

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

### **Ejercicio 1 (3 puntos)**

a) Se dispone de un procesador segmentado con un juego de instrucciones de tipo entero basadas en el procesador DLX. El cálculo de la condición de salto y dirección destino en caso de instrucción de bifurcación se realiza en la etapa de ejecución (EX). Dicho procesador posee un predictor del tipo BTB de 1 bit, que se accede durante la fase IF, obteniendo la respuesta al final de dicha fase. Dada la siguiente secuencia de código, y suponiendo que el buffer del predictor contiene el estado “predicción no salta” para la instrucción de salto, mostrar las fases de la instrucción de salto y de las tres instrucciones que se ejecutan inmediatamente después del mismo en cada una de las 3 primeras iteraciones. **(1.5 puntos)**

```
ADDI r1,r0,#3
bucle: SUB r2,r2,r3
        ADD r5,r1,r5
        SUBI r1,r1,#1
        BNEZ r1,bucle
        OR r6,r6,r2
        AND r2,r5,r2
        XOR r3,r3,r2
```

b) Un programa P consta de dos tipos de instrucciones: enteras (que suponen el 60% del total de instrucciones) y en punto flotante (el 40% restante). El CPI medio del programa al ejecutarse en cierto procesador es 6, mientras que el CPI para las instrucciones enteras es 2. Si introducimos un coprocesador matemático que reduce el tiempo de ejecución de las instrucciones en punto flotante a una cuarta parte, ¿cuál será el speedup global del sistema? **(1.5 puntos)**

### **Ejercicio 2 (2 puntos)**

La memoria principal de un sistema es de 1 megabyte y direccionable en bytes, a la que se dota de una memoria cache con las siguientes características:

- $2^5$  bytes
- tamaño de línea de 4 bytes
- prebúsqueda bajo fallo
- algoritmo de reemplazamiento FIFO
- grado de asociatividad de 2.

Calcular:

- a) el cronograma de la cache de etiquetas al ejecutar el siguiente código indicando los fallos, aciertos y prebúsquedas
- b) Calcular la tasa de fallos
- c) Calcular la tasa de aciertos de prebúsqueda

La secuencia de código es la siguiente:

Nº Referencia	Dirección
1	0xA0
2	0xCE
3	0xD1
4	0xCF
5	0xA1
6	0x32
7	0xCC
8	0x30
9	0xA1
10	0xCE

# Solución

## 1.

a)

1ª Iteración (r1=2); Predictor = salto no tomado → Fallo → predictor pasa a salto tomado

BNEZ r1, bucle	IF	ID	EX	M	WB
OR r6,r6,r2		IF	ID	X	X....
AND r2,r5,r2		IF	X	X.....	
SUB r2,r2,r3		IF	ID	EX.....	

2ª Iteración (r1=1); Predictor = salto tomado → Acierto → predictor sigue a salto tomado

BNEZ r1, bucle	IF	ID	EX	M	WB
SUB r2,r2,r3		IF	ID	EX	M.....
ADD r5,r1,r5		IF	ID	EX.....	
SUBI r1,r1,#		IF	ID	EX.....	

3ª Iteración (r1=0); Predictor = salto tomado → Fallo → predictor pasa a salto no tomado

BNEZ r1, bucle	IF	ID	EX	M	WB
SUB r2,r2,r3		IF	ID	X	X.....
ADD r5,r1,r5		IF	X	X.....	
OR r6,r6,r2		IF	ID	EX.....	

b)

$CPI_{programa} = CPI_{int} * F_{int} + CPI_{pf} * F_{pf} \rightarrow 6 = 2 * 0.6 + CPI_{pf} * 0.4$  ; De aquí obtenemos que  $CPI_{pf} = 12$

La fracción de mejora, es decir el tiempo que se están ejecutando instrucciones en punto flotante respecto al tiempo de ejecución total será  $(0.4 * 12) / (0.6 * 2 + 0.4 * 12) = 0.8$

De acuerdo con la ley de Amdhal, el speedup global del sistema será  $1 / (1 - 0.8 + 0.8 / 4) = 1 / 0.4 = 2.5$

## 2.

Primero vamos a calcular el formato de la dirección de memoria caché:

Memoria principal=1 megabyte= $2^{20}$  byte=> la dirección tendrá 20 bits

Número de bytes por línea es 4=> se necesitan 2 bits para referenciar a cada uno de los bytes

Número de líneas de la memoria cache =  $2^5 \text{byte} / 2^2 = 2^3$  líneas

Número de conjuntos de la memoria cache =  $2^3/2^1$  conjuntos =  $2^2$  => se necesitan dos bits para referenciar a cada uno de los conjuntos de la cache

La dirección de memoria cache queda:

16 bits	2bits	2 bits
Cache tag	Cache index	Byte select

En función de esto pasamos las direcciones que están en hexadecimal a binario (sólo trabajamos con los bits significativos de la dirección):

Cache tag	Cache index	Byte select
1010	00	00
1100	11	10
1101	00	01
1100	11	11
1010	00	01
0011	00	10
1100	11	00
0011	00	00
1010	00	01
1100	11	10

CICLO	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
INST	1010-00-00	1100-11-10	1101-00-01	1100-11-11	1010-00-01	0011-00-10	1100-11-00	0011-00-00	1010-00-01	1100-11-10
C0	<b>1010 F</b>	1010	1010	1010	<b>1010 A</b>	<b>0011 F</b>	0011	<b>0011 A</b>	0011	0011
		<b>1101 P</b>	<b>1101 A</b>	1101	1101	1101	1101	1101	<b>1010 F</b>	1010
C1	<b>1010 P</b>	1010	1010	1010	1010	1010	1010	1010	<b>1010 P*</b>	1010
						<b>0011 P</b>	0011	0011	0011	0011
C2										
C3		<b>1100 F</b>	1100	<b>1100 A</b>	1100	1100	<b>1100 A</b>	1100	1100	<b>1100 A</b>

TASA DE FALLOS EN LA EJECUCIÓN DEL CÓDIGO ES (4 FALLOS)  $4/10 = 0,4$   
TASA DE ACIERTOS DEBIDA A LA PREBUSQUEDA  $\frac{1}{4}$

4 son las prebúsquedas totales, aunque en el caso de P\* el bloque ya se encontraba en memoria

Solo se acierta una vez, el caso de P\*, en realidad no es un acierto puesto que el dato no se está solicitando desde la memoria