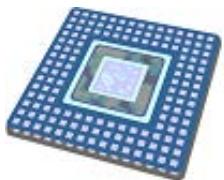




Escuela Técnica Superior de
Ingeniería Informática

ARQUITECTURA DE COMPUTADORES

BOLETÍN DE EJERCICIOS RESUELTOS



**Departamento de Arquitectura y
Tecnología de Computadores**

Rubén Bueno Menéndez

BOLETÍN DE EJERCICIOS DEL TEMA 2: EVALUACIÓN DE PRESTACIONES

ARQUITECTURA DE COMPUTADORES, 2º ING. INFORMÁTICA-INGENIERÍA DEL SOFTWARE
DEPT. DE ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES, UNIVERSIDAD DE SEVILLA

1. (Ejercicios 1.4.1, 1.4.2 y 1.4.3 de [PH-2011])¹

Considere dos implementaciones diferentes de un mismo repertorio de instrucciones, con cuatro tipos de instrucciones, A, B, C y D. La frecuencia de reloj y el CPI de cada implementación se muestran en la siguiente tabla.

	Frec. reloj	CPI tipo A	CPI tipo B	CPI tipo C	CPI tipo D
M1	1.5 GHz	1	2	3	4
M2	2 GHz	2	2	2	2

- a) Considere un programa con 10^6 instrucciones repartidas de la siguiente forma: 10% de tipo A, 20% de tipo B, 50% de tipo C y 20% de tipo D. Calcule el tiempo de ejecución del programa en ambas máquinas. ¿Qué implementación es más rápida para este programa?

[Sol.: $t(M1) = 18.67 \times 10^{-4}$ s $t(M2) = 10 \times 10^{-4}$ s Implementación más rápida: M2]

- b) ¿Cuál es el CPI global (medio) de cada implementación?

[Sol.: CPI (M1) = 2.8 CPI (M2) = 2.0]

- c) Determine el número de ciclos de reloj en ambos casos.

[Sol.: N° ciclos (M1) = 28×10^5 N° ciclos (M2) = 20×10^5]

2. (Ejercicios 1.4.4, 1.4.5 y 1.4.6 de [PH-2011])

La siguiente tabla muestra el número y tipo de instrucciones de un programa.

	Aritméticas	Almacenamiento	Carga	Saltos	Total
P1	500	50	100	50	700

- a) Suponiendo que las instrucciones aritméticas necesitan 1 ciclo para ejecutarse, las de carga y almacenamiento 5 ciclos y los saltos 2 ciclos, ¿cuál es el tiempo de ejecución en un procesador a 2 GHz?

[Sol.: t = 675 ns]

- b) Determine el CPI del programa.

[Sol.: CPI = 1.93]

- c) Si el número de instrucciones de carga se reduce a la mitad, ¿cuál es la aceleración y el nuevo CPI?

[Sol.: t = 550 ns aceleración = 1.23 CPI = 1.69]

¹ [PH-2011]: D.A. Patterson, J.L. Hennessy, "Estructura y Diseño de Computadores", Ed. Reverté, 2011.

PANDURA



MOTIVACIÓN



CONCENTRACIÓN



PANDURA



"Sin Pandora,
juegas en
desventaja."



"Memorizo un
50% más en el
mismo tiempo."



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEE ARABICA

Realiza tu diseño sobre el tema de color negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

Evaluación de prestaciones:

1.

a) Siendo $n_{instTotal} = 10^6$ instrucciones, podemos calcular el número de ciclos totales de la siguiente manera: $n_{ciclos} = \sum_{i=0}^4 n_{inst_i} \cdot CPI_i$

Para M1:

$$n_{ciclos} = 10^6 \cdot 0,1 \cdot 1 + 10^6 \cdot 0,2 \cdot 2 + 10^6 \cdot 0,5 \cdot 3 + 10^6 \cdot 0,2 \cdot 4 = 2800000 \text{ ciclos}$$

$$\text{Por tanto, como } t_{CPU} = \frac{n_{ciclos}}{\text{f_reloj}} = \frac{2800000}{1,5 \cdot 10^9} = 0,001866 \approx 18,67 \cdot 10^{-4} \text{ s}$$

Para M2:

$$n_{ciclos} = 10^6 \cdot 0,1 \cdot 2 + 10^6 \cdot 0,2 \cdot 2 + 10^6 \cdot 0,5 \cdot 2 + 10^6 \cdot 0,2 \cdot 2 = 2000000 \text{ ciclos}$$

$$\text{Por tanto, como } t_{CPU} = \frac{n_{ciclos}}{\text{f_reloj}} = \frac{2000000}{2 \cdot 10^9} = 0,001 = 10 \cdot 10^{-4} \text{ s}$$

Como M2 tarda menos tiempo, esta será la implementación más rápida.

b)

Para M1:

$$CPI_{medio} = \frac{n_{ciclos}}{n_{inst}} = \frac{2800000}{10^6} = 2,8$$

Para M2:

$$CPI_{medio} = \frac{n_{ciclos}}{n_{inst}} = \frac{2000000}{10^6} = 2$$

c) Ya lo hemos calculado:

$$\text{Para M1: } n_{ciclos} = 2800000 = 28 \cdot 10^5 \text{ ciclos}$$

$$\text{Para M2: } n_{ciclos} = 2000000 = 20 \cdot 10^5 \text{ ciclos}$$

3.

a) Podemos calcular el número de ciclos totales de la siguiente manera:

$$n_{ciclos} = \sum_{i=0}^3 n_{inst_i} \cdot CPI_i$$

Para P1:

$$n_{ciclos} = 10^6 \cdot 0,5 \cdot 0,75 + 10^6 \cdot 0,4 \cdot 1 + 10^6 \cdot 0,1 \cdot 1,5 = 925000 \text{ ciclos}$$

$$\text{Por tanto, como } t_{CPU} = \frac{n_{ciclos}}{\text{f_reloj}} = \frac{925000}{3 \cdot 10^9} = 0,0003083 = 3,083 \cdot 10^{-4} \text{ s}$$

$$\text{Luego, MIPS} = \frac{n_{inst}}{t_{CPU} \cdot 10^6} = \frac{10^6}{3,083 \cdot 10^{-4} \cdot 10^6} = 3243,24 \text{ millones de inst/s}$$

Para P2:

$$n_{ciclos} = 3 \cdot 10^6 \cdot 0,4 \cdot 1,25 + 3 \cdot 10^6 \cdot 0,4 \cdot 0,7 + 3 \cdot 10^6 \cdot 0,2 \cdot 1,25 = 3040000 \text{ ciclos}$$

$$\text{Por tanto, como } t_{CPU} = \frac{n_{ciclos}}{\text{f_reloj}} = \frac{3040000}{3 \cdot 10^9} = 0,00103 = 1,03 \cdot 10^{-3} \text{ s}$$

$$\text{Luego, MIPS} = \frac{n_{inst}}{t_{CPU} \cdot 10^6} = \frac{3 \cdot 10^6}{1,03 \cdot 10^{-3} \cdot 10^6} = 2912,62 \text{ millones de inst/s}$$

b) Podemos calcular el número de ciclos totales para instrucciones en punto flotante de la siguiente manera: $n_{ciclos_{pf}} = n_{inst_{pf}} \cdot CPI_{pf}$

Para P1:

$$n_{ciclos_{pf}} = 10^6 \cdot 0,4 \cdot 1 = 400000 \text{ ciclos}$$

$$\text{Por tanto, como } t_{CPU_{pf}} = \frac{n_{ciclos_{pf}}}{\text{f_reloj}} = \frac{400000}{3 \cdot 10^9} = 0,000133 = 1,33 \cdot 10^{-4} \text{ s}$$

$$\text{Luego, MFLOPS} = \frac{n_{inst_{pf}}}{t_{CPU_{pf}} \cdot 10^6} = \frac{10^6 \cdot 0,4}{1,33 \cdot 10^{-4} \cdot 10^6} = 3000 \text{ millones de op en pf/s}$$

Para P2:

$$n_{ciclos_{pf}} = 3 \cdot 10^6 \cdot 0,4 \cdot 0,7 = 840000 \text{ ciclos}$$

$$\text{Por tanto, como } t_{CPU_{pf}} = \frac{n_{ciclos_{pf}}}{\text{f_reloj}} = \frac{840000}{3 \cdot 10^9} = 0,00028 = 2,8 \cdot 10^{-4} \text{ s}$$

PANDORA

MOTIVACIÓN



CONCENTRACIÓN



MEMORIA

**PANDORA**

Realiza tu diseño sobre el tema de color negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

*"Sin Pandora,
juegas en
desventaja."*

4.

a) Podemos medir el consumo de la potencia de la siguiente manera:

$$P = C \cdot V^2 \cdot f$$

Si queremos saber cuánto se ha reducido la carga capacitiva entre ambas versiones si P se reduce un 10%, estableceremos la siguiente relación:

$$\frac{P_2}{P_1} = 1 - 0,1 \rightarrow \frac{P_2}{P_1} = 0,9 \text{, por tanto:}$$

$$\frac{C_2 \cdot V_2^2 \cdot f_2}{C_1 \cdot V_1^2 \cdot f_1} = 0,9 \rightarrow \frac{C_2 \cdot 3,3^2 \cdot 1 \cdot 10^9}{C_1 \cdot 5^2 \cdot 0,5 \cdot 10^9} = 0,9 \rightarrow \frac{C_2}{C_1} = 1,103$$

Es decir, no se ha reducido, sino que ha aumentado un 3% la carga capacitiva.

b) Si la carga capacitativa no ha cambiado, entonces no hace falta tenerla en cuenta. Por tanto:

$$\frac{P_2}{P_1} = \frac{C_2 \cdot V_2^2 \cdot f_2}{C_1 \cdot V_1^2 \cdot f_1} = \frac{3,3^2 \cdot 1 \cdot 10^9}{5^2 \cdot 0,5 \cdot 10^9} = 0,8712$$

Es decir, se ha reducido un 12,8%.

c) Estamos ante la situación de que $C_2 = 0,8 \cdot C_1$ y $P_2 = (1 - 0,4)P_1 = 0,6P_1$, por tanto:

$$\frac{P_2}{P_1} = 0,6 \rightarrow \frac{0,8 \cdot C_1 \cdot V_2^2 \cdot f_2}{C_1 \cdot V_1^2 \cdot f_1} = 0,6 \rightarrow \frac{V_2^2 \cdot 0,8 \cdot 1 \cdot 10^9}{5^2 \cdot 0,5 \cdot 10^9} = 0,6 \rightarrow$$

$$\rightarrow V_2 = 3,0618 \text{ V}$$



*"Memorizo un
50% más en el
mismo tiempo."*



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEE ARABICA

CASO PRÁCTICO DEL TEMA 3: ARQUITECTURA DEL REPERTORIO DE INSTRUCCIONES
ARQUITECTURA DE COMPUTADORES, 2º ING. INFORMÁTICA-INTENIERÍA DEL SOFTWARE

CASO PRÁCTICO

Se desea diseñar un procesador de 8 bits que realice 30 operaciones diferentes (el número de instrucciones del procesador dependerá de la estrategia de diseño) y que operan de la forma $A \leftarrow A \text{ op } B$.

Debe admitir 3 modos de direccionamiento: Inmediato, Registro y acceso a Memoria indirecto mediante registro. El operando destino (A) no puede utilizar el modo inmediato. El tamaño del campo dedicado a ambos operandos debe ser idéntico.

La búsqueda de la instrucción supone 2 accesos a memoria.

- Proponga un formato de instrucción lo más simple posible para un juego de instrucciones **ortogonal**¹. ¿De qué tamaño son los registros? En todos los casos, justifique muy brevemente su decisión.
- Proponga un nuevo formato suponiendo que el juego de instrucciones no es ortogonal y que no se permiten operaciones con los dos operandos en memoria. ¿Cuál es el tamaño del código de operación? ¿Cuántas operaciones diferentes podríamos codificar?
- ¿Es el valor inmediato del tamaño de palabra? En caso de que sea menor, ¿Qué podemos hacer para que el juego de instrucciones permita operar con valores del tamaño de palabra?
- En ambos casos, ¿Cuántos registros tendrá el procesador?

SOLUCIÓN

- El procesador es de 8 bits y la búsqueda de la instrucción requiere dos accesos, por lo tanto, el tamaño de la instrucción es de 16 bits.

El procesador es ortogonal y realiza 30 operaciones. Al ser ortogonal, podemos asignar un *opcode* diferente a cada operación y codificar el modo de direccionamiento de cada operando en campos diferentes del *opcode*. Por lo tanto, para codificar 30 operaciones nos bastan 5 bits (*opcode* de 5 bits).

Puesto que las operaciones son de la forma $A \leftarrow A \text{ op } B$, tenemos 2 operandos en el que A actúa como fuente y destino.

Puesto que los modos de direccionamiento son Inmediato, Registro y Memoria mediante registro:

- El operando A no puede emplear direccionamiento inmediato (al ser destino de la operación); entonces, tenemos 2 modos de direccionamiento para el operando A que se codifican con 1 bit.
- El operando B emplea los 3 modos de direccionamiento; por lo tanto, requiere 2 bits para codificar su campo de modo de direccionamiento.

El espacio restante de la instrucción ($16 - 5 - 1 - 2 = 8$ bits) se asigna a los 2 operandos. Exceptuando el direccionamiento inmediato, ambos operandos admiten los mismos modos de direccionamiento, por lo tanto, una posible solución consiste en asignar el mismo tamaño a cada operando: $8 / 2 = 4$ bits por operando. Observe que en esta solución el operando B sólo podrá contener valores inmediatos de tamaño media palabra.

Op_code	Modo dir. Operando A	Operando A	Modo dir. Operando B	Operando B
5	1	4	2	4

¹ Se dice que un juego de instrucciones es **ortogonal** cuando los operandos de todas las instrucciones admiten todos los modos de direccionamiento posibles (es decir, el modo de direccionamiento de los operandos es independiente de la instrucción).

PANDURA

MOTIVACIÓN



CONCENTRACIÓN

**PANDURA**

*"Sin Pandora,
juegas en
desventaja."*



*"Memorizo un
50% más en el
mismo tiempo."*



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Recarga tu energía sobre el fondo de color negro.

ARQ. DE COMPUTADORES

GRADO EN ING. DEL SOFTWARE, 2º CURSO

Arquitectura de Computadores
Rubén Bueno Menéndez

Indique qué opción es la correcta en el margen izquierdo de cada pregunta. Solo se puede seleccionar una opción. Cada respuesta incorrecta resta un tercio de una correcta.

1. Una CPU tiene una cache con 32 líneas que divide las direcciones en los siguientes campos: DESPLAZAMIENTO de 8 bits, ÍNDICE de 4 bits y ETIQUETA de 20 bits. Indique que afirmación es la correcta:

- A. La caché es asociativa por conjuntos de 8 vías.
- B. La caché es asociativa por conjuntos de 4 vías.
- C. La caché es asociativa por conjuntos de 2 vías. **(Correcta)**
- D. La caché es de mapeado directo.

Como el índice es 4, entonces $2^4 = 16$ bloques o conjuntos. Siendo x el número de vías necesario:
 $16 \cdot x = 32 \rightarrow x = 2$ vías.

2. Se tiene una cache de mapeado directo con la siguiente división de direcciones: DESPLAZAMIENTO de 8 bits, ÍNDICE de 4 bits y ETIQUETA de 20 bits. Indique el número y el tipo de fallos que se producen para la siguiente secuencia de referencias a direcciones de memoria: 0xA101, 0xA104, 0xB102, 0xBC03, 0xA1FF. Suponga que la caché está inicialmente vacía.

- A. 4 fallos forzados y 0 fallos por conflicto.
- B. 3 fallos forzados y 0 fallos por conflicto.
- C. 3 fallos forzados y 1 fallo por conflicto. **(Correcta)**
- D. 2 fallos forzados y 1 fallo por conflicto.

Dir. física	Etiqueta	Línea	F/A
0xA101	A	1	Fallo forzoso
0xA104	A	1	Acierto
0xB102	B	1	Fallo forzoso
0xBC03	B	C	Fallo forzoso
0xA1FF	A	1	Fallo por conflicto

3. Un sistema de memoria virtual tiene las siguientes características: direcciones virtuales de V bits, direcciones físicas de F bits, tamaño de página de 2^P bytes y una TLB de mapeado directo con 2^L líneas. Indique cuantos elementos tiene la tabla de páginas.

- A. 2^{F-P-L}
- B. 2^{F-P}
- C. 2^{V-P-L}
- D. 2^{V-P} **(Correcta)**

Sea V los bits de la dirección virtual, P los bits de desplazamiento de página y x los bits del número de página virtual, entonces $x + P = V \rightarrow x = V - P$. Por tanto, habrá 2^{V-P} elementos en la tabla de páginas.

Problema de memoria virtual y caché:

a)

La memoria virtual es de $4\text{ GB} = 2^2 \cdot 2^{30} = 2^{32}$ bytes, por tanto, las direcciones virtuales son de 32 bits.

La memoria principal es de $256\text{ MB} = 2^8 \cdot 2^{20} = 2^{28}$ bytes, por tanto, las direcciones físicas son de 28 bits.

Podemos calcular cuántos bits de desplazamiento de página tenemos tanto para la dirección física como la virtual:

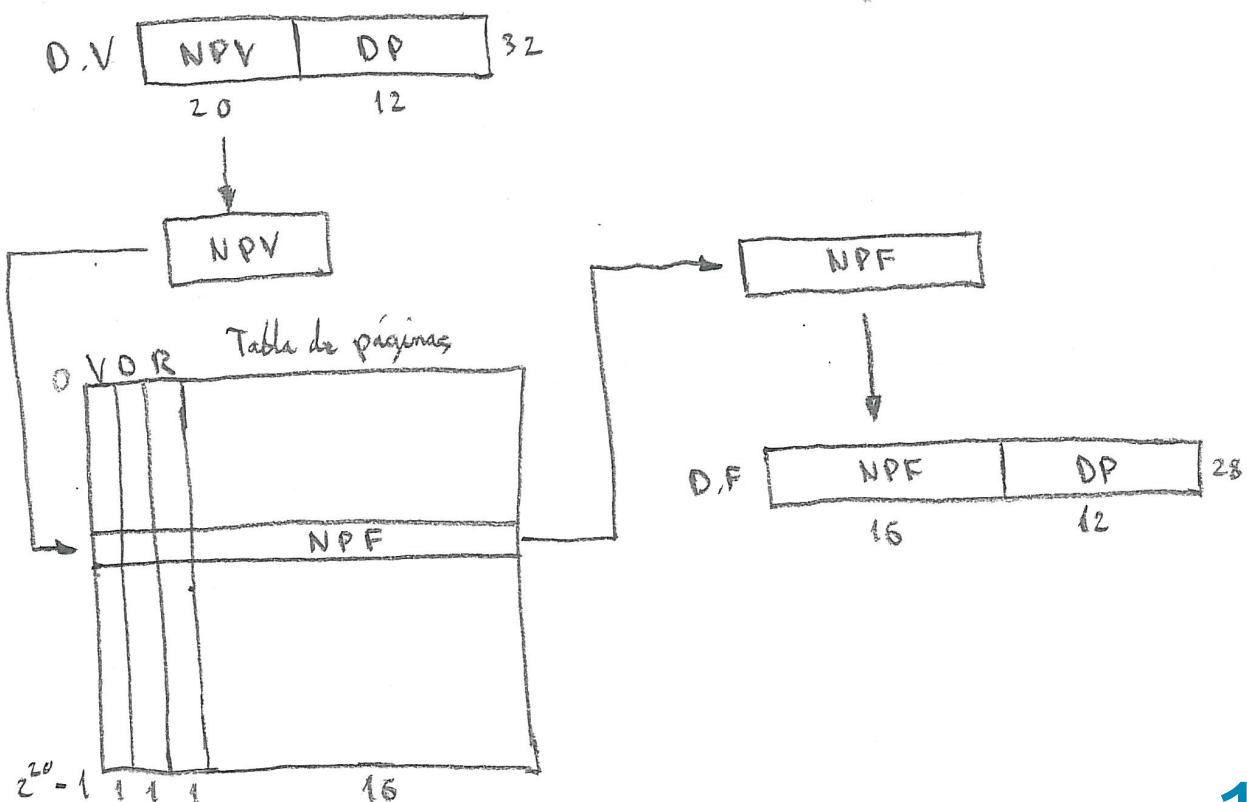
$$\text{Desp. de pági.} = \log_2(4\text{ KB}) = \log_2(2^2 \cdot 2^{10}) = 12 \text{ bits de desplazamiento de pági.}$$

Sabiendo esto podemos calcular el número de página virtual y físico:

$$Nº \text{ pag. virtual} = 32 - 12 = 20 \text{ bits}$$

$$Nº \text{ pag. físico} = 28 - 12 = 16 \text{ bits}$$

Por tanto, la estructura del sistema sería la siguiente:



PANDURA



MOTIVACIÓN



CONCENTRACIÓN



MEMORIA



"Sin Pandora,
juegas en
desventaja."



"Memorizo un
50% más en el
mismo tiempo."

Dale lo mejor a tu cerebro con la fórmula de colores negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

El estado final de la caché sería el siguiente:

V	Etagera	Blanca
L0	1	0x4004 Letras [0..15]
L1	1	0x4004 Letras [0..16]
L2		
L3		

d)

$$\text{Frecuencia de fallos} = \frac{\text{Nº fallos}}{\text{Nº accesos}} = \frac{2}{26} = 0,0769$$

Es decir, ha habido un 7,6% de fallos respecto al total de números de accesos.

e)

Si hubiera sido WT>WTA, entonces cuando se produce un fallo, solo se lleva el dato a la memoria principal y, cuando se produce acierto, se lleva a memoria principal y caché.

Por tanto, se habría producido siempre fallos forzados y entonces la frecuencia de fallos sería del 100%, debido a que nunca se lleva el dato a memoria caché al ser todos los accesos de escritura.



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

BOLETÍN DE EJERCICIOS DEL TEMA 6: PARALELISMO EN UN PROCESADOR

ARQUITECTURA DE COMPUTADORES

DEPT. DE ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES, UNIVERSIDAD DE SEVILLA

EJERCICIO 1. Considere un procesador MIPS que ejecuta el siguiente trozo de código:

```
add $t4,$t0,$t0
addi $t2,$t2,24
SIG: lw $t1,996($t2)
      sgei $t3,$t1,0
      bnez $t3,POS
      sub $t1,$t0,$t1
      addi $t4,$t4,1
      sw $t1, 996($t2)
POS: subi $t2,$t2,4
      bnez $t2,SIG
      sw $t4, 2000($t0)
```

Considere que el procesador MIPS es un procesador segmentado que usa la técnica de anticipación de resultados o bypass y predicción de salto no tomado. Inicialmente la memoria contiene a partir de la dirección 1000 los siguientes valores de 32 bits: 300, -15, 1241, -10, 400, -100, 10, 50, ...

Se pide :

- a) Escriba un trozo de código usando un lenguaje de alto nivel (p.e. Lenguaje C) que realice la misma acción que el trozo de código escrito en el enunciado.

```
negativos = 0 ;
for(i=5 ; i>=0 ; i--) {
    if (x[i]<0) {
        negativos ++ ;
        x[i] = (-1)*x[i] ;
    }
}
```

PANDURA**MOTIVACIÓN****CONCENTRACIÓN****PANDURA**

**"Sin Pandora,
juegas en
desventaja."**



**"Memorizo un
50% más en el
mismo tiempo."**



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Realiza tu diseño sobre el tema de color negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

Teniendo en cuenta estas trazas de ejecución, se responde a las siguientes cuestiones, con el objetivo de calcular el CPI.

- a) Para el procesador MIPS segmentado que usa la técnica de anticipación de resultados o bypass y predicción de salto no tomado. Enumere los desvíos o bypass utilizados durante la ejecución del código dado. Para cada uno de ellos, indique la siguiente información (pueden quedar líneas en blanco):

Nº de iteración (p.e.: la primera, las tres primeras, todas, etc)	Origen del bypass		Destino del bypass	
	Número de línea de la instrucción:	Nombre de la etapa:	Número de línea de la instrucción:	Nombre de la etapa :
PRIMERA	2	EX	3	EX
TODAS	3	MEM	4	EX
TODAS	4	EX	5	ID
1 ^a , 3 ^a y 5 ^a (negativos)	6	MEM	8	EX
TODAS	9	EX	10	ID

- b) Indique el número total de instrucciones que se ejecutan.

Instrucciones que se ejecutan una vez, porque están fuera del bucle, ya sea al principio o al final = 2 + 1 = 3

Instrucciones que se ejecutan en cada iteración: Hay que distinguir entre el número de instrucciones que se ejecutan en el caso de ser un número positivo (5 instrucciones) o negativo (8 instrucciones) = 5*3 + 8*3 = 39

Nº de instrucciones = 3 + 39 = 42 instrucciones

- c) Calcule el número medio de ciclos por instrucción (CPI) empleados en la ejecución completa del programa dado.

Para calcular el CPI, debemos calcular el número de ciclos de bloqueo de datos y de control:

- Todas las iteraciones tienen el mismo número de ciclos de bloqueos de datos = 3 ciclos x 6 iteraciones
- En cuanto a los ciclos de bloqueo de control, hay 8 ciclos de bloqueo de control:
 - o el salto correspondiente al bucle for (instrucción 10) se toma 5 veces, provocando 5 ciclos de bloqueo de control
 - o el salto correspondiente al if (instrucción 5) se toma cuando el número es positivo, en 3 ocasiones, provocando 3 ciclos de bloqueo de control

$$\text{CPI} = (4 + 42 + 3 \times 6 + 8) / 42 = 1,714$$

A continuación se muestra la traza de ejecución para la primera iteración, que se trabaja con un número de valor 0. Para las otras iteraciones correspondientes a números cuyo valor es 0, sería equivalente, excepto que las dos primeras instrucciones no se ejecutarían (no pertenecen al bucle for).

1ª iteración: Nº valor 0 (*) SALTO TOMADO; (**) SALTO NO TOMADO														
addi \$t2,\$t0,20	IF	ID	EX	MEM	WB									
addi \$t3,\$t0,\$t0		IF	ID	EX	MEM									
lw \$t1,0(\$t2)		IF	ID	EX	MEM	WB								
beqz \$t1,etiq2			IF	-	-	ID	EX	MEM	WB					
add \$t3,\$t3,1(*)						IF	"aborted"							
subi \$t2,\$t2,4							IF	ID	EX	MEM	WB			
bnez \$t2,etiq1(**)								IF	-	ID	EX	MEM	WB	
sw \$t3,1000(\$t0)									IF _{sw}	"aborted"				

A continuación se muestra la traza de ejecución para la segunda iteración, que se trabaja con un número distinto de 0. Para las otras iteraciones correspondientes a números distintos de 0, sería equivalente, excepto para la última iteración en la que la última instrucción (la que está fuera del bucle) si se ejecutaría, es decir no se aborta en la etapa IF.

2ª iteración: Nº POSITIVO (*) SALTO NO TOMADO; (**) SALTO TOMADO														
lw \$t1,0(\$t2)		IF	ID	EX	MEM	WB								
beqz \$t1,etiq2(*)			IF	-	-	ID	EX	MEM						
add \$t3,\$t3,1						IF	ID	EX	MEM	WB				
subi \$t2,\$t2,4							IF	ID	EX	MEM	WB			
bnez \$t2,etiq1(**)								IF	-	ID	EX	MEM	WB	
sw \$t3,1000(\$t0)									IF _{sw}	"aborted"				

Teniendo en cuenta estas trazas de ejecución, se responde a las siguientes cuestiones, con el objetivo de calcular el CPI.

- b) Para el procesador MIPS segmentado que usa la técnica de anticipación de resultados o bypass y predicción de salto no tomado. Enumere los desvíos o bypass utilizados durante la ejecución del código dado. Para cada uno de ellos, indique la siguiente información (pueden quedar líneas en blanco):

Nº de iteración (p.e.: la primera, las tres primeras, todas, etc)	Origen del bypass			Destino del bypass	
	Número de línea de la instrucción:	Nombre de la etapa:		Número de línea de la instrucción:	Nombre de la etapa :
PRIMERA	1	MEM		3	EX
TODAS	6	EX		7	ID

PANDURA



MOTIVACIÓN



CONCENTRACIÓN



PANDURA



*"Sin Pandora,
juegas en
desventaja."*



*"Memorizo un
50% más en el
mismo tiempo."*



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Realiza tu diseño sobre el tema de color negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

A continuación se muestra la traza de ejecución para la primera iteración. Hay que tener en cuenta que la instrucción seqi se comporta como una función aritmética y por tanto se procesa en la etapa EX, etapa en la que necesitará el valor de sus operandos.

1ª iteración: (*) SALTO TOMADO							
addi \$t2,\$t0,100	IF	ID	EX	MEM	WB		
lw \$t1,0(\$t2)		IF	ID	EX	MEM	WB	
addi \$t1,\$t1,1		IF	ID	-	EX	MEM	WB
sw \$t1,0(\$t2)			IF	-	ID	EX	MEM
addi \$t2,\$t2,4					IF	ID	EX
seqi \$t3,\$t2,140						IF	ID
beqz \$t3,INC(*)							EX
sw \$t0,0(\$t2)							IF _{sw}
							"aborted"

Para el resto de iteraciones, sería equivalente a la siguiente traza. Se diferencia de la anterior, en que la primera instrucción no se ejecuta. También hay que tener en cuenta que en la última iteración, el salto no se toma y por tanto la última instrucción del código (instrucción 8) no se abortaría.

2ª- 9ª iteración: (*) SALTO TOMADO							
lw \$t1,0(\$t2)		IF	ID	EX	MEM	WB	
addi \$t1,\$t1,1			IF	ID	-	EX	MEM
sw \$t1,0(\$t2)				IF	-	ID	EX
addi \$t2,\$t2,4					IF	ID	EX
seqi \$t3,\$t2,140						IF	ID
beqz \$t3,INC(*)							EX
sw \$t0,0(\$t2)							IF _{sw}
							"aborted"

Teniendo en cuenta estas trazas de ejecución, se responde a las siguientes cuestiones, con el objetivo de calcular el CPI.

- b) Para el procesador MIPS segmentado que usa la técnica de anticipación de resultados o bypass y predicción de salto no tomado. Enumere los desvíos o bypass utilizados durante la ejecución del código dado. Para cada uno de ellos, indique la siguiente información (pueden quedar líneas en blanco):

Nº de iteración	Origen del bypass			Destino del bypass	
	Número de línea	Nombre de la etapa:		Número de línea	Nombre de la etapa :
PRIMERA	1	EX		2	EX
TODAS	2	MEM		3	EX
TODAS	3	MEM		4	MEM
TODAS	5	EX		6	EX
TODAS	6	EX		7	ID

25

Reservados todos los derechos. Queda permitida la impresión en su totalidad.

No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

TRAZA de ejecución:

Para hacer la traza de ejecución, hay que tener en cuenta que en el código hay 1 instrucción de salto (correspondiente al bucle for). Este salto ES TOMADO en todas las iteraciones del bucle, excepto en la última.

A continuación se muestra la traza de ejecución para la primera iteración.

1ª iteración: (*) SALTO TOMADO															
add \$t0,\$t0,\$zero	IF	ID	EX	MEM	WB										
lw \$t1,1000(\$t0)		IF	ID	EX	MEM	WB									
lw \$t2,1040(\$t0)			IF	ID	EX	MEM	WB								
addu \$t3, \$t1, \$t2				IF	ID	-	EX	MEM	WB						
sw \$t3, 1000(\$t0)					IF	-	ID	EX	MEM	WB					
addi \$t0,\$t0,4							IF	ID	EX	MEM	WB				
seqi \$t4, \$t0, 40								IF	ID	EX	MEM	WB			
beq \$t4,\$zero,etq(*)								IF	-	ID	EX	MEM	WB		

Para el resto de iteraciones, sería equivalente a la siguiente traza. Se diferencia de la anterior, en que la primera instrucción no se ejecuta. También hay que tener en cuenta que en la última iteración, el salto no se toma.

2ª- 9ª iteración: (*) SALTO TOMADO															
lw \$t1,1000(\$t0)		IF	ID	EX	MEM	WB									
lw \$t2,1040(\$t0)			IF	ID	EX	MEM	WB								
addu \$t3, \$t1, \$t2				IF	ID	-	EX	MEM	WB						
sw \$t3, 1000(\$t0)					IF	-	ID	EX	MEM	WB					
addi \$t0,\$t0,4							IF	ID	EX	MEM	WB				
seqi \$t4, \$t0, 40								IF	ID	EX	MEM	WB			
beq \$t4,\$zero,etq(*)								IF	-	ID	EX	MEM	WB		

Teniendo en cuenta estas trazas de ejecución, se responde a las siguientes cuestiones, con el objetivo de calcular el CPI.

- b) Para el procesador MIPS segmentado que usa la técnica de anticipación de resultados o bypass y predicción de salto no tomado. Enumere los desvíos o bypass utilizados durante la ejecución del código dado. Para cada uno de ellos, indique la siguiente información (pueden quedar líneas en blanco):

Nº de iteración (p.e.: la primera, las tres primeras, todas, etc)	Origen del bypass			Destino del bypass		
	Número de línea de la instrucción:	Nombre de la etapa:		Número de línea de la instrucción:	Nombre de la etapa :	
PRIMERA	1	EX		2	EX	
PRIMERA	1	MEM		3	EX	

PANDURA



MOTIVACIÓN



CONCENTRACIÓN



PANDURA



*"Sin Pandura,
juegas en
desventaja."*



*"Memorizo un
50% más en el
mismo tiempo."*



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Realiza tu diseño sobre el tema de color negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

A continuación, se muestra la traza de ejecución de la segunda a la 9^a iteración, en la que se observa que ya no existen ningún ciclo de bloqueo de datos. La primera iteración, será equivalente, salvo que se ejecuta una instrucción más al principio que activarán dos "bypass" (al igual que en el código sin reordenar).

2 ^a - 9 ^a iteración: (*) SALTO TOMADO									
2 lw \$t1,1000(\$t0)		IF	ID	EX	MEM	WB			
3 lw \$t2,1040(\$t0)		IF	ID	EX	MEM	WB			
4 addi \$t0,\$t0,4			IF	ID	EX	MEM	WB		
5 addu \$t3, \$t1, \$t2				IF	ID	EX	MEM	WB	
6 seqi \$t4, \$t0, 40					IF	ID	EX	MEM	WB
7 sw \$t3, 996(\$t0)						IF	ID	EX	MEM
8 beq \$t4,\$zero,etq(*)							IF	ID	EX
								MEM	WB

Teniendo en cuenta estas trazas de ejecución, se responde a las siguientes cuestiones, con el objetivo de calcular el CPI.

- f) Para el procesador MIPS segmentado que usa la técnica de anticipación de resultados o bypass y predicción de salto no tomado. Enumere los desvíos o bypass utilizados durante la ejecución del código dado. Para cada uno de ellos, indique la siguiente información (pueden quedar líneas en blanco):

Nº de iteración (p.e.: la primera, las tres primeras, todas, etc)	Origen del bypass		Destino del bypass	
	Número de línea de la instrucción:	Nombre de la etapa:	Número de línea de la instrucción:	Nombre de la etapa :
PRIMERA	1	EX	2	EX
PRIMERA	1	MEM	3	EX
TODAS	3	MEM	5	EX
TODAS	4	MEM	6	EX
TODAS	5	MEM	7	EX
TODAS	6	EX	8	ID

- g) Calcule el número medio de ciclos por instrucción (CPI) empleados en la ejecución completa del programa dado.

Para calcular el CPI, debemos calcular el número de ciclos de bloqueo de datos y de control:

- Todas las iteraciones tienen el mismo número de ciclos de bloques de datos = 0
- En cuanto a los ciclos de bloqueo de control, hay 9 ciclos de bloqueo de control (10 iteraciones -1):

$$\text{CPI} = (4 + 71 + 0 + 9) / 71 = 1,183$$

Existe una mejora de CPI

$$S = 1,464 / 1,183 = 1,2375$$

Computadores paralelos:

1.

Según la ley de Amdahl:

$$A = \frac{1}{1-f + f/p} = \frac{1}{1-0,95 + 0,95/10} = 16,8965$$

2.

Según la ley de Amdahl:

$$A = \frac{1}{1-f + f/p} \rightarrow 10 \geq \frac{1}{0,05 + \frac{1-0,05}{p}} \rightarrow 0,6 + \frac{9,4}{p} \geq 1 \rightarrow$$

$$\frac{9,4}{p} \geq 0,4 \rightarrow p \geq \frac{9,4}{0,4} = 23,5 \approx 24 \text{ procesadores al menos.}$$

3.

Según la ley de Amdahl:

$$A = \frac{1}{1-f + f/p} \rightarrow 80 = \frac{1}{f_{\max} + \frac{1-f_{\max}}{100}} \rightarrow 80 \cdot f_{\max} + \frac{80 - 80 f_{\max}}{100} = 1 \rightarrow$$

$$\rightarrow \frac{8000 f_{\max} + 80 - 80 f_{\max}}{100} = 1 \rightarrow 7920 f_{\max} + 80 = 100 \rightarrow$$

$$\rightarrow 7920 f_{\max} = 20 \rightarrow f_{\max} = \frac{20}{7920} = 0,0025, \text{ es decir, la fracción máxima del programa que puede ser secuencial es del } 0,25\%.$$

PANDURA

MOTIVACIÓN



CONCENTRACIÓN



MEMORIA

**PANDURA**

**"Sin Pandora,
juegas en
desventaja."**



**"Memorizo un
50% más en el
mismo tiempo."**



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Recarga tu energía sobre el fondo de colores negros.

Arquitectura de Computadores
Rubén Bueno Menéndez

PROBLEMA 1-a. (1 punto) Un procesador de frecuencia de reloj de 2 GHz, ejecuta un programa formado por una secuencia de 10^6 instrucciones, con un CPI de 1,25. Se quiere desarrollar una nueva versión del procesador capaz de ejecutar el mismo programa con el mismo valor de CPI, consiguiendo una reducción del 40% en el tiempo de ejecución. Determine la frecuencia de reloj de esta nueva versión del procesador.

$$\begin{aligned} t_{cpu} &= \text{nciclos} \times \text{tciclos} = (\text{ninst} \times \text{CPI}) / \text{fclk} \\ t_1 &= (10^6 \times 1,25) / 2 \times 10^9 \text{ s-1} = 0,625 \text{ ms} \\ t_{nuevo} &= (10^6 \times 1,25) / \text{fclk_nuevo} = 0,60 \times t_1 = 0,60 \times (10^6 \times 1,25) / 2 \times 10^9 \text{ s-1} \\ \text{fclk_nuevo} &= (2 \times 10^9 \text{ s-1}) / 0,60 = 3,33 \text{ GHz} \end{aligned}$$

PROBLEMA 1-b. (1 punto) Un procesador de frecuencia de reloj de 4 GHz, ejecuta un programa formado por una secuencia de 10^6 instrucciones, con un CPI de 1,25. Calcule cuanto hay que mejorar el CPI de las instrucciones si se quiere ejecutar el mismo programa el doble de rápido.

$$\begin{aligned} t_{cpu} &= \text{nciclos} \times \text{tciclos} = (\text{ninst} \times \text{CPI}) / \text{fclk} \\ t_1 &= (10^6 \times 1,25) / 4 \times 10^9 \text{ s-1} = 0,3125 \text{ ms} \\ t_{rapido} &= (10^6 \times \text{CPIrapido}) / 4 \times 10^9 \text{ s-1} = 0,50 \times t_1 = 0,50 \times (10^6 \times 1,25) / 4 \times 10^9 \text{ s-1} \\ \text{CPIrapido} &= 0,50 \times 1,25 = 0,625 \end{aligned}$$

PROBLEMA 1-c. (1 punto) Un procesador de frecuencia de reloj de 3 GHz, ejecuta un programa formado por una secuencia de 10^3 instrucciones, con un CPI de 0,75. Calcule cuánto se mejora el tiempo de ejecución si se reduce un 20% el CPI de las instrucciones, para ello calcule la aceleración.

$$\begin{aligned} t_{cpu} &= \text{nciclos} \times \text{tciclos} = (\text{ninst} \times \text{CPI}) / \text{fclk} \\ t_{cpu} &= (10^3 \times 0,75) / 3 \times 10^9 \text{ s-1} = 0,25 \times 10^{-6} \text{ s} \\ t_{mejora} &= (10^3 \times 0,80 \times 0,75) / 3 \times 10^9 \text{ s-1} = 0,2 \times 10^{-6} \text{ s} \\ \text{aceleracion} &= 0,25 / 0,2 = 1,25 \end{aligned}$$

EXAMEN OFICIAL, CONVOCATORIA DE JUNIO DE 2015

ARQUITECTURA DE COMPUTADORES
GRADO EN INGENIERÍA INFORMÁTICA – INGENIERÍA DEL SOFTWARE, 2º CURSO
UNIVERSIDAD DE SEVILLA.

Apellidos y nombre:

PROBLEMA 1 (2 puntos) Se dispone de un ordenador compuesto por un procesador de 32 bits y un único nivel de memoria caché separada. La memoria caché de datos es asociativa por conjunto de 2 vías, de 1KByte de capacidad y con bloques de 16 bytes. Su política de escritura es CB-WA y su política de reemplazo es la LRU. Por su parte, la capacidad máxima de direccionamiento de la memoria física es de 4GB.

El procesador ejecuta el siguiente código:

```
main(){  
    int a[8] ={1,2,3,4,5,6,7,8};  
    int b[8] ={10,11,12,13,14,15,16,17};  
    int c[8];  
    int i;  
  
    for(i=0;i<8;i++)  
        c[i]=a[i] + b[i];  
}
```

NOTA: El código trabaja con tres vectores de 8 números enteros, a, b y c, almacenados a partir de las siguientes direcciones 0x10010000, 0x10020000 y 0x10030000, respectivamente. La variable i se almacena en un registro interno del procesador. Suponemos, también, que inicialmente la memoria caché de datos está vacía. Recuerde, el tipo de dato int ocupa 4 bytes en memoria.

Se pide:

- 1.a** Dibuje un esquema indicando en cuántos campos se debe descomponer una dirección física de memoria, teniendo en cuenta la configuración de la memoria caché de datos descrita en el enunciado del problema. Para cada campo especifique el nombre del campo y su tamaño en bits.
- 1.b** Indique el número de fallos de acceso a memoria y su tipo (forzoso, conflicto y/o de capacidad) que se producen al ejecutar el código anterior, teniendo en cuenta la configuración de la caché de datos dada.
- 1.c** Calcule el *Miss Rate* o frecuencia de fallos de la memoria caché de datos al ejecutar el código anterior. Justifique su respuesta.
- 1.d** Describa **dos posibles soluciones** para mejorar el Miss Rate obtenido en el apartado anterior. Para cada una de las soluciones indique qué parámetro de la configuración de la caché cambiaría (sólo uno) e indique cuánto mejora el *Miss Rate* con cada cambio.

PANDURA



MOTIVACIÓN



CONCENTRACIÓN



MEMORIA



"Sin Pandora,
juegas en
desventaja."

PANDURA

Realiza tu diseño sobre el tema de color negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

c)

$$\text{Frecuencia de fallos} = \frac{\text{Nº fallos}}{\text{Nº accesos}} = \frac{6 + 18}{24} = 1$$

Es decir, ha habido un 100% de fallos respecto al número de accesos.



"Memorizo un
50% más en el
mismo tiempo."



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

d)

Aumentando la asociatividad por vías en 2 (es decir, 4 vías) sería una posible solución. Es decir, tendríamos una caché 4-asociativa de 16 conjuntos. En esta situación no habría fallos por conflicto.

Otra solución sería cambiando la política de reemplazo. En este caso la política coincide tanto para LRU, como FIFO, como LFU, por tanto, no queda otra que probar con aleatoria y tener suerte.

Problema de caché, prueba de evaluación continua, 2013:

a) La memoria principal es de $1\text{ KB} = 2^{10}$ bytes, por tanto, las direcciones físicas son de 10 bits.

A partir del tamaño de la línea, podemos obtener los bits de direcciones que se usan para el desplazamiento del bloque: $16\text{ B} \rightarrow \log_2(16) = 4$ bits de desplazamiento

Podemos obtener el número de conjuntos en total que tiene la caché de la siguiente manera:

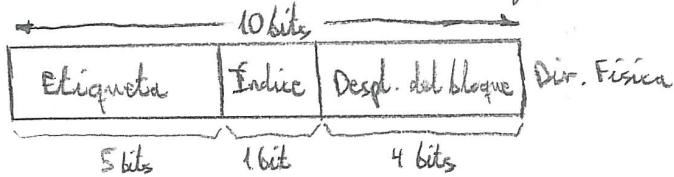
$$\text{Nº conjuntos} = \frac{\text{Tam de la cache}}{\text{Tam de la línea} \cdot \text{Nº vías}} = \frac{128}{16 \cdot 4} = 2 \text{ conjuntos en total}$$

Por tanto el número de bits necesarios para el índice será:

$$\text{Índice} = \log_2(\text{nº conjuntos}) = \log_2(2) = 1 \text{ bit de dirección.}$$

Entonces la etiqueta tendrá $10 - 4 - 1 = 5$ bits de direcciones.

Por tanto, el esquema sería el siguiente:



b)

Dir. de Memoria	Etiqueta	Conj	Línea	Acierto/Fallo	C0, L0	C0, L1	C0, L2	C0, L3	C1, L0	C1, L1	C1, L2	C1, L3
0x155	01010	1	0	F. Forzoso	-	-	-	-	M[01010]	-	-	-
0x015	00000	1	1	F. Forzoso	-	-	-	-	M[00000]	M[00000]	-	-
0x0F8	00111	1	2	F. Forzoso	-	-	-	-	M[01010]	M[00000]	M[00111]	-
0x051	00010	1	3	F. Forzoso	-	-	-	-	M[01010]	M[00000]	M[00111]	M[00010]
0x150	01010	1	0	Acierto	-	-	-	-	M[01010]	M[00000]	M[00111]	"
0x380	11100	0	0	F. Forzoso	M[11100]	-	-	-	M[01010]	M[00000]	M[00111]	"
0x393	11100	1	1	F. Forzoso	M[11100]	-	-	-	M[01010]	M[11100]	M[00111]	"
0x01F	00000	1	2	F. por Conjunto	M[11100]	-	-	-	M[01010]	M[11100]	M[00111]	"



MOTIVACIÓN



CONCENTRACIÓN



MEMORIA



*"Sin Pandora,
juegas en
desventaja."*



*"Memorizo un
50% más en el
mismo tiempo."*



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINKGO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Realiza tu diseño sobre el tema de color negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

ARQ. DE COMPUTS. I.S. - PRUEBA 1 DE EVALUACIÓN CONTINUA, 2013/14. GRUPO 1.

APELLIDOS, NOMBRE:

CUESTIÓN 3. (3 puntos)

Sea un sistema con un procesador cuya capacidad de direccionamiento es de 4GBytes, que utiliza memoria virtual paginada con un tamaño de página de 4KBytes y que puede acceder a una memoria principal de hasta 256MBytes.

- a) (0,5 punto) Dibuje la estructura del sistema, indicando los diferentes campos en que se divide cada dirección (dirección virtual y dirección física); así como los diferentes elementos necesarios para realizar el mecanismo de traducción. Indique el tamaño de todos los campos.

El sistema dispone de un sistema de memoria con caches separados. La cache de datos tiene 4 líneas y es de correspondencia directa. El tamaño de un bloque es de 16 bytes. Usa la política de escritura WT-NWA.

- b) (0,5 punto) Dibuje la estructura de la cache de datos, teniendo en cuenta la configuración de la caché dada en el enunciado. Y especifique también los campos en los que se divide la dirección física para esta configuración de cache. Indique el tamaño de todos los campos.
- c) (1 punto) Se quiere ejecutar el siguiente fragmento de programa escrito en C, en el que se inicializa un vector de caracteres con las 26 letras del alfabeto en mayúsculas.

```
int i;
char letras[26];
i = 0;
while (i < 26) {
    letras[i] = 'A' + i;
    i++;
}
```

El proceso encargado de la ejecución de este código tiene asignado un total de 6 páginas virtuales. El mecanismo de asignación de marcos usado es a partir del marco 256 (0x0100) de memoria principal.

Realice una traza de ejecución y represente el estado final de la cache, tras ejecutar el código, teniendo en cuenta solo los accesos a datos, suponiendo que la tabla de páginas y la cache inicialmente está vacía. Considere que el vector <>letras<> tiene asociado la dirección virtual 0x00001100. La variable «i» es una variable registro que reside en un registro del procesador.

Tenga en cuenta que el tipo de dato char ocupa en memoria 1 byte.

Para ello, ayúdese de una tabla con la siguiente información:

Variable/tipo de acceso (p.e.: letras[0]/write)	Dirección física	Etiqueta	Línea en cache	Fallo-tipo/Acierto

- d) (0,5 punto) Calcule el Miss rate o frecuencia de fallos de accesos a caché, para dicha traza de accesos.
- e) (0,5 punto) Explique cómo cambiaría el Miss rate del apartado anterior, si se cambiara la política de escritura de la memoria cache a CB-WA .

- b) Sabiendo que tenemos 28 bits de direcciones físicas y que ésta se divide en etiqueta, índice y desplazamiento de bloque:

$$\text{Desplazamiento de bloque} = \log_2(16) = 4 \text{ bits}$$

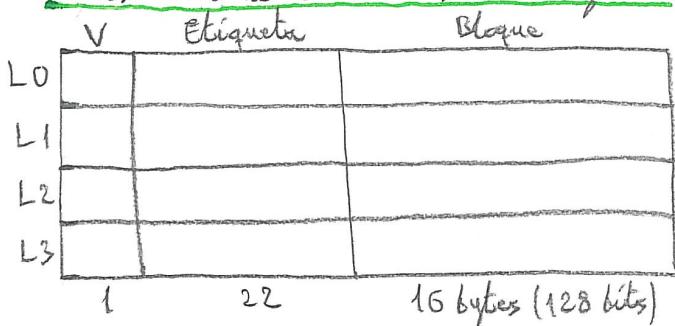
$$\text{Índice} = \log_2(4) = 2 \text{ bits}$$

$$\text{Etiqueta} = 28 - 4 - 2 = 22 \text{ bits}$$

Por tanto, la dirección física queda dividida de la siguiente manera:

DF	Etiqueta	Índice	Despl. Bloq.	28
	22	2	4	

La estructura de la caché será la siguiente:



- c) Hay que tener en cuenta que la política de escritura es WT-NWA, por lo que en caso de fallo, no se lleva el dato a la memoria caché.

La dirección virtual 0x00001100 corresponde al marco o número de página física 256 (0x0100), luego la dirección física se concatena con el desplazamiento de página $\rightarrow 0x0100100$

Variáble / tipo de acceso	Dirección física	Etiqueta	Línea en caché	Fallo / Acceso
letras [0] / write	0x0100100	0x00004004	0	Fallo forzoso
letras [1] / write	0x01001001	0x00004004	0	Fallo forzoso
...
letras [15] / write	0x0100101F	0x00004004	0	Fallo forzoso
letras [16] / write	0x0100110	0x00004004	1	Fallo forzoso
letras [17] / write	0x01001111	0x00004004	1	Fallo forzoso
...
letras [25] / write	0x0100111F	0x00004004	1	Fallo forzoso

PANDURA

MOTIVACIÓN



CONCENTRACIÓN



MEMORIA

**PANDURA**

Realiza tu diseño sobre el tema de color negro.



*"Sin Pandora,
juegas en
desventaja."*



*"Memorizo un
50% más en el
mismo tiempo."*



MELEENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Arquitectura de Computadores
Rubén Bueno Menéndez

ARQ. DE COMPUTS. I.S. - PRUEBA 1 DE EVALUACIÓN CONTINUA, 2013/14. GRUPO 2.

APELLOS, NOMBRE: _____

CUESTIÓN 3. (3 puntos)

Sea un sistema con un procesador cuya capacidad de direccionamiento es de 4GBytes, que utiliza memoria virtual paginada con un tamaño de página de 4KBytes y que puede acceder a una memoria principal de hasta 1GBytes.

- a) (0,5 punto) Dibuje la estructura del sistema, indicando los diferentes campos en que se divide cada dirección (dirección virtual y dirección física); así como los diferentes elementos necesarios para realizar el mecanismo de traducción. Indique el tamaño de todos los campos.

El sistema dispone de un sistema de memoria con caches separados. La cache de datos de 1KByte de capacidad es totalmente asociativa. El tamaño de un bloque es de 16 bytes. Usa la política de escritura WT-NWA y la política de reemplazo LRU.

- b) (0,5 punto) Dibuje la estructura de la cache de datos, teniendo en cuenta la configuración de la caché dada en el enunciado. Y especifique también los campos en los que se divide la dirección física para esta configuración de cache. Indique el tamaño de todos los campos.
- c) (1 punto) Se quiere ejecutar el siguiente fragmento de programa escrito en C, en el que se calcula los resultados de la tabla de multiplicar del número 8.

```
int i;
int producto[10];
for(i=1;i<11;i++) {
    producto[i-1] = 8*i;
}
```

El proceso encargado de la ejecución de este código tiene asignado un total de 6 páginas virtuales. El mecanismo de asignación de marcos usado es a partir del marco 7 (0x0007) de memoria principal.

Realice una traza de ejecución y represente el estado final de la cache, tras ejecutar el código, teniendo en cuenta solo los accesos a datos, suponiendo que la tabla de páginas y la cache inicialmente está vacía. Considere que el vector <>producto>> tiene asociado la dirección virtual 0x00032100. La variable «i» es una variable registro que reside en un registro del procesador.

Tenga en cuenta que el tipo de dato int ocupa en memoria 4 bytes.

Para ello, ayúdese de una tabla con la siguiente información:

Variable/tipo de acceso (p.e.: producto[0]/write)	Dirección física	Etiqueta	Línea en cache	Fallo-tipo/Acierto

- d) (0,5 punto) Calcule el Miss rate o frecuencia de fallos de accesos a caché, para dicha traza de accesos.
- e) (0,5 punto) Explique cómo cambiaría el Miss rate del apartado anterior, si se cambiara la política de escritura de la memoria cache a CB-WA.

6)

Sabiendo que tenemos 30 bits de direcciones físicas y que esta se divide en etiqueta y desplazamiento de bloque por ser una caché totalmente asociativa, entonces:

$$\text{Nº de líneas} = \frac{\text{Tam. de la caché}}{\text{Tam. del bloque}} = \frac{1\text{KB}}{16} = \frac{2^{10}}{16} = 64 \text{ líneas, o en}$$

este caso, sería como 64 vías de un único conjunto.

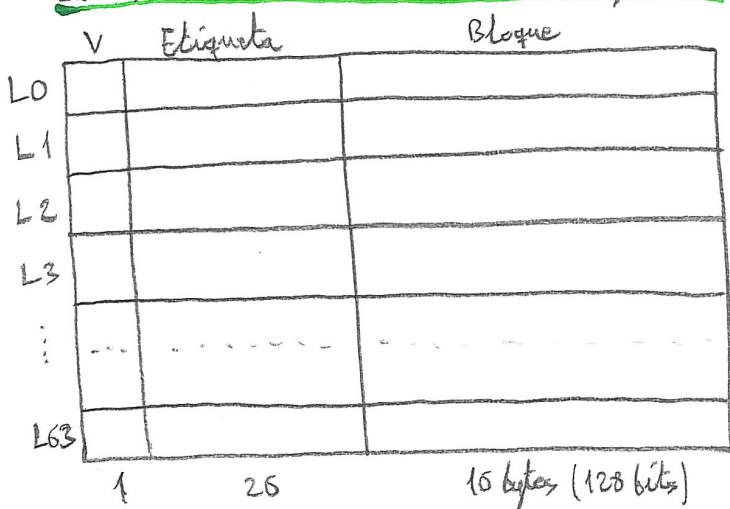
$$\text{Desplazamiento del bloque} = \log_2(16) = 4 \text{ bits}$$

$$\text{Etiqueta} = 30 - 4 = 26 \text{ bits}$$

Por tanto, la dirección física queda dividida de la siguiente manera:

DF	Etiqueta	Desp. Bloque	30
	26	4	

La estructura de la caché será la siguiente:



- c) Hay que tener en cuenta que la política de escritura es WT-NWA, por lo que en caso de fallo de escritura, no se lleva el dato a la memoria caché. La dirección virtual 0x00032100 corresponde al marco o número de página física 7 (0x0007), luego la dirección física se concatena con el desplazamiento de página → 0x0007100.

PANDURA**MOTIVACIÓN****CONCENTRACIÓN****MEMORIA****PANDURA****Realiza tu diseño sobre el tema de color negro.**

*"Sin Pandora,
juegas en
desventaja."*



*"Memorizo un
50% más en el
mismo tiempo."*



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Arquitectura de Computadores
Rubén Bueno Menéndez

ARQ. DE COMPUTS. I.S. - PRUEBA 1 DE EVALUACIÓN CONTINUA, 2013/14. GRUPO 3.**APELLIDOS, NOMBRE: _____****CUESTIÓN 3. (3 puntos)**

Sea un sistema con un procesador cuya capacidad de direccionamiento es de 4GBytes, que utiliza memoria virtual paginada con un tamaño de página de 4KBytes y que puede acceder a una memoria principal de hasta 1GBbytes.

- a) (0,5 punto) Dibuje la estructura del sistema, indicando los diferentes campos en que se divide cada dirección (dirección virtual y dirección física); así como los diferentes elementos necesarios para realizar el mecanismo de traducción. Indique el tamaño de todos los campos.

El sistema dispone de un sistema de memoria con caches separados. La cache de datos de 1 KByte de capacidad es asocialiva por conjuntos de 4 vías. El tamaño de un bloque es de 16 bytes. Usa la política de escritura WT-NWA y la política de reemplazo LRU.

- b) (0,5 punto) Dibuje la estructura de la cache de datos, teniendo en cuenta la configuración de la caché dada en el enunciado. Y especifique también los campos en los que se divide la dirección física para esta configuración de cache. Indique el tamaño de todos los campos.
- c) (1 punto) Se quiere ejecutar el siguiente fragmento de programa escrito en C, en el que se calcula el factorial de los 10 primeros números.

```
int i,j;
int factorial[10]={1,1,1,1,1,1,1,1,1,1};
for(i=0;i<10;i++){
    for(j=i+1;j>0; j--) {
        factorial[i] = factorial[i]*j;
    }
}
```

El proceso encargado de la ejecución de este código tiene asignado un total de 6 páginas virtuales. El mecanismo de asignación de marcos usado es a partir del marco 16 (0x0010) de memoria principal.

Realice una traza de ejecución y represente el estado final de la cache, tras ejecutar el código, teniendo en cuenta solo los accesos a datos, suponiendo que la tabla de páginas y la cache inicialmente está vacía. Considere que el vector <<factorial>> tiene asociado la dirección virtual 0x00076500. Las variables «i» y «j» son variables registro que residen en los registros del procesador.

Tenga en cuenta que el tipo de dato int ocupa en memoria 4 bytes.

Para ello, ayúdese de una tabla con la siguiente información:

Variable/tipo de acceso (p.e.: factorial[0]/write)	Dirección física	Etiqueta	Línea en cache	Fallo-tipo /Acierto

- d) (0,5 punto) Calcule el Miss rate o frecuencia de fallos de accesos a caché, para dicha traza de accesos.
- e) (0,5 punto) Explique cómo afecta al funcionamiento del apartado anterior, si se cambiara la política de escritura de la memoria cache a CB-WA .

Reservados todos los derechos.
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

6)

Sabiendo que tenemos 30 bits de direcciones físicas y que ésta se divide en etiqueta, índice y desplazamiento de bloque, entonces:

$$\text{Nº de conjuntos} = \frac{\text{Tam de La caché}}{\text{Tam del bloque} \cdot \text{Nº de vías}} = \frac{1 \text{ KB}}{16 \cdot 4} = \frac{2^{10}}{16 \cdot 4} = 16 \text{ conjuntos en total.}$$

Desplazamiento del bloque = $\log_2(15) = 4$ bits

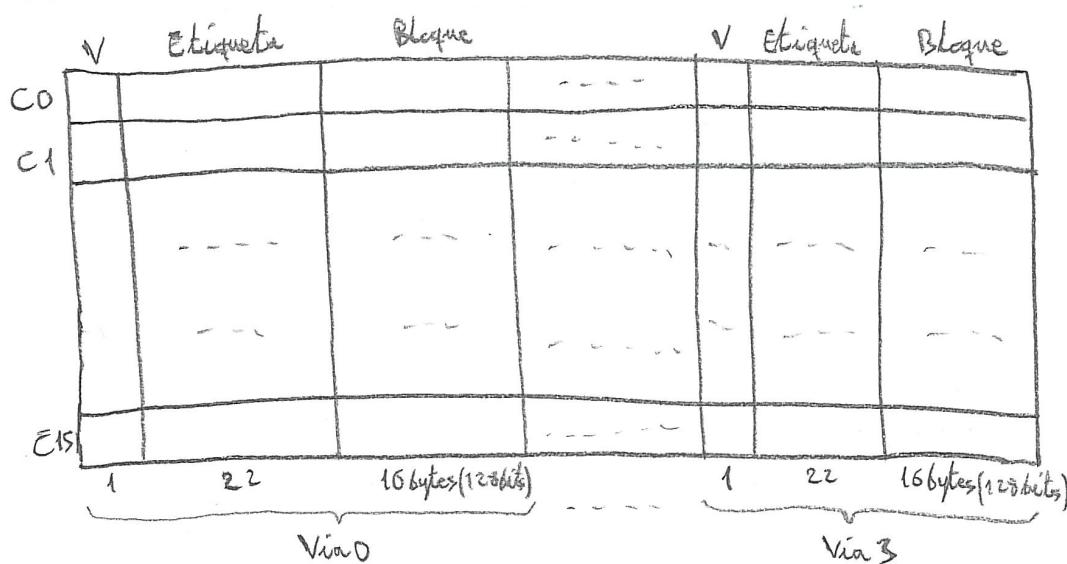
$$\text{Índice} = \log_2(\text{número de conjuntos}) = \log_2(16) = 4 \text{ bits}$$

$$\text{Etiquetas} = 30 - 4 - 4 = 22 \text{ bits}$$

Por tanto, la dirección física queda dividida de la siguiente manera:

DF	Etiqueta	Indice	Desp. Blague	30
22	4	4		

La estructura de la caché será la siguiente:



c) Hay que tener en cuenta que la política de escritura es WT-NWA, por lo que en caso de fallo de escritura, no se lleva el dato a la memoria caché. La dirección virtual 0x00076500 corresponde al marco 16 (0x0010), luego la dirección física se concatena con el desplazamiento de página → 0x0010500

PANDORA



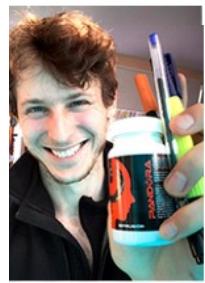
MOTIVACIÓN



CONCENTRACIÓN



MEMORIA



"Sin Pandora,
juegas en
desventaja."



"Memorizo un
50% más en el
mismo tiempo."

Realiza tu diseño sobre el tema de color negro.

Arquitectura de Computadores
Rubén Bueno Menéndez

e)

Si se cambia la política de escritura de la memoria caché a CB-WA, entonces en caso de acierto cuando haya escritura, se llevará el dato solo a memoria caché, y en caso de fallo cuando haya escritura, se llevará el dato a memoria caché y memoria principal.

En este caso, el miss rate no va a variar, ya que los aciertos que hay en caché son producidos por el acceso de lectura (la primera vez falso forzoso).



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Supongamos un MIPS con camino de datos optimizado para saltos, Arquitectura de Computadores
Rubén Bueno Menéndez
y toda la lógica de anticipación de resultados necesaria para evitar bloqueos. $f_{reloj} = 1\text{ GHz}$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
addu \$t0, \$t0, \$zero	IF	ID	EX	MEM	WB													
lw \$t1, 1000(\$t0)		IF	ID	EX	MEM	WB												
addiu \$t0, \$t0, 4			IF	ID	EX	MEM	WB											
et1: lw \$t3, 1000(\$t0)				IF	ID	EX	MEM	WB										
slt \$t2, \$t1, \$t3					IF	ID	EX	MEM	WB									
beq \$t2, \$zero, et2						IF	-	-	ID	EX	MEM	WB						
addu \$t1, \$t3, \$zero						-	-	IF _{addu}	====	ABORTO								
et2: addiu \$t0, \$t0, 4						-	-		IF	ID	EX	MEM	WB					
seq \$t2, \$t0, 40						-	-			IF	ID	EX	MEM	WB				
beq \$t2, \$zero, et1						-	-				IF	-	ID	EX	MEM	WB		
sw \$t1, 2000(\$t0)						-	-				-	IF _{sw}	====	ABORTO				

Considere que la memoria a partir de la posición 1000 contiene los siguientes valores en base 10: 24, 17, 20, 14, 16, 25, 22, 21, 19, 18.

Para el programa completo tenemos:

Nº de instrucciones:

$$3 + \underbrace{(3+3) \cdot 5}_{3 \text{ primeras instrucciones de et1 y et2}} + 4 + \underbrace{3 + (3+3) \cdot 3}_{1 \text{ iteración con etl completo y 3 primeras instrucciones de et2 con las 3 primeras instrucciones de cada etiquete}} + 1 = \boxed{59 \text{ instrucciones}}$$

Instrucción sw

3 iteraciones de et1 y et2 con las 3 primeras instrucciones de cada etiqueta

Nº total de bloques de datos:

Hay 3 bloques de datos en la primera iteración, por tanto, en el programa completo:

$$3 \cdot 9 = \boxed{27 \text{ bloques de datos}}$$

Nº total de bloques de control:

Se produce un bloqueo de control por cada salto tomado, por tanto:

$$(1+1) \cdot 5 + 1 + 1 \cdot 3 + 1 \cdot 2 = \boxed{16 \text{ bloques de control}}$$

Tiempo total que consume el programa:

$$t_{CPU} = \frac{\text{Latencia inicial} + \text{nº instr.} + \text{nº bloques}}{f_{reloj}} = \frac{4 + 59 + 27 + 16}{1 \cdot 10^9} = \boxed{106 \cdot 10^{-9} \text{ s}}$$

$$CPI_{medio}: CPI_m = \frac{\text{Latencia inicial} + \text{nº instr.} + \text{nº bloques}}{\text{nº instr.}} = \frac{4 + 59 + 27 + 16}{59} = \boxed{1,79 \text{ ciclos/instr.}}$$

PANDURA



MOTIVACIÓN



CONCENTRACIÓN



MEMORIA



PANDURA



"Sin Pandora,
juegas en
desventaja."



"Memorizo un
50% más en el
mismo tiempo."



MELENA DE LEÓN



BACOPA MONNIERI



CAMELLIA SINENSIS



GINGKO BILOBA



HUPERZIA SERRATA



COFFEA ARABICA

Positiva tu diseño sobre el fondo de color negro.

Suponemos un MIPS con camino de datos optimizado para saltos, predicción de salto, ejecución de la lógica de anticipación de resultados necesaria para evitar bloqueos. Frecu. = 1 GHz

Arquitectura de Computadores

Rubén Bueno Menéndez

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
addu \$t0, \$t0, \$zero	IF	ID	EX	MEM	WB													
addu \$t1, \$t1, \$zero		IF	ID	EX	MEM	WB												
et1: lb \$t2, 1000(\$t0)			IF	ID	EX	MEM	WB											
seq \$t3, \$t2, 65				IF	ID	-	EX	MEM	WB									
beq \$t3, \$zero, et2					IF	-	-	ID	EX	MEM	WB							
addiu \$t1, \$t1, 1						-	-	IF	ID	EX	MEM	WB						
et2: addiu \$t0, \$t0, 1						-	-		IF	ID	EX	MEM	WB					
seq \$t3, \$t0, 7						-	-			IF	ID	EX	MEM	WB				
beq \$t3, \$zero, et1						-	-				IF	-	ID	EX	MEM	WB		
sw \$t1, 2000(\$t0)						-	-				-	IF	-	ID	EX	MEM	WB	ABORTO

Considerese que la memoria a partir de la posición 1000 contiene la cadena de caracteres "AMAPOLA"; y recuerde que el número 65 es el valor del código ASCII del carácter 'A'.

Para el programa completo tenemos:

Nº de instrucciones:

$$2 + (4+3) + (3+3) + (4+3) + (3+3) \cdot 3 + (4+4) = 48 \text{ instrucciones}$$

2 primeras iteración de et1 completa y et2 con las 3 primeras instrucciones
3 primeras instrucciones de et2 etiqueta

1 iteración de et1 y et2 completas

Nº total de bloques de datos:

Hay 3 bloques de datos en la primera iteración, por tanto en el programa completo:

$$3 \cdot 7 = 21 \text{ bloques de datos}$$

Nº total de bloques de control:

Se produce un bloqueo de control por cada salto tomado, por tanto:

$$1 + (1+1) + 1 + (1+1) \cdot 3 = 10 \text{ bloques de control}$$

Tiempo total que consume el programa:

$$t_{CPU} = \frac{\text{Latencia inicial} + \text{nº inst.} + \text{nº bloques}}{\text{frecu.}} = \frac{4 + 48 + 21 + 10}{1 \cdot 10^9} = 83 \cdot 10^{-9} \text{ s}$$

CPI medio:

$$CPI_m = \frac{\text{Latencia inicial} + \text{nº inst.} + \text{nº bloques}}{\text{nº inst.}} = \frac{4 + 48 + 21 + 10}{48} = 1,72 \text{ ciclos/instrucción}$$