

IIC2343 – Arquitectura de Computadores

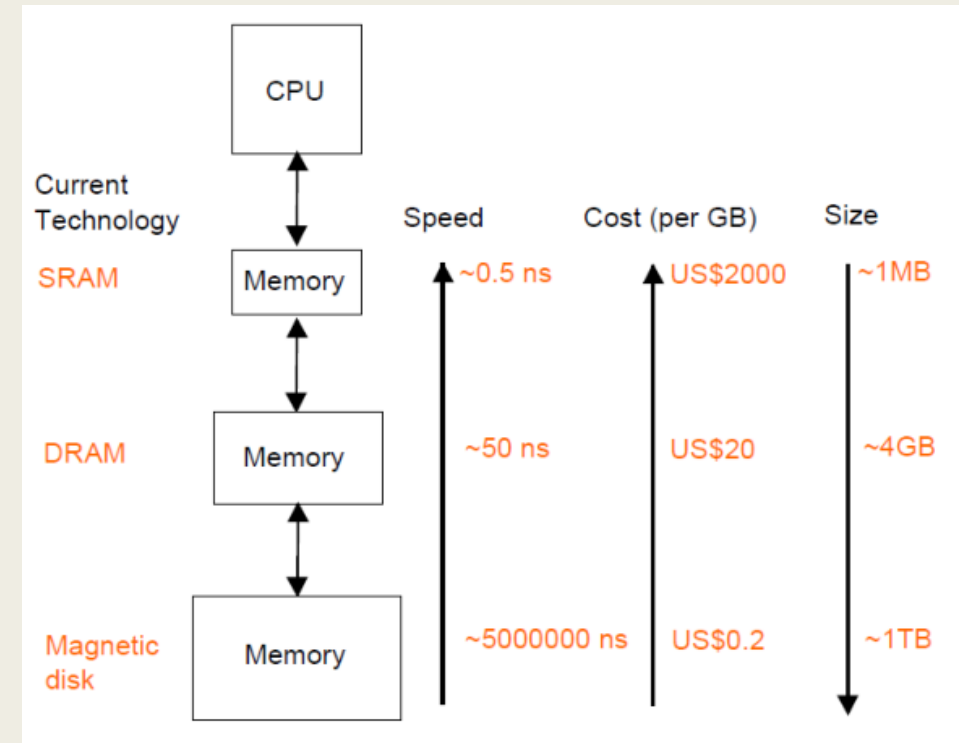
AYUDANTÍA 7

Jerarquía de Memoria y Memoria Caché

Constanza Osorio

Jerarquía de Memoria (*´∀`).。 00

- Existe un *trade-off* entre velocidad y capacidad de memoria para cierto costo fijo, por lo que debemos organizar nuestra memoria de forma más eficiente.
- La **jerarquía de memoria** se basa en usar múltiples niveles de memoria, en el que cada nivel es un tipo distinto de memoria. De esta manera podemos aprovechar las ventajas de cada tecnología.
- El primer nivel será el más rápido, pero de poca capacidad, mientras que el último nivel será de mucho más capacidad pero más lento.



Principio de Localidad

- **Principio de Localidad Temporal:** Un dato recientemente obtenido de memoria, es muy probable que vuelva a ser usado en el corto plazo.
- **Principio de Localidad Espacial:** Si un dato se necesita, es muy probable que los datos ubicados en posiciones cercanas a él también se van a necesitar en el corto plazo.
- Teniendo en cuenta estos principios, cuando la CPU necesite un dato y lo vaya a buscar al último nivel (el más lento), se traiga además los datos contiguos a éste, y los vaya copiando en los niveles superiores de memoria.

(o ` ε´ o) / \ (o ` ε´ o)

Función de Correspondencia

- Tres tipos:

- *Directly Mapped*: Cada bloque en memoria principal se encuentra mapeada a una línea de caché específica.
- *Fully Associative*: Cada bloque de memoria se puede asociar a cualquier línea de la caché.
- *N-Way Associative*: Cada bloque de memoria está asociado a un conjunto (mapeo directo) y dentro de ese conjunto pueden ubicarse en cualquiera de sus líneas.

Políticas de reemplazo

- Existen diversos tipos de políticas de reemplazo, pero nos centraremos en los siguientes:
 - **FIFO:** *First In First Out*. Se reemplaza línea que se escribió hace más tiempo
 - **LFU:** *Least Frequently Used*. Se reemplaza la línea que se ha accedido menos veces hasta ese momento.
 - **LRU:** *Least Recently Used*. Se reemplaza la línea que se accedió hace más tiempo.
 - **Random:** Reemplazo aleatorio. $\backslash(\text{ツ})/_$
 - **Bélády:** Se reemplaza la línea que se utilizará más lejos en el futuro. Es inalcanzable en la práctica.

Políticas de Escritura

- **Write-Through:** Cada escritura en caché se actualiza inmediatamente en la memoria principal. No se generan inconsistencias entre la memoria caché y la memoria principal, sin embargo, involucra una espera considerable para la CPU (ahora debe esperar a que se escriba en ambas memorias).
- **Write-Back:** Sólo se hará escritura en caché en un principio, y se escribirá en memoria sólo al momento de reemplazar la línea de caché modificada. Es más eficiente, pero puede generar errores de consistencia.

(a) Un programa presenta los siguientes accesos a memoria:

4 20 3 17 20 27 20

Asuma que tiene una caché de **8 líneas**, de **2 palabras** cada una. Muestre el estado final de la caché con:

- Función de correspondencia Directly Mapped.
- Función de correspondencia 4-Way Associative.

EJERCICIO 1 (a)

Directly Mapped

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0		
	1		
001	0		
	1		
010	0		
	1		
011	0		
	1		
100	0		
	1		
101	0		
	1		
110	0		
	1		
111	0		
	1		

Directly Mapped

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0		
	1		
001	0		
	1		
010	0	MEM[4]	0000
	1	MEM[5]	0000
011	0		
	1		
100	0		
	1		
101	0		
	1		
110	0		
	1		
111	0		
	1		

Directly Mapped

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0		
	1		
001	0		
	1		
010	0	MEM[20]	0001
	1	MEM[21]	0001
011	0		
	1		
100	0		
	1		
101	0		
	1		
110	0		
	1		
111	0		
	1		

Directly Mapped

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0	MEM[16]	0001
	1	MEM[17]	0001
001	0		
	1		
010	0	MEM[20]	0001
	1	MEM[21]	0001
011	0		
	1		
100	0		
	1		
101	0		
	1		
110	0		
	1		
111	0		
	1		

Directly Mapped

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0	MEM[16]	0001
	1	MEM[17]	0001
001	0		
	1		
010	0	MEM[20]	0001
	1	MEM[21]	0001
011	0		
	1		
100	0		
	1		
101	0		
	1		
110	0		
	1		
111	0		
	1		

Directly Mapped

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0	MEM[16]	0001
	1	MEM[17]	0001
001	0		
	1		
010	0	MEM[20]	0001
	1	MEM[21]	0001
011	0		
	1		
100	0		
	1		
101	0	MEM[26]	0001
	1	MEM[27]	0001
110	0		
	1		
111	0		
	1		

Directly Mapped

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0	MEM[16]	0001
	1	MEM[17]	0001
001	0		
	1		
010	0	MEM[20]	0001
	1	MEM[21]	0001
011	0		
	1		
100	0		
	1		
101	0	MEM[26]	0001
	1	MEM[27]	0001
110	0		
	1		
111	0		
	1		

Directly Mapped

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0	MEM[16]	0001
	1	MEM[17]	0001
001	0		
	1		
010	0	MEM[20]	0001
	1	MEM[21]	0001
011	0		
	1		
100	0		
	1		
101	0	MEM[26]	0001
	1	MEM[27]	0001
110	0		
	1		
111	0		
	1		

4-way Associative

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Set	Índice Línea	Posición Palabra	Dato	Tag
0	000	0	MEM[4]	000001
		1	MEM[5]	000001
	001	0		
		1		
	010	0		
		1		
	011	0		
		1		
1	100	0		
		1		
	101	0		
		1		
	110	0		
		1		
	111	0		
		1		

4-way Associative

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Set	Índice Línea	Posición Palabra	Dato	Tag
0	000	0	MEM[4]	000001
		1	MEM[5]	000001
	001	0	MEM[20]	000101
		1	MEM[21]	000101
	010	0		
		1		
	011	0		
		1		
1	100	0		
		1		
	101	0		
		1		
	110	0		
		1		
	111	0		
		1		

4-way Associative

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Set	Índice Línea	Posición Palabra	Dato	Tag
0	000	0	MEM[4]	000001
		1	MEM[5]	000001
	001	0	MEM[20]	000101
		1	MEM[21]	000101
	010	0	MEM[16]	000100
		1	MEM[17]	000100
	011	0		
		1		
1	100	0		
		1		
	101	0		
		1		
	110	0		
		1		
	111	0		
		1		

4-way Associative

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Set	Índice Línea	Posición Palabra	Dato	Tag
0	000	0	MEM[4]	000001
		1	MEM[5]	000001
	001	0	MEM[20]	000101
		1	MEM[21]	000101
	010	0	MEM[16]	000100
		1	MEM[17]	000100
	011	0		
		1		
1	100	0		
		1		
	101	0		
		1		
	110	0		
		1		
	111	0		
		1		

4-way Associative

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Set	Índice Línea	Posición Palabra	Dato	Tag
0	000	0	MEM[4]	000001
		1	MEM[5]	000001
	001	0	MEM[20]	000101
		1	MEM[21]	000101
	010	0	MEM[16]	000100
		1	MEM[17]	000100
	011	0		
		1		
1	100	0	MEM[26]	000110
		1	MEM[27]	000110
	101	0		
		1		
	110	0		
		1		
	111	0		
		1		

4-way Associative

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Set	Índice Línea	Posición Palabra	Dato	Tag
0	000	0	MEM[4]	000001
		1	MEM[5]	000001
	001	0	MEM[20]	000101
		1	MEM[21]	000101
	010	0	MEM[16]	000100
		1	MEM[17]	000100
	011	0		
		1		
1	100	0	MEM[26]	000110
		1	MEM[27]	000110
	101	0		
		1		
	110	0		
		1		
	111	0		
		1		

4-way Associative

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Set	Índice Línea	Posición Palabra	Dato	Tag
0	000	0	MEM[4]	000001
		1	MEM[5]	000001
	001	0	MEM[20]	000101
		1	MEM[21]	000101
	010	0	MEM[16]	000100
		1	MEM[17]	000100
	011	0		
		1		
1	100	0	MEM[26]	000110
		1	MEM[27]	000110
	101	0		
		1		
	110	0		
		1		
	111	0		
		1		

(b) Explique las principales diferencias que se podrían haber encontrado si hubiesemos tenido una caché *fully assosiative* en vez de una 4-way associative.

EJERCICIO 1 (b)

Fully Associative

Cache de 8 líneas, de 2 palabras cada una.

Nro	Binario
4	00000100
20	00010100
17	00010001
20	00010100
27	00011011
20	00010100

Índice Línea	Posición Palabra	Dato	Tag
000	0	MEM[4]	0000010
	1	MEM[5]	0000010
001	0	MEM[20]	0001010
	1	MEM[21]	0001010
010	0	MEM[16]	0001000
	1	MEM[17]	0001000
011	0	MEM[26]	0001101
	1	MEM[27]	0001101
100	0		
	1		
101	0		
	1		
110	0		
	1		
111	0		
	1		

[EX 2020 - 1] Dada una memoria caché *fully associative* de **4 líneas** y **2 palabras** cada una. Si la memoria principal es de 32 bytes, determine qué política de reemplazo se está utilizando (FIFO, LFU, LRU o random). Considere los siguientes accesos a memoria:

0, 1, 2, 3, 4, 14, 15, 28, 29, 4, 15, 5, 4, 3, 2, 1.

Y el estado de la caché tras cada acceso:

Nº Fila	Dirección	Binario	Línea 0	Línea 1	Línea 2	Línea 3	
1	0	00000	0 1	- -	- -	- -	Miss
2	1	00001	0 1	- -	- -	- -	Hit
3	2	00010	0 1	2 3	- -	- -	Miss
4	3	00011	0 1	2 3	- -	- -	Hit
5	4	00100	0 1	2 3	4 5	- -	Miss
6	14	01110	0 1	2 3	4 5	14 15	Miss
7	15	01111	0 1	2 3	4 5	14 15	Hit
8	28	11100	0 1	2 3	28 29	14 15	Miss
9	29	11101	0 1	2 3	28 29	14 15	Hit
10	4	00100	0 1	2 3	28 29	4 5	Miss
11	15	01111	0 1	2 3	14 15	4 5	Miss
12	5	00101	0 1	2 3	14 15	4 5	Hit
13	4	00100	0 1	2 3	14 15	4 5	Hit
14	3	00011	0 1	2 3	14 15	4 5	Hit
15	2	00010	0 1	2 3	14 15	4 5	Hit
16	1	00001	0 1	2 3	14 15	4 5	Hit

EJERCICIO 2

Acceso	Binario	Línea 0		Línea 1		Línea 2		Línea 3		H/M	Análisis
0	00000	0	1	-	-	-	-	-	-	Miss	Se llena normal FA.
1	00001	0	1	-	-	-	-	-	-	Hit	Hit.
2	00010	0	1	2	3	-	-	-	-	Miss	Se llena normal FA.
3	00011	0	1	2	3	-	-	-	-	Hit	Hit.
4	00100	0	1	2	3	4	5	-	-	Miss	Se llena normal FA.
14	01110	0	1	2	3	4	5	14	15	Miss	Se llena normal FA.
15	01111	0	1	2	3	4	5	14	15	Hit	Hit.
28	11100	0	1	2	3	28	29	14	15	Miss	Se reemplaza usando LFU; descartamos FIFO y LRU.
29	11101	0	1	2	3	28	29	14	15	Hit	Hit.
4	00100	0	1	2	3	28	29	4	5	Miss	Reemplazo no tiene sentido con ningún tipo de desempate; es RANDOM.
15	01111	0	1	2	3	14	15	4	5	Miss	Reemplazo no tiene sentido con ningún tipo de desempate; es RANDOM.
5	00101	0	1	2	3	14	15	4	5	Hit	Hit.
4	00100	0	1	2	3	14	15	4	5	Hit	Hit.
3	00011	0	1	2	3	14	15	4	5	Hit	Hit.
2	00010	0	1	2	3	14	15	4	5	Hit	Hit.
1	00001	0	1	2	3	14	15	4	5	Hit	Hit.