

Nombre-----Grupo-----

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

```

SUBI r1,r0,#1
loop: MULD F0,F2,F4
      LD F2, 0(r2)
      DIVD F6,F6,F2
      SD 0(r2), F6
      ADDD F6,F0,F2
      ADDI r1,r1,#1
      ADDD F4,F6,F0
      DIVD F0,F6,F8
      LD F2, 0(r1)
      DIVD F8,F4,F4
      ADDD F8,F0,F6
      BNEZ r1,loop
      SUBD F8,F8,F2
end:  ADDD F6,F0,F2
    
```

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 tienen tratamiento de instrucción entera y se resuelven en la etapa de ejecución (EX).
- La detección de todo tipo de riesgos (estructurales, LDE y EDE) y generación de paradas se realiza en la etapa de decodificación.
- 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 ADDD	1	2	Sí
FP SUBD	1	2	Sí
FP MULD	1	3	Sí
FP DIVD	1	4	No
INT ALU	1	1	No

- A) Representar el diagrama instrucción-tiempo para la ejecución del código e indicar mediante flechas sobre el diagrama los cortocircuitos realizados. Indicar claramente las paradas y sus causas. A la vista del diagrama obtenido, indicar los ciclos por instrucción (CPI) resultantes de la ejecución completa del código. **(3.5 pts)**
- B) ¿Cómo se vería afectado el valor del CPI del apartado anterior si el estado inicial del predictor fuese ahora “salto tomado”? ¿Cuál sería su nuevo valor? Representar el nuevo diagrama instrucción-tiempo únicamente para la resolución del salto (instrucción de salto y las que se ejecutan tras éste). **(1.5 pts)**

2) Considerar una memoria principal de 32M bytes direccionable en bytes a la que se dota de una memoria cache con las siguientes características:

- Tamaño de 2K bytes
- Líneas de 256 bytes,
- Prebúsqueda bajo fallo (en caso de fallo se trae el bloque que lo provoca y el siguiente)
- Algoritmo de reemplazamiento FIFO.

Sea la secuencia de acceso a memoria principal dada por las siguientes direcciones: 23FAh, 14A2h, 3F02h, 40B1h, 5572h, 23AAh. Mostrar la evolución del directorio cache indicando los fallos (F), las prebúsquedas (P) y los aciertos (A) que se producen suponiendo:

- a) emplazamiento directo
- b) emplazamiento asociativo por conjuntos de 2 vías

Representar la evolución de la cache en las siguientes tablas. 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 y señalar los aciertos, fallos y prebúsquedas.

(5 pts)

a)

Dirección línea	23FAh	14A2h	3F02h	40B1h	5572h	23AAh
0						
1						
2						
3						
4						
5						
6						
7						

b)

Dirección conjunto	23FAh	14A2h	3F02h	40B1h	5572h	23AAh
0						
1						
2						
3						

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
SUBI r1,r0,#1	IF	ID	EX	M	WB																									
MULD F0,F2,F4		IF	ID	M1	M2	M3	M	WB																						
LD F2, 0(r2)			IF	ID	EX	M	WB																							
DIVD F6,F6,F2				IF	ID	ID	D1	D2	D3	D4	M	WB																		
SD 0(r2), F6					IF	IF	ID	ID	ID	ID	EX	M	WB																	
ADDD F6,F0,F2							IF	IF	IF	IF	ID	A1	A2	M	WB															
ADDI r1,r1,#1											IF	ID	ID	EX	M	WB														
ADDD F4,F6,F0												IF	IF	ID	A1	A2	M	WB												
DIVD F0,F6,F8														IF	ID	D1	D2	D3	D4	M	WB									
LD F2, 0(r1)															IF	ID	EX	M	WB											
DIVD F8,F4,F4																IF	ID	ID	ID	D1	D2	D3	D4	M	WB					
ADDD F8,F0,F6																	IF	IF	IF	ID	ID	ID	A1	A2	M	WB				
BNEZ r1,loop																				IF	IF	IF	ID	ID	EX	M	WB			
SUBD F8,F8,F2																						IF	IF	ID	S1	S2	M	WB		
ADDD F6,F0,F2																									IF	ID	A1	A2	M	WB

XX: Parada por riesgo LDE

XX: Parada por riesgo estructural (siguiente etapa ocupada)

XX: Riesgo estructural: unidad no segmentada

XX: Riesgo EDE

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

A) La ejecución completa del código implica 30 ciclos para ejecutar 15 instrucciones, por tanto $CPI=30/15=2$

B) Si el predictor inicialmente estuviese a "salto tomado" la secuencia de instrucciones ejecutadas a partir del salto sería:

BNEZ r1, loop IF IF iF ID ID EX M WB
 MULF F0,F2,F4 IF IF ID XXXXXXXXXXXX
 LD F2, 0(r2) IF XXXXXXXXXXXX
 SUBD F8,F8,F2 IF ID S1 S2 M WB
 ADDD F6,F0,F2 IF ID A1 A2 M WB

Tendríamos 2 ciclos más como consecuencia de los 2 ciclos que tarda el salto en resolverse tras haberse hecho efectiva la predicción errónea, por tanto CPI=32

2)

Solución:

Como la memoria principal es de 32 Mb = 2^{25} --> necesitamos 25 bits para la dirección.

Memoria cache de 2kbytes = 2^{11} bytes

Como las líneas son de 256 bytes --> 8 bits para determinar byte de la línea.

Total de líneas es $2^{11}/2^8=8$ líneas --> 3 bits para direccionar la línea:

Cache tag	Cache index	Byte select
24:11	10:8	7:0

En la siguiente tabla aparecen las direcciones en hexadecimal y en binario:

Dirección línea	2-3-FA	1-4-A2	3-F-02	4-0-B1	5-5-72	2-3-AA
0	00100-011	00010-100	00111-111	01000-000	01010-101	00100-011
1			01000 (P)	01000 (A)	01000	01000
2						
3	00100 (F)	00100	00100	00100	00100	00100 (A)
4	00100 (P)	00010 (F)	00010	00010	00010	00010
5		00010 (P)	00010	00010	01010 (F)	01010
6					01010 (P)	01010
7			00111 (F)	00111	00111	00111

b) Asociativa de 2 vías

Como teníamos un total de 8 líneas en la memoria cache y ahora es asociativa de 2 vías, el total de conjuntos es $8/2 = 4$, luego ahora necesitamos 2 bits para determinar el conjunto. La dirección queda:

Cache tag	Cache index	Byte select
24:10	9:8	7:0

Dirección conjunto	2-3-FA 001000-11	1-4-A2 000101-00	3-F-02 001111-11	4-0-B1 010000-00	5-5-72 010101-01	2-3-AA 001000-11
0	001001 (P)	001001	010000 (P)(F)	010000 (A)	010000	010000
		000101 (F)	000101	000101	000101	000101
1		000101 (P)	000101	000101	000101	000101
					010101 (F)	010101
2					010101 (P)	010101
3	001000 (F)	001000	001000	001000	001000	001000 (A)
			001111 (F)	001111	001111	001111

