



IIC2343 - Arquitectura de Computadores (I/2025)

Ayudantía 10 - Caché

Ayudantes: Daniela Ríos (danielaarp@uc.cl), Joaquín Peralta (jperaltaperez@uc.cl), Mario Rojas (mario.denzel@estudiante.uc.cl)

Pregunta 1: Preguntas teóricas

Responda las siguientes preguntas:

- (a) 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?

**Solución:** Para obtener el tiempo de acceso promedio en una memoria, se utiliza la siguiente fórmula:

$$TP = HR \times HT + (1 - HR) \times (HT + MP)$$

Siendo  $TP$  el tiempo de acceso promedio;  $HT$  el *hit-time* promedio;  $HR$  el *hit-rate* promedio; y  $MP$  el *miss penalty*. Luego, reemplazando valores:

$$20\text{ns} = HR \times 10\text{ns} + (1 - HR) \times (10 + 100)$$

$$20\text{ns} = 110 - 100HR$$

$$HR = \frac{90}{100}$$

$$HR = 0,9$$

- (b) Suponga que posee una memoria caché significativamente pequeña y rápida, por lo que los tiempos de búsqueda en ella son bajos. ¿Qué función de correspondencia es la más adecuada para optimizar su uso? Justifique su respuesta.

**Solución:** Para estos casos conviene el uso de la función de correspondencia *fully associative*. Dado que el tiempo de búsqueda es bajo por el tamaño de la caché, es preferible aprovechar al máximo su espacio. Un ejemplo puntual de esto es la TLB, que es *fully associative* por su tamaño y rapidez. Se otorgan **0.5 pts.** por indicar correctamente el tipo de función de correspondencia y **0.5 pts.** por justificación.

- (c) Señale qué tipo de ganancia tendría el computador básico visto en el curso si a la memoria de datos (RAM) se le agregara una caché de tipo *split*.

**Solución:** Existe solo una ganancia marginal, dado que la caché *split* es idónea para memorias de tipo Von Neumann, *i.e.* que almacenan tanto datos como instrucciones. La caché ideal para la RAM del computador básico es de tipo *unified*. Se otorga **1 pto.** si se menciona de alguna manera la desventaja de usar una caché *split* en este contexto.

- (d) ¿Qué incidencia tiene la política de reemplazo en una caché con función de correspondencia *directly mapped*? Justifique.

**Solución:** No tiene ninguna incidencia. En la función de correspondencia *directly mapped*, los reemplazos solo se realizan sobre una línea que ya se encuentra ocupada contenido con un bloque de memoria distinto al que se busca acceder. Por este motivo, no se utiliza ningún criterio de elección de línea a ser reemplazada. Se otorgan **0.5 pts.** por responder correctamente; y **0.5 pts.** por justificación.

## Pregunta 2: Caché: Pregunta práctica

- (a) Suponga que, en un instante dado, posee el siguiente estado en una caché de 8 líneas y 2 palabras por línea, cada una de 1 byte:

Línea	Validez	Tag
0	1	5
1	1	15
2	0	9
3	1	1
4	1	3
5	1	9
6	1	5
7	1	8

La columna “Línea” representa el índice de línea; “Validez” representa el *valid bit*; y “Tag” representa el valor, en base decimal, del *tag* que indica el bloque de memoria almacenado en la línea. Con este estado de caché, se realiza el acceso a memoria a las siguientes direcciones: 0x8F, 0x17, 0x11, 0x3D, 0xF5, 0x16. Indique, para cada acceso, si existe un *hit* o *miss*; si corresponde, la línea de la caché que es modificada; y el *hit-rate* para la función de correspondencia Directly mapped.

Asuma que posee una memoria principal de 256 palabras de 1 byte. Para responder, complete la tabla adjunta al enunciado. El *tag* de cada línea lo puede escribir en base binaria o decimal, pero sí debe considerar su valor en caso de reemplazos.

**Accesos *Directly mapped*:**

N° Acceso	Dirección	Tag	Hit/Miss	ID Línea modificada	Comentarios
0	0x8F				
1	0x17				
2	0x11				
3	0x3D				
4	0xF5				
5	0x16				

***Hit-rate:***

**Solución:** Independiente de la función de correspondencia, al tener una memoria de 256 palabras se necesitan  $\log_2(256) = 8$  bits para cada dirección. Por otra parte, al ser las líneas de 2 palabras, se requiere  $\log_2(2) = 1$  bit de *offset* para ubicar una palabra dentro de una línea. Con esto en consideración, se tiene el siguiente resultado para cada secuencia de accesos según función:

1. *Directly mapped*: Al tener 8 líneas, se necesitan  $\log_2(8) = 3$  bits de índice de línea, resultando en  $8 - 3 - 1 = 4$  bits de *tag*. Por otra parte, el *timestamp* es irrelevante ya que no aplican las políticas de reemplazo en esta función, si el contenido no está en la línea a la que pertenece, se reemplaza directamente. A continuación, la tabla de accesos resultante:

N° Acceso	Dirección	Tag	Hit/Miss	ID Línea modificada	Comentarios
0	0x8F	1000b = 8 = 0x8	Hit	-	Índice de línea igual a 111b = 7. El bit de validez es 1 y coinciden los <i>tags</i> .
1	0x17	0001b = 1 = 0x1	Hit	-	Índice de línea igual a 011b = 3. El bit de validez es 1 y coinciden los <i>tags</i> .
2	0x11	0001b = 1 = 0x1	Miss	000b = 0	Índice de línea igual a 000b = 0. El bit de validez es 1 pero no coinciden los <i>tags</i> .
3	0x3D	0011b = 3 = 0x3	Miss	110b = 6	Índice de línea igual a 110b = 6. El bit de validez es 1 pero no coinciden los <i>tags</i> .
4	0xF5	1111b = 15 = 0xF	Miss	010b = 2	Índice de línea igual a 010b = 2. El bit de validez es 0, por lo que el contenido buscado no está.
5	0x16	0001b = 1 = 0x1	Hit	-	Índice de línea igual a 011b = 3. El bit de validez es 1 y coinciden los <i>tags</i> .

**Hit-rate:**  $\frac{3}{6} = 0,5 = 50 \%$

## 1. Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.

