

Arquitectura e Ingeniería de Computadores. Examen Parcial (Problemas). 19/02/2008

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

```
L0: ADDD F2,F4,F0
    SD F2,0(r1)
    DIVD F4,F4,F0
    SUBI r1,r1,#1
    ADDD F4,F0,F2
    MULD F2,F2,F4
    BEQZ r1,L2
L1: LD F2,0(r1)
L2: LD F4,0(r1)
    ADDD F0,F2,F4
    DIVD F4,F2,F0
    ADDD F8,F6,F8
    ADDD F4,F0,F6
    ADDI r1,r1,#1
    ADDI r2,r2,#1
    BNEZ r1,L3
    ADDD F0,F0,F2
L3: ADDD F2,F0,F4
    end
```

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.
- * La detección de todo tipo de riesgos y generación de paradas se realiza en la etapa de decodificación.
- * En el caso de los riesgos WAW se produce una parada hasta que la instrucción ya lanzada entre en la etapa de memoria.
- * Los registros r1 y r2 contienen inicialmente el valor unidad.
- * Se dispone de un predictor de saltos de 1 bit del tipo BTB, al que se accede en la etapa IF, obteniendo la respuesta al final de dicha fase. El estado inicial del predictor es "salto no tomado".
- * Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- * Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	2	No
FP MUL	1	3	No
FP DIV	1	5	No
Int ALU	1	1	No

Dibujar el diagrama instrucción-tiempo e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas. Calcular los CPI (ciclos por instrucción) resultantes de la ejecución del código.

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	3/3	No
FP MUL/DIV	2	6/12	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
DIVD F2, F2, F6			
ADDD F4, F6, F4			
MULD F8, F2, F4			
DIVD F0, F6, F4			
ADDD F2, F4, F0			
ADDD F8, F8, F10			
MULD F0, F2, F8			
SUBD F12, F2, F4			

PUNTUACIÓN: 1 punto

3) Considerar un computador con una memoria principal de 64K bytes, direccionable en bytes, al que se dota de una memoria cache de 2K bytes., con líneas de 256 bytes, y prebúsqueda bajo fallo (en caso de fallo se trae el bloque que lo provoca y el siguiente).

Sea la secuencia de acceso a memoria principal dada por las direcciones que aparecen en la tabla.

Mostrar la evolución del directorio cache indicando los fallos y las prebúsquedas que se producen para:

a) una organización directa

	Dirección					
Marco de bloque	3345h	14BFh	1584h	4021h	55ABh	41F1h
0						
1						
2						
3						
4						
5						
6						
7						

b) una organización asociativa por conjuntos de 2 vías, con algoritmo de reemplazo LRU

	Dirección					
Nº Conjunto	3345h	14BFh	1584h	4021h	55ABh	41F1h
0						
1						
2						
3						

Representar la evolución de la cache en las tablas anteriores. Cada columna representa el directorio cache para cada una de las referencias a memoria, excepto la primera columna que indica el marco de bloque o el conjunto. En cada casilla indicar el valor de la información de directorio en **binario** y un símbolo indicando fallo, prebúsqueda y acierto. El símbolo “*” indica fallo, “-” prebúsqueda y “+” acierto.

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	30	31	32	33	34	
ADDD F2,F4,F0	I F	ID	A1	A2 ↘	M	WB																													
SD 0(r1),F2		IF	ID	ID	EX	M	WB																												
DIVD F4,F4,F0			IF	IF	ID	D1	D2	D3	D4	D5	M	WB																							
SUBI r1,r1,#1					IF	ID	EX	M	WB																										
ADDD F4,F0,F2						IF	ID	ID	ID	ID	A1	A2 ↘	M	WB																					
MULD F2,F2,F4							IF	IF	IF	IF	ID	ID	M1	M2	M3	M	WB																		
BEQZ r1,L2											IF	IF	ID	EX	M	WB																			
LD F2,0(r1)													IF																						
LD F4,0(r1)														IF	ID	EX	M	WB																	
ADDD F0,F2,F4															IF	ID	A1	A2 ↘	M	WB															
DIVD F4,F2,F0																IF	IF	ID	ID	D1	D2	D3	D4	D5	M	WB									
ADDD F8,F6,F8																		IF	IF	ID	A1	A2	M	WB											
ADDD F4,F0,F6																			IF	ID	ID	ID	ID	A1	A2	M	WB								
ADDI r1,r1,#1																					IF	IF	IF	IF	ID	ID	EX	M	WB						
ADDI r2,r2,#1																									IF	IF	ID	EX	M	WB					
BNEZ r1,L3																											IF	ID							
ADDD F0,F0,F2																												IF							
ADDD F2,F0,F4																													IF	ID	A1	A2	M	WB	

XX: Parada por riesgo LDE
XX: Parada por riesgo estructural (siguiente etapa ocupada)

XX: Riesgo EDE
XX: Riesgo estructural: dos instrucciones intentan acceder simultáneamente a memoria

$CPI = 34/16 = 2.12$

2)

Instrucción	Issue Emisión	Ejecución	Escribe resultado
DIVD F2, F2, F6	1	2-13	14
ADDD F4, F6, F4	2	3-5	6
MULD F8, F2, F4	3	15-20(*)	21
DIVD F0, F6, F4	15(\$)	16-27	28
ADDD F2, F4, F0	16	29-31(*)	32
ADDD F8, F8, F10	17	22-24(*)	25
MULD F0, F2, F8	22(\$)	33-38(*)	39
SUBD F12, F2, F4	26(\$)	33-35(*)	36

(*): LDE

(\$): Riesgo estructural (no suficientes estaciones de reserva).

3)

- a) Como la memoria principal es de $64k = 2^{16}$ --> necesitamos 16 bits para construir la dirección. Como las líneas son de 256 bytes --> 8 bits para determinar la palabra dentro de la línea. Tenemos en total 8 líneas --> 3 bits

Marco de bloque	3345h	14BFh	1584h	4021h	55ABh	41F1h
0				01000*	01000	01000
1				01000-	01000	01000+
2						
3	00110*	00110	00110	00110	00110	00110
4	00110-	00010*	00010	00010	00010	00010
5		00010-	00010+	00010	01010*	01010
6					01010-	01010
7						

- b) Necesitamos 2 bits para determinar el conjunto:

Nº Conjunto	3345h	14BFh	1584h	4021h	55ABh	41F1h
0	001101-	001101	001101	010000*	010000	010000
		000101*	000101	000101	000101	000101
1		000101-	000101+	000101	010101*	010101
				010000-	010000	010000+
2					010101-	010101
3	001100*	001100	001100	001100	001100	001100