usuario Profesores</p> <p><b>F</b> </p>
5	Suponga un procesador donde todas las instrucciones pueden predicarse, y los valores de los predicados se asignan con instrucciones con los formatos p1,p2 cmp.cnd r1,r2, o también p1 cmp.cnd r1,r2 donde cnd es la condición que se comprueba entre r1 y r2 (lt, le, gt, ge, eq, ne, respectivamente, menor que, menor o igual que, mayor que, mayor o igual que, igual, distinto). En dicho procesador, la sentencia if ((r1==0) and (r2==0)) then r3=0 else r3=1 puede escribirse sin instrucciones de salto como
V/F	<pre>p3 comp.eq r1,r1 p1,p2 cmp.eq r1,r0 (p1) p3,p2 cmp.eq r2,r0 (p3) add r3,r0,r0 // r3=r0+r0 (p2) add r3, r0,#1 //r3=r0+1</pre> <p>donde r0=0 siempre y los predicados, p1, p2, y p3 pueden tener cualquier valor (0 ó 1) al iniciarse la ejecución del código</p> <p>Usuario Profesores</p> <p><b>V</b> </p>
6	En la predicción dinámica de dos bits, el estado del predictor (los bits de historia) no cambia si se acierta la predicción.
V/F	<p>Usuario Profesores</p> <p><b>F</b> </p>
7	Para implementar un procedimiento de predicción dinámica de saltos es preciso utilizar bits de historia.
V/F	<p>Usuario Profesores</p> <p><b>F</b> </p>
8	Un procesador podría adelantar la instrucción i+1 a la i (i precede a i+1 en el código) si implementa adelantamiento de LOADs a STORES ESPECULATIVO
V/F	<pre>(i) sw 0(r5), r2 // M(r5)&lt;-- r2 (i+1) lw r4, 0(r6) // r4 &lt;--M(r6)</pre> <p>Usuario Profesores</p> <p><b>V</b> </p>
9	Un procesador superescalar de 2 GHz cuyo ROB es capaz de retirar dos instrucciones por ciclo tiene una velocidad pico de 4 GIPS (Giga-instrucciones por segundo)
V/F	<p>Usuario Profesores</p> <p><b>V</b> </p>
10	El principal responsable del aprovechamiento eficiente del paralelismo entre instrucciones (ILP) en un procesador superescalar es el compilador.
V/F	<p>Usuario Profesores</p> <p><b>V</b> </p>

**2** Suponga que en un mismo ciclo se decodifican las cuatro instrucciones siguientes

V/F

(i) add r2, r2, r1 // r2 = r2+r1  
(i+1) mul r3, r4, r1 // r3 = r4\*r1  
(i+2) add r4, r4, r1 // r4 = r4+r1  
(i+3) sub r5, r2, r1 // r5 = r2-r1

(entre paréntesis se indica el orden en el que están en el código), y pasan a una ventana de instrucciones única desde la que se emiten a las distintas unidades funcionales. El procesador puede emitir (con emisión DESORDENADA) CUATRO instrucciones por ciclo y tiene TRES unidades de suma/resta (con un retardo de 1 ciclo) y UN multiplicador (con un retardo de 3 ciclos).

Las instrucción (i+1) se puede emitir en el mismo ciclo que la (i)

Usuario Profesores

F



**3** El buffer de reordenamiento (ROB) permite implementar la finalización ordenada en un procesador superescalar

V/F

Usuario Profesores

V



**4** La planificación estática local que realiza el compilador en un procesador VLIW se facilita al aumentar el tamaño de los bloques básicos

V/F

Usuario Profesores

V



**5** Suponga un procesador donde todas las instrucciones pueden predicarse, y los valores de los predicados se asignan con instrucciones con los formatos p1,p2 cmp.cnd r1,r2, o también p1 cmp.cnd r1,r2 donde cnd es la condición que se comprueba entre r1 y r2 (lt, le, gt, ge, eq, ne, respectivamente, menor que, menor o igual que, mayor que, mayor o igual que, igual, distinto). En dicho procesador, la sentencia if ((r1==0) and (r2==0)) then r3=0 else r3=1 puede escribirse sin instrucciones de salto como

V/F

p3 comp.eq r1,r1  
p1,p2 cmp.eq r1,r0  
(p1) p3,p2 cmp.eq r2,r0  
(p3) add r3,r0,r0 // r3=r0+r0  
(p2) add r3, r0,#1 //r3=r0+1

donde r0=0 siempre y los predicados, p1, p2, y p3 pueden tener cualquier valor (0 ó 1) al iniciarse la ejecución del código

Usuario Profesores



**6** En la predicción dinámica de dos bits, el estado del predictor (los bits de historia) no cambia si se acierta la predicción.

BS

SWAD: plataforma de apoyo a la docencia / UGR /

V/F

Usuario Profesores

V



**7** Para implementar un procedimiento de predicción dinámica de saltos es preciso utilizar bits de historia.

V/F

Usuario Profesores

V



**8** Un procesador podría adelantar la instrucción i+1 a la i (i precede a i+1 en el código) si implementa adelantamiento de LOADs a STORES ESPECULATIVO

V/F

(i) sw 0(r5), r2 // M(r5)<-- r2  
(i+1) lw r4, 0(r6) // r4 <--M(r6)

Usuario Profesores

V



**9** Un procesador superescalar de 2 GHz cuyo ROB es capaz de retirar dos instrucciones por ciclo tiene una velocidad pico de 4 GIPS (Giga-instrucciones por segundo)

V/F

Usuario Profesores

V



