



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IIC2343 – Arquitectura de Computadores

Ayudantía 7

Profesor: Yadran Francisco Eterovic Solano

Ayudante: Germán Leandro Contreras Sagredo (glcontreras@uc.cl)

Temas a tratar

Los temas a tratar dentro de esta ayudantía son:

- Coherencia de *caché*.

Preguntas

- a. 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.
- b. ¿En qué consiste el mecanismo de consistencia de “*caché snooping*”?
- c. Existe otro mecanismo de consistencia de *caché* conocido como coherencia basada en directorio. En este, existen uno o varios directorios en la memoria que mantienen registro de los elementos compartidos por las *cachés* y sus estados. Tiene la ventaja de ser escalable, pero la desventaja de que puede producir *overhead* por el contenido que no se busca acceder dentro de este. En base a esto, ¿qué ventaja y desventaja posee el mecanismo de *snooping* si se compara con el basado en directorio?
- d. Explique el protocolo *write through* de consistencia de *cachés*. ¿Qué otras variaciones existen?
- e. Mencione y explique cada estado del protocolo *write-back* MESI.

2. a. **(IIC2523 - I1 - II/2015)** Una arquitectura UMA, aún cuando es más sencilla, es siempre más eficiente que una arquitectura NUMA. ¿Es esta sentencia verdadera o falsa? Justifique.
- b. **(Examen - II/2018)** En un multiprocesador (varias CPUs que comparten una memoria común a través de un bus), cada CPU puede tener su propia memoria *caché* para así evitar tener que recurrir frecuentemente a la memoria principal a través del bus. Sin embargo, esto normalmente da origen al problema de coherencia de *cachés*. Para solucionar estos problemas, los controladores de *cachés* son diseñados de manera que “observen” (*snoop*) las solicitudes que pasan por el bus (y que fueron hechas por alguna otra *caché*) y que hagan algo en ciertos casos. El conjunto de reglas que define qué hacer y cuándo se llama protocolo de consistencia de *cachés*.
- I. Considera el protocolo *write through* estudiado en clase, cuya esencia es que todas las operaciones de escritura resultan en que la palabra que está siendo escrita en la *caché* también es escrita en la memoria para mantener la memoria actualizada todo el tiempo. Si el controlador de *caché* solo pudiera observar las líneas de dirección del bus, y no las de datos, ¿se vería el protocolo *write through* afectado por esta situación? Explica.
- II. En los protocolos de tipo *write-back* no todas las escrituras van directamente a la memoria: cuando una línea de la *caché* es modificada, se pone un bit de la *caché* en 1 indicando que la línea de la *caché* está correcta pero la memoria no; finalmente, la línea es escrita en la memoria, pero posiblemente después de sufrir varias escrituras. El protocolo MESI estudiado en clase define cuatro estados para cada línea de la *caché*: *modified*, *exclusive*, *shared* (compartido), *invalid*. Si solo pudiéramos tener tres estados, ¿cuáles estados podrían ser eliminados (solo uno a la vez) y cuáles serían las consecuencias en cada caso?