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

IIC2343 - Arquitectura de Computadores (II/2020)

Ayudantía 7 - Jerarquía de memoria y Memoria Caché

2 de Noviembre de 2020

1. Funciones de Correspondencia

(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.

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

Solución:

(a) Podemos ver la Tabla para Directly Mapped a la izquierda, y a la derecha la Tabla para 4-Way Associative.

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		

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) Tabla final de caché con fully associative.

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		

Podemos notar que, a diferencia de la caché 4-way associative, aquí vamos ocupando las líneas de la caché en orden. También notamos que aquí ya no tenemos bits para distinguir el conjunto, y por ende también nuestro tag crece. Además, ahora podemos hacer mejor uso de nuestro espacio en la caché porque no estamos dejando ningún espacio libre entremedio.
Otras cosas a considerar son el hecho de que una caché Fully Associative es más compleja a nivel de hardware que una N-Way Associative (Que vendría siendo un punto medio entre Directly Mapped y N-Way Associative) ya que ahora el costo de buscar una dirección en la caché es más alto, por lo que debemos compensar en hardware.

2. Políticas de reemplazo

[**EX 2020 - 1**]Dada una memoria caché *fully associative* de **4 lineas** 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:

N° 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

Solución:

Acceso	Binario	Línea 0		Línea 1		Línea 2		Línea 3		H/M	Análisis
0	00000	0	1	-	-	-	-	-	-	Miss	*Se llena norrmal FA.*
1	00001	0	1	-	-	-	-	-	-	Hit	Hit.
2	00010	0	1	2	3	-	-	-	-	Miss	*Se llena norrmal FA.*
3	00011	0	1	2	3	-	-	-	-	Hit	Hit.
4	00100	0	1	2	3	4	5	-	-	Miss	*Se llena norrmal FA.*
14	01110	0	1	2	3	4	5	14	15	Miss	*Se llena norrmal 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.

Figura 1: Análisis de la tabla

3. Políticas de Escritura

[**I3 2014 - 2**] Describa dos posibles soluciones para el problema de consistencia de memoria que se genera al tener un esquema de escritura de caché *write-back*.

Solución:

Solución 1: Tener un bit dentro de la memoria principal que indique si el dato fue modificado dentro de la caché o no (*dirty bit*). Entonces, al querer revisar dicha posición, si se tiene un *dirty bit* igual a 1, se puede realizar una interrupción que realice una sobreescritura en la memoria principal desde la caché, devolviéndole al bit su estado original igual a 0. De esta forma, la inconsistencia se arreglaría solo cuando fuera necesario
Solución 2: En general, tenemos problemas cuando otros dispositivos tratan de acceder de forma directa a la memoria (DMA). Entonces, podemos hacer que la DMA interactúe de forma directa con la caché, por lo que siempre tendría datos consistentes (sin embargo, en caso de necesitar datos almacenados en la memoria principal, sería necesario aplicar políticas de reemplazo).