

Arquitectura e Ingeniería de Computadores. Examen Parcial (Problemas). 9/02/2006

- 1) Sea un DLX segmentado en cinco etapas, sin planificación dinámica, y con las siguientes características:
- Un dato se puede escribir y leer de un registro en el mismo ciclo de reloj.
 - Existe anticipación de operandos (forwarding)
 - Todas las dependencias se detectan y se resuelven en la etapa DE.
 - Los saltos se resuelven en la etapa DE y se predicen estáticamente como no tomados
 - La etapa de ejecución es multiciclo y, además de la habitual ALU de operaciones enteras, existen las siguientes unidades funcionales en punto flotante (tabla izquierda):

UNIDAD	Cantidad	Latencia	Segmentada
Sumador/ restador	1	2	Si
Multiplica dor	1	4	Si
Divisor	1	7	No

Esta máquina ejecuta el siguiente programa:

```

loop: ld      f2, 0(r1)
      divd    f4, f4, f2
      ld      f2, 8(r1)
      ld      f6, 16(r1)
      divd    f8, f2, f6
      addd    f12, f4, f6
      sd      16(r1), f8
      addi    r1, r1, #32
      sgti    r5, r1, done
      sd      24(r1), f12
      beqz    r5, loop
    
```

- Construya el diagrama de tiempo (instrucciones – ciclos de reloj) para la 1ª iteración del bucle, marcando sobre él todos los riesgos que se presenten y los cortocircuitos que se apliquen. Explique detalladamente, en cada caso, la causa de cualquier posible riesgo y cualquier posible parada. (1,5 puntos)
- Suponiendo que el bucle se ejecuta 1000 veces, ¿cuál es el valor del CPI? (0,5 puntos)

2) Supongamos que disponemos de un procesador dotado de planificación dinámica de instrucciones según el método de Tomasulo. Se dispone de las unidades funcionales que se muestran en la tabla siguiente (tabla derecha) y se dispone de un Bus de Datos Común. El procesador ejecuta el programa descrito (tabla izquierda):

Instrucción	Issue Emisión	Ejecución	Escribe resultado
Mul f2, f3, f4			
Add f6, f2, f4			
Mul f7, f2, f6			
Div f8, f3, f7			
Add f3, f5, f6			
Add f6, f9, f3			
Mul f3, f2, f7			
Add f10, f6, f7			

UF	Cantidad	Latencia	Segmentación
FP ADD/SUB	2	2/2	No
FP MUL/DIV	2	5/15	No

Muestre la evolución de la tabla de estado de instrucciones para este programa, siguiendo el algoritmo de Tomasulo (considere que cada unidad funcional dispone de una estación de reserva). En cada casilla de la tabla indique en qué ciclos de reloj comienza y finaliza cada fase de cada instrucción. (1,5 puntos)

3) Disponemos de un computador de 32 bits con una memoria direccionable en bytes y dotado de una cache de 512 bytes. Cada bloque es de 16 bytes.

- Si la cache es de organización directa, indique para la siguiente secuencia de referencias a memoria si se trata de un acierto o un fallo: 0xA01, 0xB0F, 0x77A, 0x60F, 0xA70, 0xB01, 0xA7A, 0xA0B, 0x67A, 0xB7A, 0x071, 0x67F. Indique además qué fallos producen reemplazamiento. (0,75 puntos)
- Si la cache es asociativa por conjuntos con dos vías, con reemplazamiento LRU, indique para la misma secuencia de referencias del apartado anterior si se trata de un acierto o un fallo. Indique además qué fallos producen reemplazamiento. (0,75 puntos)

SOLUCIONES:

Problema 1

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
ld f2, 0(r1)	IF	DE	EX	ME	WB																				
divd f4, f4, f2		IF	DE ¹	DE	D1	D2	D3	D4	D5	D6	D7	ME	WB												
ld f2, 8(r1)			IF ²	IF	DE	EX	ME	WB																	
ld f6, 16(r1)					IF	DE	EX	ME	WB																
divd f8, f2, f6						IF	DE ^{2*}	DE ^{2*}	DE ^{2*}	DE ^{2*}	DE	D1	D2	D3	D4	D5	D6	D7	ME	WB					
add f12, f4, f6							IF ²	IF ²	IF ²	IF ²	IF	DE	A1	A2	ME	WB									
sd 16(r1), f8												IF	DE ¹	DE ¹	DE ¹	DE ¹	DE ¹	DE	EX	ME	WB				
addi r1, r1, #32													IF ²	IF ²	IF ²	IF ²	IF ²	IF	DE	EX	ME	WB			
sgti r5, r1, done																			IF	DE	EX	ME	WB		
sd 24(r1), f12																			IF	DE	EX	ME	WB		
beqz r5, loop																				IF	DE				
ld f2, 0(r1)																					IF ³	IF	DE	...	

XX¹ Parada por riesgo LDE

XX³ Parada por dependencia de control

XX² Parada por riesgo estructural (etapa sigte ocupada)

(*) La segunda divd no puede empezar hasta fin de la 1ª (UF no segmentada)

b) El primer ld de la segunda iteración hace la búsqueda efectiva en el ciclo 23.

Por tanto tenemos 22 ciclos por iteración.

En 22 ciclos se ejecutan 11 instrucciones => CPI = 22/11 = 2

Problema 2

La siguiente es una solución válida:

Instrucción	Issue Emisión	Ejecución	Escribe resultado
Mul f2, f3, f4	1	2-6	7
Add f6, f2, f4	2	8-9 (*)	10
Mul f7, f2, f6	3	11-15 (*)	16
Div f8, f3, f7	8 (\$)	17-31 (*)	32
Add f3, f5, f6	9	11-12 (*)	13
Add f6, f9, f3	11 (\$)	14-16 (*)	17 (%)
Mul f3, f2, f7	17 (\$)	18-22	23
Add f10, f6, f7	18	19-20	21

(*) : Dependencia LDE.

(\$): No se puede lanzar hasta disponer de una ER libre apta para la operación.

(%) Sólo tenemos un CDB.

Solución alternativa:

Instrucción	Issue Emisión	Ejecución	Escribe resultado
Mul f2, f3, f4	1	2-6	7
Add f6, f2, f4	2	7-8 (*)	9
Mul f7, f2, f6	3	9-13 (*)	14
Div f8, f3, f7	8 (\$)	14-28 (*)	29
Add f3, f5, f6	9	10-11	12
Add f6, f9, f3	10	12-14 (*)	15(%)
Mul f3, f2, f7	15 (\$)	16-20	21
Add f10, f6, f7	16	17-18	19

Problema 3

a) El número de bloques de la cache es 32 ($512/16$). Por tanto el byte dentro del bloque vendrá dado por los bits 3-0, mientras que el bloque vendrá dado por los bits 8-4.

0xA01 → Fallo
0xB0F → Fallo
0x77A → Fallo
0x60F → Fallo con reemplazo
0xA70 → Fallo
0xB01 → Acierto
0xA7A → Acierto
0xA0B → Fallo con reemplazo
0x67A → Fallo con reemplazo
0xB7A → Fallo con reemplazo
0x071 → Fallo con reemplazo
0x67F → Fallo con reemplazo

b) Tenemos 16 conjuntos. El byte dentro del bloque vendrá dado por los bits 3-0, mientras que el conjunto vendrá dado por los bits 7-4.

0xA01 → Fallo
0xB0F → Fallo
0x77A → Fallo
0x60F → Fallo con reemplazo
0xA70 → Fallo
0xB01 → Acierto
0xA7A → Acierto
0xA0B → Fallo con reemplazo
0x67A → Fallo con reemplazo
0xB7A → Fallo con reemplazo
0x071 → Fallo con reemplazo
0x67F → Fallo con reemplazo