



IIC2343 - Arquitectura de Computadores (II/2025)

Guía de ejercicios: Caché y Coherencia de Caché

Ayudantes: Daniela Ríos (danielaarp@uc.cl), Alberto Maturana (alberto.maturana@uc.cl), Tomás López Massaro (tomas.lopezm20@uc.cl), Ignacio Gajardo (gajardo.ignacio@uc.cl)

Pregunta 1: Preguntas Conceptuales

- (a) [P1b | I3-2013-1] Describa las tres funciones de correspondencia vistas en clase.
- (b) [P1b | I3-2014-2] El algoritmo de reemplazo MRU (*Most Recently Used*), a diferencia de LRU (*Least Recently Used*), descarta primero los elementos que han sido ocupados más recientemente. ¿En qué casos podría ser útil el uso de este esquema?
- (c) [P2b | I3-2015-2] ¿Por qué es importante que algunos niveles de memoria *caché* sean exclusivos para cada núcleo y otros compartidos?
- (d) Dentro del contexto de múltiples procesadores, existen dos formas de que estos tengan acceso a una misma fuente de memoria: UMA (*Uniform Memory Access*) y NUMA (*Non-Uniform Memory Access*). ¿En qué consiste cada una de estas? Mencione, además, una ventaja y desventaja para cada una.
- (e) [I3-2018-1] Mencione una ventaja y una desventaja de actualizar las líneas de caché desactualizadas en vez de invalidarlas.
- (f) [EX-2018-1] El protocolo MOESI corresponde a una variante del protocolo MESI estudiado en clases. Este posee el mismo comportamiento, salvo por la existencia de un nuevo estado **O** (*Owned*). Este es utilizado para indicar que uno y solo uno de los controladores de caché tiene el permiso para modificar el valor de una línea de la caché asociada a un bloque de la memoria y compartirlo con el resto de las cachés que la posean. Esto permite que, en caso de que se hagan lecturas de un recurso compartido, la transferencia de datos se haga entre cachés y, además, no es necesario que se escriban los cambios directamente en la memoria principal, solo cuando es estrictamente necesario.
 - I. ¿Cómo se podría asegurar que sea una sola caché la que posea el estado **O** para un recurso compartido? Comente considerando el procedimiento del protocolo MESI para recursos compartidos (en particular, para los estados **M**, **S** e **I**).
 - II. Suponga que un par de cachés comparten una línea con estados **O** y **S**, respectivamente. Si la segunda caché, en estado **S**, solicita realizar cambios en la línea compartida, ¿qué protocolo se puede seguir para mantener la consistencia?

- (g) [P4d-Sec. 1 | I3-2024-1] Una variación del protocolo MESI de coherencia de caché es el protocolo MSI (sin el estado *Exclusive*). ¿Qué ventaja tiene tener el estado *Exclusive*? Y ¿cómo se hace cargo el protocolo MSI de no tener el estado *Exclusive*?

Pregunta 2: Evaluación de Jerarquía de Memoria [P1a | I3-2024-1]

En la jerarquía de memoria de un computador, cuenta con una memoria caché de 64KiB, líneas de 16 palabras y tiempo de acceso de 10ns. Por otra parte, cuenta con un *miss penalty* de 100ns. ¿Qué *hit-rate* debe tener la caché para obtener un tiempo de acceso promedio de 20ns en el computador?

Pregunta 3: Accesos a Memoria *Caché* [P1c | I3R-2024-1]

Suponga que posee una memoria caché de 16 líneas, 4 palabras por línea y una memoria principal de 128 bytes. Para cada una de las siguientes funciones de correspondencia: *directly mapped*, *fully associative*, *2-way associative* y *4-way associative*, indique para ella el *hit-rate* y el estado final de la caché (bits de *tag* y bit de validez) para los siguientes accesos de memoria: 0, 2, 1, 3, 96, 98, 97, 99, 24, 26, 25, 27. Asuma que se implementa una política de reemplazo LRU, si es que la requiere.

Pregunta 4: *Streaming* y *Caché* [P2c | I3-2015-2]

Las aplicaciones que reproducen audio o video son parte de un tipo de aplicaciones que generan un trabajo en la memoria del tipo *streaming*, que consiste en la lectura de grandes cantidades de datos, pero una muy baja reutilización de estos. Para este ejercicio considere un computador que corre una aplicación que realiza *streaming* para acceder a 512KB de datos. El patrón de acceso a memoria es el siguiente: 0, 4, 8, 12, 16, 20, ...

- (a) Asuma que el computador tiene una memoria caché de 64KB con función de correspondencia *directly mapped* y líneas de 32 bytes. ¿Cuál es el *miss-rate* para el patrón de accesos a memoria descrito anteriormente? ¿Cuán sensible es este *miss-rate* con respecto al tamaño de la memoria *caché*? ¿Mejora el *hit-rate* si se utiliza una *caché 4-way associative*?
- (b) Recalcule el *miss-rate* tomando los siguientes tamaños de línea en una *caché* que usa *directly mapped*: 16 bytes, 64 bytes y 128 bytes. ¿Qué tipo de localidad aprovecha este tipo de aplicaciones?
- (c) El *prefetching* es una técnica que saca provecho de patrones de lectura predecibles para traer desde la memoria, de manera predictiva, bloques de memoria adicionales cuando una línea de la *caché* en particular es accedida.

Un ejemplo es el *stream buffering*, que consiste en copiar en un *buffer* separado bloques adyacentes a la línea de la *caché* accedida. Si el dato se encuentra en este *buffer*, se considera como *hit* y es copiado a la *caché*, mientras que el espacio que este dato usaba en el *buffer*, es llenado con un nuevo bloque adyacente.

Asuma que el computador cuenta con un *stream buffer* de dos entradas y que la latencia de la *caché* es tal que una línea puede ser cargada antes de que el cálculo realizado sobre la línea anterior ha sido completado. En base a esto y suponiendo una *caché* con líneas de 32 bytes, ¿cuál es el *hit-rate* para el patrón de acceso a memoria del apartado anterior?

Pregunta 5: Protocolo MESI [P4-Sec. 3 | I3-2024-1]

Considere la siguiente memoria. Esta es compartida por 3 CPUs con paralelismo UMA. Para mantener la coherencia entre sus caché, se usa el protocolo MESI.

Memoria compartida

Dir	Label	Valor
0	x	1
1	y	2
2	arr	3
3		4
4		5

CPU 0

MOV B, arr
INC B
MOV A, (B)
MOV (x), A
MOV A, 3
MOV B, (4)
ADD A, B
MOV (4), A

CPU 1

MOV A, (x)
MOV B, (y)
ADD A, B
MOV (3), A
MOV A, 2
MOV B, (x)
MOV (3), B
ADD A, B

CPU 2

MOV B, arr
MOV A, 0
ADD A, (B)
INC B
ADD A, (B)
INC B
MOV (y), A
MOV B, (3)

Puede asumir que las líneas de la caché son de una palabra y todas las líneas de las caché privadas parten en estado inválido. Complete las siguientes tablas a partir de la ejecución de los programas en las tres CPU. En cada entrada, indique si el valor fue **modificado** (M), es **exclusivo** (E), **compartido** (S) o **inválido** (I).

Caché - CPU 0					
Inst	x	y	arr[0]	arr[1]	arr[2]
1					
2					
3					
4					
5					
6					
7					
8					

Caché - CPU 1					
Inst	x	y	arr[0]	arr[1]	arr[2]
1					
2					
3					
4					
5					
6					
7					
8					

Caché - CPU 2					
Inst	x	y	arr[0]	arr[1]	arr[2]
1					
2					
3					
4					
5					
6					
7					
8					

Pregunta 6: Protocolo MESI [P6b | EXR-2023-1]

Suponga que posee una arquitectura MIMD de memoria compartida con 2 CPU que poseen su propia caché y que siguen el protocolo MESI para asegurar consistencia. La memoria contiene los siguientes datos de variables y las CPU0 y CPU1 ejecutan los siguientes programas:

Dirección	Label	Valor
0x00	i	1
0x01	arr	1
0x02		3
0x03		2

```
// CPU0      // CPU1
MOV B,(i)    MOV A,2
MOV A,(arr)  MOV B,arr
SUB B,A      ADD B,A
SHL A,A      MOV B,(B)
MOV (B),A    MOV (B),A
MOV B,A      MOV B,(B)
MOV B,(B)    NOP
```

Asumiendo que cada dirección se almacena en una línea distinta, indique el estado de cada línea de las caché (M/E/S/I) para cada iteración completando la tabla adjunta, asumiendo que todas las líneas parten en estado I:

Instrucción	CPU0				CPU1			
	i	arr[0]	arr[1]	arr[2]	i	arr[0]	arr[1]	arr[2]
1								
2								
3								
4								
5								
6								
7								

Pregunta 7: Protocolo MESI [P4-Sec. 2,4 | I3-2024-1]

Suponga que posee una arquitectura MIMD de memoria compartida con 3 CPU que poseen su propia caché y que siguen el protocolo MESI para asegurar consistencia. La memoria contiene los siguientes datos de variables; mientras que CPU0, CPU1 y CPU2 ejecutan los siguientes programas:

Dirección	<i>Label</i>	Valor
0x00	arr	2
0x01		0
0x02		1
...
0xFE		0
0xFF		0

// CPU0	// CPU1	// CPU2
MOV B,arr	MOV A,2	MOV A,5
INC B	PUSH A	MOV B,1
MOV B,(B)	POP A	PUSH A
MOV (B),B		PUSH B

Asumiendo que cada dirección se almacena en una línea distinta, y que el contador SP posee un valor inicial igual a 255 en **todas** las CPU, indique el estado de cada línea de las caché (M/E/S/I) para cada ciclo completando la tabla adjunta, asumiendo que todas las líneas parten en estado I (ciclo 0):

[illegible]