



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

IIC2343 – Arquitectura de Computadores

## Ayudantía 7

Profesores: Hans-Albert Löbel Díaz, Jurgen Dieter Heysen Palacios

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

### Temas a tratar

Los temas a tratar dentro de esta ayudantía son:

- Jerarquía de Memoria.

### Preguntas

- Explique los principios de localidad espacial y localidad temporal con sus propias palabras, dando un ejemplo para cada caso.
  - (I3 - I/2017) Sin considerar el precio, ¿por qué no tiene sentido usar una *caché* infinita?
  - Suponga que tiene una memoria principal de 16 bytes y una memoria *caché* de 8 bytes y 4 líneas. Además, asuma que tiene un programa que accede, en este orden, a las direcciones de la memoria principal: 0, 1, 5, 7, 10, 13, 4, 6.  
Obtenga el estado final de la memoria *caché* (en una tabla) y el *hit-rate* para cada una de las siguientes funciones de correspondencia:
    - Directly mapped*.
    - Fully associative*.
    - 2-way associative*.Puede asumir una política de reemplazo LRU, en caso de necesitarla.
  - (I3 - I/2016) Comente sobre las ventajas de tener una memoria *caché split* en vez de una *unified* conectada a la memoria de datos del computador básico.
  - (I2 - I/2018) Suponga que tiene una *caché* con *mapeo* directo. ¿Se demora más esta en sustituir una línea que una *caché fully associative*? ¿Por qué?

2. a. **(I3 - I/2016)** Un computador tiene una memoria *caché* de 16KB, con líneas de 32 bytes que almacenan 8 palabras, y un tiempo de acceso de 10ns. La memoria *caché* está conectada a la memoria principal mediante un bus capaz de transferir 8 bytes en 120ns. ¿Cuál es el *hit-rate* que debe tener la memoria *caché* para tener un tiempo de acceso promedio de 20ns?
- b. **(I3 - I/2017)** Considere una memoria *caché fully-associative*, con *hit-time* igual a  $16L^3 - 100L$  ns, donde  $L$  es la cantidad de líneas de la *caché*. Si el *hit-rate* de esta memoria es de 0.95, ¿cuál es la cantidad de líneas que genera el tiempo de acceso promedio mínimo?
- c. **(I3 - I/2015)** Un computador de 64 bits tiene una memoria *caché* de 32KB, con 1024 líneas de 32 palabras. ¿Cuánto espacio de esta *caché* es usado por información distinta de los datos?
- d. **(I3 - I/2015)** Considere un computador con microarquitectura Von Neumann, donde la tasa de ciclos de *clock* por instrucción es igual a  $N$ , cuando todos los accesos a memoria producen *hits* en la *caché*. La memoria *caché* tiene *miss-rate* de 4% y *miss-penalty* de  $25 \times N$  ciclos de *clock*. Si en un programa de  $K$  instrucciones, el 50% de estas realizan lectura de un dato en memoria, ¿cuántos ciclos de *clock* menos tomaría la ejecución del programa, si todas las instrucciones produjeran *hits* en la memoria *caché*?
3. a. **(I3 - I/2013)** El principio de localidad espacial explica en parte el buen funcionamiento de la memoria *caché*. Sin embargo, es posible no cumplir este principio, disminuyendo el rendimiento de la memoria. Describa un ejemplo específico de esto y explique por qué se produce.
- b. **(I3 - I/2015)** ¿Cómo es el rendimiento de una memoria *caché*, si el patrón de accesos a memoria distribuye de manera uniforme sobre todas las posibles direcciones? Ejemplifique el o los posibles casos.
- c. **(I3 - I/2017)** La contención de bloques es un problema del esquema de *mapeo* directo, donde 2 o más bloques pelean por la misma línea, existiendo otras líneas no utilizadas en la *caché*. ¿Existe un problema similar en el esquema *N-way*? Si su respuesta es negativa, justifíquela y, si es positiva, indique detalladamente un caso en que esto se de.
- d. **(I3 - II/2014)** El algoritmo de reemplazo MRU (*Most Recently Used*), a diferencia de LRU, descarta primero los elementos que han sido ocupados más recientemente. ¿En qué casos podría ser útil el uso de este esquema?
- e. **(I3 - II/2014)** Describa al menos dos posibles soluciones para el problema de consistencia de memoria que se genera al tener un esquema de escritura de *caché write-back*.