

### Código de honor y excelencia académica

Me comprometo a responder esta prueba respetando los valores de la Universidad de Santiago de Chile, en particular aquellos referidos a la excelencia y la transparencia y probidad. Esto significa que no revisaré el material del curso ni otras fuentes durante el desarrollo de esta prueba, y que trabajaré en mis respuestas de manera individual, sin discutir ni consultar con otras personas por ningún medio.

Firma



7

¡ Muy buen  
trabajo!

## **PREGUNTA 1**

### **Pregunta 1. Mapeo Directo (2 pts.)**

Considere la siguiente secuencia de accesos a memoria, entregadas como direcciones de byte y escritas en decimal. Las direcciones de memoria son de 32 bits, y las palabras son de tamaño 4 bytes.

0, 0, 4, 4, 8, 64, 128, 512, 0, 32, 1024, 1028, 1032, 1036, 0, 524, 512, 1036

Se quiere encontrar un diseño óptimo de caché para este caso. Se tienen 3 diseños posibles, todos de mapeo directo y con una capacidad total de 512 bytes para datos, pero que difieren en el tamaño del bloque: C1 tiene bloques de 2 palabras; C2, bloques de 4 palabras, y C3, bloques de 8 palabras.

- A) (0,6 pts.) Para cada una de las direcciones y para cada diseño de caché, escriba el número correspondiente de bloque de la memoria.
- B) (0,6 pts.) Para cada una de las direcciones y para cada diseño de caché, indique el “*index*” en **binario**.
- C) (0,2 pts.) ¿Cuántos **bits** de “tag” requiere cada diseño de caché?
- D) (0,3 pts.) Considerando solamente desaciertos de caché, ¿cuál diseño es el mejor?
- E) (0,3 pts.) Si la penalización de memoria es de 60 ciclos, y C1, C2 y C3 tienen tiempos de acceso de 2, 3 y 5 ciclos, respectivamente, ¿qué diseño de caché es el mejor para esta secuencia de accesos a memoria?

# Pregunta 1. Mapeo Directo

Capacidad total: 512 bytes para datos

C1 → bloques de 2 palabras → 64 bloques ( $2^6$ )  
C2 → bloques de 4 palabras → 32 bloques ( $2^5$ )  
C3 → bloques de 8 palabras → 16 bloques ( $2^4$ )

A)

Dirección	# bloque C1 (8 bytes)	# bloque C2 (16 bytes)	# bloque C3 (32 bytes)
0	0 M	0 M	0 M
0	0 H	0 H	0 H
4	0 H	0 H	0 H
4	0 H	0 H	0 H
8	1 M	0 H	0 H
64	8 M	4 M	2 M
128	16 M	8 M	4 M
512	64 M	32 M	16 M
0	0 M	0 M	0 M
32	4 M	2 M	1 M
1024	128 M	64 M	32 M
1028	128 H	64 H	32 H
1032	129 M	64 H	32 H
1036	129 H	64 H	32 H
0	0 M	0 M	0 M
524	65 M	32 M	16 M
512	64 M	32 H	16 H
1036	129 M	64 M	32 M

0,6

### B) Index en binario

Dir	C1 (6 bits)	C2 (5 bits)	C3 (4 bits)
0	0 → 000000	0 → 00000	0 → 0000
0	0 → 000000	0 → 00000	0 → 0000
4	0 → 000000	0 → 00000	0 → 0000
4	0 → 000000	0 → 00000	0 → 0000
8	1 → 000001	0 → 00000	0 → 0000
64	8 → 001000	4 → 00100	2 → 0010
128	16 → 010000	8 → 01000	4 → 0100
512	0 → 000000	0 → 00000	0 → 0000
0	0 → 000000	0 → 00000	0 → 0000
32	4 → 000100	2 → 00010	1 → 0001
1024	0 → 000000	0 → 00000	0 → 0000
1028	0 → 000000	0 → 00000	0 → 0000
1032	1 → 000001	0 → 00000	0 → 0000
1036	1 → 000001	0 → 00000	0 → 0000
0	0 → 000000	0 → 00000	0 → 0000
512	-1 → 000001	0 → 00000	0 → 0000
512	0 → 000000	0 → 00000	0 → 0000
1036	1 → 000001	0 → 00000	0 → 0000

### c) Tag

$$\begin{aligned}
 C1 &: 32 - 6 - 3 = 23 \text{ bits} \\
 C2 &: 32 - 5 - 4 = 23 \text{ bits} \\
 C3 &: 32 - 4 - 5 = 23 \text{ bits}
 \end{aligned}$$

0, 6

0, 2

D)

C1: 13 desaciertos  
C2: 10 desaciertos  
C3: 10 desaciertos

Por lo tanto C2 y C3 son mejores que C1.

E)

Penalización de memoria 60 ciclos  
Tiempo acceso C1 = 2 ciclos  
C2 = 3 "  
C3 = 5 "

#accesos a mem  $\times$  #ciclos caché + #desac.  $\times$  #ciclos mem

$$\begin{aligned} C1 &= 18 \times 2 + 13 \times 60 = 816 \\ C2 &= 18 \times 3 + 10 \times 60 = 654 \\ C3 &= 18 \times 5 + 10 \times 60 = 690 \end{aligned}$$

Por lo tanto, el mejor diseño es C2



p. 13



## PREGUNTA 2

### Pregunta 2. Caché, caché, y más caché... (2 pts.)

Considere direcciones de memoria de 32 bits, y asuma un largo de palabra de 4 bytes en todos los casos. Se tienen tres configuraciones de caché distintas, con sus respectivos tamaños de datos indicados entre paréntesis, y para las cuales los bits de la dirección de memoria son usados para “tag”, “index” y “offset” como se indica.

Caché X (4 MiB):

Tag	Offset
31-10	9-0

Caché Y (2 MiB):

Tag	Index	Offset
31-21	20-7	6-0

Caché Z (1 MiB):

Tag	Index	Offset
31-18	17-2	1-0

- (0,3 pts.) Considere que la memoria principal no puede ser direccionada más allá de lo que permite el espacio de direcciones posibles. ¿Cuál es el tamaño máximo, en bytes, de la memoria principal?
- (0,3 pts.) Para cada caché, indique el tamaño del bloque en palabras y el número total de bloques.
- (0,3 pts.) Indique el tipo de mapeo de caché en cada caso (si hay alguna caché de conjunto asociativo, especifique el número de vías).
- (0,3 pts.) Para cada caché, encuentre la razón entre el total de bits requeridos para implementar la caché y los bits usados para datos.
- (0,8 pts.) Suponga que inicialmente cada caché está vacía. Para la siguiente secuencia de accesos a memoria principal expresadas en *byte*, calcule la tasa de desaciertos (“miss rate”) de cada caché, donde  $N$  es el número de bloques de esa caché y  $m$  es el número de bits de offset usados para esa caché.

$0 \cdot 2^m$	$1 \cdot 2^m$	$2 \cdot 2^m$	...	$(N-1) \cdot 2^m$	$0 \cdot 2^m + 1$	$1 \cdot 2^m + 1$	$2 \cdot 2^m + 1$	...	$(N-1) \cdot 2^m + 1$
---------------	---------------	---------------	-----	-------------------	-------------------	-------------------	-------------------	-----	-----------------------

## Pregunta 2

A) Como se tienen 32 bits de dirección de memoria y apuntan a un byte, entonces el tamaño máximo de la memoria principal es

$$2^{32} = 2^2 \times 2^{30} = 4 \text{ GiB}$$

✓  
0,3

B)

B) Cache X: 10 bits de offset → 1024

Cache X: 10 bits de offset → 1024 / 4 = 256 palabras/bloque

Cache Y: 7 bits de offset → 128 / 4 = 32 palabras/bloque

Cache Z: 2 bits de offset → 4 / 4 = 1 palabra/bloque

Número total de bloques:

X: 4 MiB =  $4 \times 2^{20} \text{ B} / 1024 = 4096 \rightarrow 4 \text{ Ki Bloques}$

Y: 2 MiB =  $2 \times 2^{20} \text{ B} / 128 = 16384 \rightarrow 16 \text{ Ki Bloques}$

Z: 1 MiB =  $2^{20} \text{ B} / 4 = 262144 \rightarrow 256 \text{ Ki Bloques}$

✓  
✓  
✓

0,3

c) Cache X: completamente asociativa, pues no tiene bits para el index. ✓

Cache Y: mapeo directo, pues se tiene 16384 :  $2^{14}$  (Bloques/Conjuntos)  
→ Se tiene 1 bloque por conjunto. ✓

Cache Z: asociativa de 4 vías, pues  
$$\frac{\# \text{Bloques}}{\# \text{Conjuntos}} = \frac{262144}{2^{16}} = 4$$
  
Hay 4 bloques por conjunto. ✓

D) 
$$\frac{\text{datos} + \text{extras}}{\text{datos}} = 1 + \frac{\text{extras}}{\text{datos}}$$

Cache X:  $1 + \frac{(22+1)}{1024 \times 8} = 1,0028$  ✓

Cache Y:  $1 + \frac{(11+1)}{128 \times 8} = 1,011718$  ✓

Cache Z:  $1 + \frac{(4+1)}{4 \times 8} = 1,46875$  ✓

0,3

0,3



E)

$$0 \cdot 2^{10} \mid 1 \cdot 2^{10} \mid 2 \cdot 2^{10}$$

Caché X  $\rightarrow m = 10$

Cada bloque: 1024 bytes, por lo que las direcciones en bytes de la memoria principal quedan como:

0 1 2 ... 1023 | 1024 ... 2047 | 2048 ...  
bloque 1                      bloque 2

Como es posible notar, la primera mitad de la secuencia toma el primer byte de cada bloque, comenzando con el bloque 0 hasta el bloque "N-1" y como la caché inicia vacía, estos N accesos son desaciertos de caché.

La otra mitad de la secuencia toma el segundo byte de cada bloque, pero como todos estos ya están en caché, se consideran aciertos.

Por lo tanto, para caché X la tasa de desaciertos es  $1/2 = 50\%$ .

Caché Y  $\rightarrow m=7$

$0 \cdot 2^7 \mid 1 \cdot 2^7 \mid 2 \cdot 2^7$

Cada bloque: 128 bytes

①... 127 | ①28... 255 |

Siguiendo la misma dinámica de la caché X, se tendrá que en la primera mitad se toma el primer byte de cada bloque, por lo que se tienen N aciertos.

Luego en la segunda mitad se toma el segundo byte de cada bloque, pero como ya están en caché, son aciertos.

Por lo tanto, para caché Y, la tasa de aciertos es  $1/2 = 50\%$ .

Caché Z  $\rightarrow m=2$

$0 \cdot 2^2 \mid 1 \cdot 2^2 \mid 2 \cdot 2^2$

Cada bloque: 4 bytes

①... 3 | ④... 7 | ⑧... 11

La primera mitad de la secuencia toma el primer byte de cada bloque, y como la caché está vacía, se tienen N aciertos.

Luego en la segunda mitad se toma el segundo byte de cada bloque, pero como ya están, se tiene N aciertos.

Por lo tanto, para caché Z, la tasa de aciertos es de un 50%.

0,8

## **PREGUNTA 3**

### **Pregunta 3. Varios (2 pts.)**

#### **Parte 1 – Rendimiento procesadores paralelos**

- A) (0,8 pts.) Considere una suma *no* paralelizable de  $N^2$  escalares, y una suma paralelizable de matrices  $M \times M$  (i.e.,  $M^2$  sumas). Encuentre la razón entre  $N$  y  $M$  que permite obtener una aceleración -con respecto del caso con 1 procesador- igual a la mitad del número de procesadores,  $P$ .

#### **Parte 2 – Memoria virtual**

Considere un sistema de memoria virtual con las siguientes características:

Dirección virtual (bits)	Memoria DRAM física instalada (GiB)	Tamaño de página (KiB)	Tamaño de una entrada de la tabla de páginas (bytes)
42	32	8	4

- B) (0,4 pts.) ¿Cuántas entradas en total se requieren para la tabla de páginas, y cuánta memoria física se requiere para almacenar la tabla de páginas?

#### **Parte 3 – ¡De nuevo caché!**

Considere un computador con una memoria principal de 4 GiB, una caché asociativa de 4 vías con política de reemplazo MRU, direcciones de memoria de 32 bits con 9 bits de "*index*" y bloques de tamaño 8 bytes. Además, considere el siguiente bucle escrito en C. El arreglo "*a*" contiene números enteros de 32 bits y está almacenado en un espacio contiguo de la memoria principal, comenzando en la dirección 0.

```
int i, j, sum1;
for(i=0; i<100; i=i+1){
    for(j=0; j<=5120; j=j+1024){
        sum1 += a[j];
    }
}
```

- C) (0,8 pt.) Tradicionalmente, la indexación de una caché se realiza mediante la función módulo, de modo tal que para el bloque de memoria número "*d*", el índice de la caché se calcula como " $d \bmod 2^N$ ", con " $N$ " el número de bits para indexar la caché. Considere una indexación "Mersenne" que utiliza el número de Mersenne " $2^N - 1$ " para el índice, de modo que, en este caso, el índice se calcula como " $d \bmod (2^N - 1)$ ". Para el bucle mostrado arriba, determine las tasas de desacierto para ambos modos de indexación: tradicional y "Mersenne". Asuma que la caché está inicialmente vacía.

### Pregunta 3. Varios

#### Parte 1 - Rendimiento procesadores paralelos

$N^2$  escalares

Suma matriz  $M \times M$

Procesador Único  $\rightarrow$  Tiempo =  $(N^2 + M^2) \times t_{add}$

Aceleración =  $0,5P$

$$T_P = \left( N^2 + \frac{M^2}{P} \right) \times t_{add} \rightarrow \frac{N^2 + M^2}{N^2 + \frac{M^2}{P}} = 0,5P$$

$$N^2 + M^2 = 0,5P N^2 + 0,5 M^2$$

$$N^2 - 0,5P N^2 + M^2 - 0,5 M^2 = 0$$

$$N^2 (1 - 0,5P) + 0,5 M^2 = 0$$

$$0,5 M^2 = N^2 (0,5P - 1)$$

$$\frac{0,5}{0,5P - 1} = \frac{N^2}{M^2}$$

$$\frac{N}{M} = \sqrt{\frac{0,5}{0,5P - 1}}$$

//

0,8



## Parte 2

Tamaño de página  $\rightarrow 2^{13}$  bytes (8 KiB)

Por lo tanto, el número máximo de entradas que se requieren para la tabla de páginas es  $2^{42-13}$ :

$\rightarrow (2^{29} \text{ entradas})$

Memoria física para almacenar la tabla de páginas:

$$2^{29} \times 4 \text{ bytes} = 2^{31} \text{ bytes} = 2 \text{ GiB}$$

0,4

### Parte 3

Memoria principal  $\rightarrow 4 \text{ GiB} = 2^{32} \text{ B}$   
Cache asociativa de 4 vas (LRU)  
Direcciones de memoria de 32 bits  
9 bits de index  
Bloques de 8 bytes ( $2^3$ ) 2 palabras

→ más usado recientemente

Tradicional:  $d \bmod 2^9$

Mersenne:  $d \bmod 2^9 - 1$

El segundo for recorre los elementos:

0 1024 2048 3072 4096 5120

Los bloques son:

0 - 512 1024 1536 2048 2560

0  $\bmod 2^9 = 0$   
512  $\bmod 2^9 = 0$   
1024  $\bmod 2^9 = 0$   
1536  $\bmod 2^9 = 0$   
2048  $\bmod 2^9 = 0$   
2560  $\bmod 2^9 = 0$  ✓

0  $\bmod (2^9 - 1) = 0$   
512  $\bmod (2^9 - 1) = 1$   
1024  $\bmod (2^9 - 1) = 2$   
1536  $\bmod (2^9 - 1) = 3$   
2048  $\bmod (2^9 - 1) = 4$   
2560  $\bmod (2^9 - 1) = 5$  ✓

En la indexación tradicional en el primer ciclo del primer for se tienen 6 desaciertos, pues la caché inicia vacía. Y como hay sólo cuatro vías se tiene:

Conjunto 0				} Mapean al mismo conjunto (0)
0	512	1024	1536	
"	"	"	2048	
"	"	"	2560	

Como se utiliza reemplazo NRU, en el ciclo siguiente habrá 3 aciertos, pues ya está 0, 512 y 1024, mientras que los otros 3 serán desaciertos, por lo tanto la tasa de desaciertos es:

$$6 + 3 * 99 / 600 = 0,505 = 50,5\%$$

Por otro lado, en la indexación Mersenne no habrá ningún reemplazo, pues cada bloque de memoria mapea a un conjunto distinto de la caché, por lo tanto hay 6 desaciertos la primera vez que se acceden los elementos. Luego, como hay 600 iteraciones, se tiene que la tasa de desaciertos es  $6 / 600 = 0,01 = 1\%$ .

0,8