

Tema 6. Memòria Cache Problemes

Joan Manuel Parcerisa



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona



Test: Veritat o Fals?

1. En una memòria cache amb política d'escriptura immediata sense assignació, un accés a la memòria cache pot implicar dos accesos a memòria principal
2. Si en una cache canviem la política d'escriptura immediata amb assignació a retardada amb assignació, sense cap més canvi, el nombre total de fallades no canvia
3. En un processador amb adreces de 32 bits, una cache associativa de 4 vies, de 32KB i blocs de 32 bytes, s'han de dedicar 19 bits a etiqueta (tag), 8 al número d'entrada (l'índex de conjunt) i 5 a l'offset

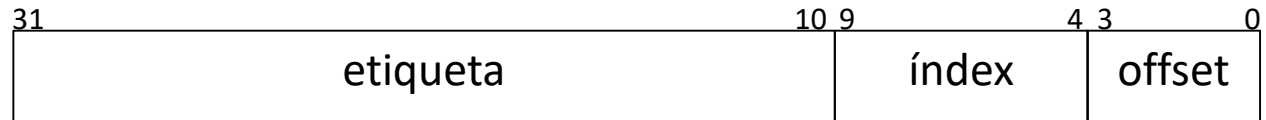
Test: Veritat o Fals?

4. Un sistema que tingués una memòria cache més gran que la memòria principal mai tindria fallades.
5. En les memòries cache que utilitzen escriptura immediata s'ha de posar el dirty bit a 1 només quan hi ha un encert d'escriptura.
6. El temps de penalització en cas de fallada d'escriptura d'una memòria cache d'escriptura immediata sense assignació és zero segons el model estudiat.
7. Si s'executa un mateix programa en dues memòries cache diferents, de correspondència directa i d'igual capacitat, sempre tindrà una major taxa d'encerts la que tingui els blocs de mida més gran.

Exercici 1 (a,b,c)

Problema:

Suposem un processador amb paraules i adreces de 32 bits. I una cache de dades de *correspondència directa, d'escriptura immediata sense assignació*, amb la següent distribució dels bits d'adreça:



- a) Quina és la mida del bloc, en bytes?.....
- b) Quantes línies té la cache?.....
- c) Quants bits d'emmagatzematge fan falta en total, per cada línia, incloent-hi les dades?.....

Exercici 1 (d)

d) Omple la següent taula suposant que la cache és inicialment buida, i fem la següent seqüència de referències (lectures i escriptures), indicant també al final quina és la taxa d'encert:

adreça	L/E	índex (hex)	Etiqueta (hex)	Encert (S/N)	Etiqueta bloc reemplaçat (hex)	Núm. bytes llegits a MP	Núm bytes escrits a MP
0x0000 0000	L						
0x0000 0004	L						
0x0000 004C	E						
0x0000 0404	E						
0x0000 0048	L						
0x0000 0008	E						
0x0000 0400	L						
0x0000 0044	L						

Taxa d'encert	
---------------	--

Exercici 1 (e)

e) Omple la mateixa taula, suposant que la cache és inicialment buida, però ara suposant que la cache té una política d'*escriptura retardada amb assignació*

adreça	L/E	índex (hex)	Etiqueta (hex)	Encert (S/N)	Etiqueta bloc reemplaçat (hex)	Núm. bytes llegits a MP	Núm bytes escrits a MP
0x0000 0000	L						
0x0000 0004	L						
0x0000 004C	E						
0x0000 0404	E						
0x0000 0048	L						
0x0000 0008	E						
0x0000 0400	L						
0x0000 0044	L						

Taxa d'encert	
---------------	--

Exercici 2

Suposem un sistema computador amb un processador MIPS de 32 bits. Suposem que té una cache de dades de *correspondència directa* amb *escriptura immediata sense assignació*, amb 32 línies de 8 bytes cada una, i que està inicialment buida. Calcula els encerts i fallades en tots els accessos a memòria a les variables A, B i C del següent programa, suposant que la matriu A s'emmagatzema a partir de l'adreça 0x00001000.

```
short A[8][4], B[8], C[4][8];
main() {
    int i, j; /* emmagatzemats en registres */
    for (i=0; i<8; i++)
        for (j=0; j<4; j++)
            C[j][i] = A[i][j] + B[i] + C[j][i];
}
```

Variable	Fallades	Encerts
A		
B		
C		

Problema 6.2

6.2. Disposem d'un processador de 16 bits (amb bus d'adreces de 16 bits) amb una memòria cache que té les següents característiques:

- Correspondència directa
- Mida total: 256 bytes
- Mida bloc: 16 bytes
- Escriptura immediata sense assignació

a) Ompliu la següent taula a partir de la seqüència de referències donades.

tipus	adreça (hex)	etiqueta (hex)	index MC (hex)	Encert/ Fallada	#bytes llegits MP	#bytes escrits. MP	lectura dades MC (Si/No)	escript. dades MC (Si/No)
R	4534							
R	4568							
W	13A4							
W	13A8							
R	3560							
W	453C							
W	60A0							
R	453C							
W	3900							
R	A238							

Problema 6.2

- b) Ompliu ara la mateixa taula, suposant que la la MC té una política d'escriptura retardada amb assignació.

tipus	adreça (hex)	etiqueta (hex)	índex MC (hex)	Encert/ Fallada	#bytes llegits MP	#bytes escrits. MP	lectura dades MC (Si/No)	escript. dades MC (Si/No)
R	4534							
R	4568							
W	13A4							
W	13A8							
R	3560							
W	453C							
W	60A0							
R	453C							
W	3900							
R	A238							

- c) Indiqueu per cada política:

- taxa de fallades
- número de bytes llegits d'MP
- número de bytes escrits a MP

Problema 6.6

- 6.6.** Es vol definir la política d'escriptura de la memòria cache d'un determinat procesador. Es consideren les alternatives: (1) escriptura immediata sense assignació i (2) escriptura retardada amb assignació.

Mitjançant simulació s'han obtingut les següents mesures:

- percentatge d'escriptures (pe): 20%
- percentatge de blocs modificats sobre el total de blocs reemplaçats (pm): 33.33%
- taxa d'encerts cas (1): 0.9
- taxa d'encerts cas (2): 0.85

El temps d'accés a memòria cache en cas d'encert (t_h) és de 10 ns. La lectura o escriptura d'un bloc de memòria principal (t_{block}) requereix 100 ns.

Es demana:

- a) Calculeu el temps mitjà d'accés a memòria (t_{am}) en ambdues alternatives.
- b) Indiqueu quina alternativa seria la més ràpida per a un programa que només fes lectures.

Problema 6.7

- 6.7. Tenim una CPU amb una cache en què hem observat les característiques següents quan executa una col.lecció de programes representatius:
- CPI_{ideal} (CPI suposant que tots els accessos a memòria són encerts a la cache): 1.5 cicles/instr.
 - Temps de cicle (t_c): 10 ns
 - Nombre de referències per instrucció (nr): 1.6
 - Cache d'instruccions i dades separades
 - Cache de dades d'escriptura retardada amb assignació
 - Les característiques de les dues caches són les següents:

Problema 6.7

Característica	Memòria cache	
	d'Instruccions	de Dades
Nombre de referències a memòria per instrucció (n_r)	1	0.6
Percentatge d'escriptures per referència (p_e)	-	40%
Percentatge de blocs modificats sobre tots els reemplaçats (p_m)	-	20%
Taxa de fallades (m)	4%	10%
Penalització (t_p) en reemplaçar un bloc no modificat	10 cicles	15 cicles
Penalització (t_p) en reemplaçar un bloc modificat	-	20 cicles
Temps de servei en cas d'encert (t_h)	1cicle	1cicle

- a) Quin serà el temps mitjà d'accés a memòria (t_{am}) en cicles?
- b) I el temps mitjà d'execució d'una instrucció (t_{exe}) en ns?

Problema 6.9

6.9. A l'hora de dissenyar una memòria cache d'instruccions de 8 KB s'està dubtant entre les següents mides de bloc, i se n'ha medid la taxa de fallades per una col·lecció de programes representatius:

- 8 bytes: 16%
- 16 bytes: 10%

Es demana que:

a) Calculeu el temps mitjà d'accés a memòria (t_{am}) sabent que:

- El temps d'accés de les caches en cas d'encert (t_h) és 1 cicle
- El temps de servei de la memòria per servir blocs de 8 bytes és de 6 cicles, mentre que el temps per servir blocs de 16 bytes és de 12 cicles

b) Quant ocuparan totes ~~les etiquetes~~ en cada una de les dues possibilitats sabent que:

- MP té 2^{32} bytes
- Les caches són de correspondència directa
- La unitat mínima d'accés a memòria és el byte

les etiquetes i bits de validesa

Problema 6.3

6.3. El siguiente programa multiplica una matriz $A(32 \times 32)$ por un vector $B(32)$ produciendo como resultado un vector $C(32)$:

```
char A[32][32], B[32], C[32];
main() {
  int i, j;
  ...
  for (i=0; i<32; i++)
    for (j=0; j<32; j++)
      C[i] = A[i][j] * B[j] + C[i]
}
```

Los elementos de A , B y C son bytes. Todos los elementos de C han sido previamente inicializados a cero. A está almacenada a partir de la dirección 0 de memoria (direcciones 0..1023). B está almacenada justo a continuación de A (direcciones 1024..1055) y C justo a continuación de B (direcciones 1056..1087). Las variables i, j están almacenadas en registros del procesador.

El computador dispone de una memoria cache de correspondencia directa que almacena 4 bloques de 32 bytes cada uno, con escritura inmediata. Suponiendo que al inicializarse la ejecución del bucle anterior la cache no tiene ningún dato, calcula la tasa de aciertos de la memoria cache.

Problema 6.10

6.10. Disposem d'un processador de 16 bits (paraules i adreces de 16 bits) amb una memòria cache que té les següents característiques:

- Correspondència associativa de 2 vies
- Mida total: 1024 bytes
- Mida bloc: 16 bytes
- Política d'escriptura immediata sense assignació
- Algorisme de reemplaçament: LRU

Ompliu la següent taula a partir de la seqüència de referències donades, on a la columna tipus *R byte* indica lectura d'1 byte, *R word* és lectura de 2 bytes, *W byte* és escriptura d'1 byte i *W word* és escriptura de 2 bytes. La mida de les lectures i escriptures s'ha d'especificar en bytes.

Problema 6.10

tipus	mida	Adreça (hex)	#bloc (hex)	#conjunt (hex)	h/m	Lectura MP		Escr. MP	
						adreça	Mida	adreça	mida
L	B	A930	A93	13					
L	W	B930	B93	13					
E	B	A972	A97	17					
E	W	A932	A93	13					
E	B	C935	C93	13					
L	W	C934	C93	13					
E	B	B976	B97	17					
E	W	B936	B93	13					
L	B	B938	B93	13					
L	W	A978	A97	17					

Problema 6.11

6.11. Suposem que tenim un processador amb una memòria cache de dades amb les següents característiques:

- 64 conjunts
- 4 blocs per conjunt
- 32 bytes per bloc
- paraules de 4 bytes
- algorisme de reemplaçament LRU

Sobre aquest sistema de memòria s'executen 2 versions diferents d'una mateixa aplicació:

```
int A[128][1024]; /* emmagatzemada a partir de l'adreça 0 */

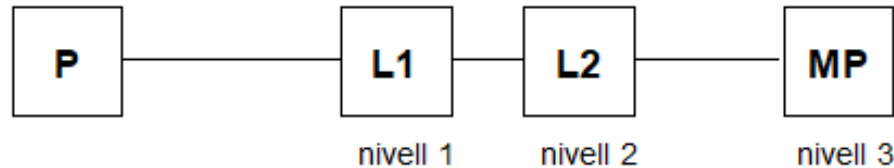
/* versió A */
sumA = 0;
for (i=0; i<128; i++)
    for (j=0; j<1024; j++)
        sumA = sumA + A[i][j];

/* versió B */
sumA = 0;
for (j=0; j<1024; j++)
    for (i=0; i<128; i++)
        sumA = sumA + A[i][j];
```

Indiqueu quantes fallades hi ha a la cache de dades per a cada una de les dues versions. Considereu que les variables i , j i $sumA$ estan guardades en registres.

Problema 6.12

6.12. El subsistema de memòria d'un determinat computador està organitzat en tres nivells:



Les característiques del processador i de cada nivell són les següents:

- **P**: La unitat d'adreçament es el byte.
- **L1**: Memòria cache de nivell 1 amb 4 blocs de 4 bytes cada un.
Correspondència directa.
- **L2**: Memòria cache de nivell 2 amb 16 blocs de 4 bytes cada un.
Correspondència associativa de 2 vies. Algorisme de reemplaçament LRU.
- **MP**: Memòria principal amb capacitat de 1 Mbyte.

El funcionament del sistema s'explica a continuació. L1 rep les peticions a memòria que genera P. En cas de fallada a L1, primer es comprova si el bloc es troba a L2. Si hi és, es copia aquest bloc a L1 sense necessitat d'accedir a MP. Si no hi és tampoc a L2, s'accedeix a MP i es copia a les dues caches. D'aquesta manera, L2 actua com una cache per a les peticions de memòria que genera L1.

Problema 6.12

Per avaluar el rendiment del subsistema de memòria, definim, per a cada nivell i , una taxa d'encerts $h_i = \text{núm. encerts a la } L_i / \text{núm. peticions a la } L_i$

Donada la següent seqüència de 28 adreces de lectura a memòria:

0, 5, 10, 12, 34, 0, 66, ... (que es repeteix 3 vegades més)

- a) Suposant que les caches estan inicialment buides, calculeu les taxes h_1 i h_2
- b) Calculeu de nou les taxes h_1 i h_2 suposant que ara canviem l'algorisme de reemplaçament de L_2 : usarem l'algorisme FIFO, que consisteix en reemplaçar el bloc que fa més temps que s'ha portat a la cache.

Examen final 2020/21-Q2

Considera un processador amb adreces de 32 bits i una memòria cache de dades amb les següents característiques:

- Capacitat: 1kB
- Mida del bloc: 64 bytes
- Correspondència associativa per conjunts
- Blocs per conjunt: 2
- Política de reemplaçament LRU
- Esriptura retardada amb assignació (write-back, write-allocate)

Suposem que la memòria cache és inicialment buida, i executem el següent programa al nostre processador:

```
int V[128];  
int M[128][16];  
  
void main() {  
int i, j, tmp;           // a $t0, $t1 i $t2 respectivament  
    for (i=0; i<128; i++)  
        for (j=0; j<16; j++) {  
            tmp = M[i][j];  
            V[i] = V[i] + tmp;  
        }  
}
```

Considera addicionalment que el vector V comença a partir de l'adreça 0x10010000 i que tant el vector V com la matriu M estan convenientment inicialitzats, tot i que no s'ha de tenir en compte aquesta inicialització en la resolució de l'exercici.

Quin serà el número del bloc de MP on es troba M[0][0], en hexadecimal?

Bloc MP M[0][0] =

Quantes referències a memòria per dades es realitzaran durant l'execució del programa, i quantes fallades de cache es produiran?

Referències =

Fallades =

Examen final 2020/21-Q2

Decidim experimentar canviant aquesta cache per una altra amb les mateixes característiques (1kB, blocs 64 bytes) però amb una política de correspondència directa.

Quina serà ara la quantitat de fallades que s'obtindrà en l'execució d'aquest codi?

Fallades =