

## Arquitectura e Ingeniería de Computadores. Examen Parcial (Problemas). 8/02/2007

**1)** En un DLX con segmentación ejecutamos el siguiente fragmento de código:

```
L0: SUBI r5,r1,#1
    LD F4,0(r5)
    ADDD F4,F4,F6
    LD F6,0(r1)
    ADDD F8,F6,F4
    ADDI r5,r5,#1
    BNEZ r5,L1
    ADDD F0,F2,F4
L1: MULD F4,F0,F2
    LD F6,0(r1)
    ADDD F8,F4,F6
    SD F8,0(r1)
    ADDD F0,F8,F6
    ADDI r5,r5,#1
    BEQZ r5,L0
end
```

Dibujar el diagrama instrucción-tiempo e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas. Se supone que:

- \* Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- \* Se dispone de lógica de cortocircuito (forwarding).
- \* Los saltos se resuelven en la etapa de decodificación (escriben el PC) y se espera a que se resuelva antes de lanzar la siguiente instrucción.
- \* El registro r1 contiene inicialmente el valor "2".
- \* Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni a la de escritura en el banco de registros.
- \* No existe realimentación combinacional de la etapa EX a DE
- \* Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	3	No
FP MUL	1	6	No
Int ALU	1	1	No

**PUNTACIÓN: 2 puntos**

**2)** 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:

UF	Cantidad	Latencia	Segmentación
FP ADD/SUB	2	2/2	No
FP MUL/DIV	2	4/10	No

Suponer que en un mismo ciclo no se puede escribir un dato en el bus común e iniciar la ejecución de una operación que estuviera en espera de ese dato, y que existe una estación de reserva por cada unidad funcional.

Para la secuencia de instrucciones de la tabla siguiente, indica en qué ciclo (o ciclos) de reloj se realiza cada una de sus fases. Indica qué tipos de riesgos se producen.

Instrucción	Issue Emisión	Ejecución	Escribe resultado
MULD F0, F2, F4			
ADDD F6, F0, F4			
MULD F8, F6, F0			
DIVD F10, F2, F8			
ADDD F0, F6, F4			
ADDD F6, F0, F10			
MULD F0, F2, F8			
ADDD F12, F8, F6			

**PUNTUACIÓN: 1 punto**

3) a) Disponemos de una memoria principal de 64 KB direccionable en bytes. Consideremos una memoria caché de 4KB con organización directa, líneas de 256 bytes y que realiza precarga en todos los casos (tanto que haya fallo como acierto se trae el bloque siguiente al que se referencia).

Suponiendo que tenemos la siguiente secuencia de accesos a memoria principal:

0B52h, 2C6Ah, 2901h, 2A26h, 08AAh, 2B10h, 1900h, 2DFFh mostrar el contenido del directorio caché, según el modelo, así como el número de fallos y de precargas que se producen. Indicar con el símbolo "\*" un fallo, con "+" un acierto y con "-" una precarga.

	0B52h	2C6Ah	2901h	2A26h	08AAh	2B10h	1900h	2DFFh
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

b) Supongamos ahora una caché totalmente asociativa compuesta por 4 líneas y un programa compuesto por un bucle que se ejecuta 100 veces y solicita las líneas A-B-C-D-E de memoria principal en cada iteración (cada letra hace referencia a una dirección distinta de memoria principal), siguiendo la secuencia A, B, D, C, A, E.. Calcular el número de fallos que se producen después de las 100 iteraciones si utilizamos el algoritmo de reemplazo LRU y suponemos la caché inicialmente vacía. Calcular lo mismo si el algoritmo utilizado ahora es el LIFO.

**PUNTUACIÓN: 1 punto cada apartado**

## SOLUCIÓN

1)

	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	26	27	28	29		
SUBI r5,r1,#1	IF	ID	EX	M	WB																										
LD F4,0(r5)		IF	ID	EX	M	WB																									
ADDD F4,F4,F6			IF	ID	S1	S2	S3	M	WB																						
LD F6,0(r1)			IF	ID	EX	M	WB																								
ADDD F8,F6,F4					IF	ID	S1	S2	S3	M	WB																				
ADDI r5,r5,#1						IF	IF	ID	EX	M	WB																				
BNEZ r5,L1								IF	ID	EX	M	WB																			
ADDD F0,F2,F4																															
MULD F4,F0,F2										IF	ID	M1	M2	M3	M4	M5	M6	M	WB												
Ld F6,0(r1)												IF	ID	EX	M	WB															
ADDD F8,F4,F6												IF	ID	ID	ID	ID	ID	S1	S2	S3	M	WB									
SD F8,0(r1)													IF	IF	IF	IF	IF	ID	ID	ID	EX	M	WB								
ADDD F0,F8,F6																		IF	IF	IF	ID	S1	S2	S3	M	WB					
ADDI r5,r5,#1																				IF	ID	EX	M	WB							
BEQZ r5,L0																					IF	ID	ID	EX	M	WB					

XX: Parada por riesgo LDE

XX: Parada por riesgo estructural (siguiente etapa ocupada)

XX: Parada por riesgo de control / LDE

$$CPI = 29/14 = 2.07$$

## SOLUCIÓN

2)

Instrucción	Issue Emisión	Ejecución	Escribe resultado
MULD F0, F2, F4	1	2-5	6
ADDD F6, F0, F4	2	7-8	9
MULD F8, F6, F0	3	10-13	14
DIVD F10, F2, F8	7	15-24	25
ADDD F0, F6, F4	8	10-11	12
ADDD F6, F0, F10	10	26-27	28
MULD F0, F2, F8	15	16-19	20
ADDD F12, F8, F6	16	29-30	31

## SOLUCIÓN

3)

a)

Como el tamaño de la memoria principal son 64 Kb necesitamos 16 bits para construir una dirección. La caché tiene 4 Kb de tamaño, y como cada línea tiene 256 bytes tenemos un total de 16 bloques. Por tanto necesitamos 8 bits para especificar el byte dentro del bloque y 4 bits para especificar la línea de cache. Por tanto el campo etiqueta tendrá  $16 - (4 + 8) = 4$  bits.

	0B52h	2C6Ah	2901h	2A26h	08AAh	2B10h	1900h	2DFFh
0								
1								
2								
3								
4								
5								
6								
7								
8					0*	0	0	0
9			2*	2	0-	0	1*	1
10			2-	2+	2	2	1-	1
11	0*	0	0	2-	2	2+	2	2
12	0-	2*	2	2	2	2-	2	2
13		2-	2	2	2	2	2	2+
14								2-
15								

b)

LRU

A	B	D	C	A	E	A	B	D	C	A	E
A	A	A	A	B	D	D	C	E	A	B	D
	B	B	B	D	C	C	E	A	B	D	C
		D	D	C	A	E	A	B	D	C	A
			C	A	E	A	B	D	C	A	E
F	F	F	F	A	F	A	F	F	F	A	F

5 fallos + 4x99 = 401 fallos

LIFO

A	B	D	C	A	E	A	B	D	C	A	E
A	A	A	A	A	A	A	A	A	A	A	A
	B	B	B	B	B	B	B	B	B	B	B
		D	D	D	D	D	D	D	D	D	D
			C	C	E	E	E	E	C	C	E
F	F	F	F	A	F	A	A	A	F	A	F

5 fallos + 2x99 = 203 fallos