

Estructura de Computadores

Tema 6. Memoria Cache

Considerem un computador amb un processador MIPS funcionant a una freqüència de 0,5Ghz, i que dissipa una potència de 10 W. Suposem que la cache d'instruccions és ideal (sempre encerta), i que la cache de dades té un temps de servei en cas d'encert $t_h = 1$ cycle. El temps necessari per copiar un bloc de memòria principal a cache és $t_{block} = 99$ cycles. Els CPI dels diversos tipus d'instruccions (en absència de fallades) són:

	Salts	Loads	Resta d'instruccions
CPI	3	5	2

A través de simulacions amb un programa de test hem mesurat una taxa de fallades de la cache de dades del 4,4% (és a dir, $m = 0,044$). Totes les referències a memòria són lectures. El nombre d'instruccions executades és:

	Salts	Loads	Resta d'instruccions
núm. instr.	$1 \cdot 10^9$	$3 \cdot 10^9$	$6 \cdot 10^9$

- a) (0,4 pts) Calcula el CPI_{ideal} del programa (CPI promig amb cache ideal sense fallades)

$$CPI_{ideal} = \boxed{}$$

- b) (0,4 pts) Calcula, en segons, el temps d'execució (incloent-hi fallades de cache)

$$t_{exe} = \boxed{} \text{ s}$$

- c) (0,1 pts) Calcula l'energia total consumida durant l'execució del programa, en Joules

$$E = \boxed{} \text{ J}$$

- d) (0,4 pts) Calcula, en cycles, el temps d'accés mitjà a memòria dels loads per a aquest programa

$$t_{am} = \boxed{} \text{ cycles}$$

Suposem que tenim un processador de 32 bits amb una memòria cache de dades de 64 bytes, on cada bloc té 16 bytes. Suposem que executem els següents programes.

```
//programa A
int M[4][16];

void main() {
int i, j;    //en registre
    for (i=0; i<3; i++)
        for (j=0; j<16; j++)
            M[i][j] = M[i+1][j];
}
```

```
//programa B
int M[4][16];

void main() {
int i, j;    //en registre
    for (j=0; j<16; j++)
        for (i=0; i<4; i++)
            M[i][j] = M[i][j] + 1;
}
```

Calcula el nombre de fallades de la cache suposant que la memòria cache és inicialment buida. L'adreça base de la matriu M és 0.

- a) Suposant que la cache és de correspondència directa i té la política d'**escriptura retardada amb assignació**.

fallades_A =

fallades_B =

- b) Suposant que la cache és **completament associativa** (algorisme de reemplaçament LRU), i que té la política d'**escriptura immediata sense assignació**.

fallades_A =

fallades_B =

Un sistema disposa d'un processador MIPS (32 bits d'adreces i mida de paraula de 4 bytes), i una memòria cache (MC) de 64 Kbytes amb la següent organització:

- Correspondència directa
- Blocs de 256 bytes
- Escriptura immediata sense assignació

Estant la cache inicialment buida, un programa fa una seqüència de referències a memòria segons s'indica a la següent taula, on apareixen les adreces en hexadecimal i si són lectures o escriptures (L/E). Completa les columnes que falten indicant, per a cada referència: l'índex de MC; si és encert (e) o fallada (f); i el nombre de bytes de Memòria Principal (MP) llegits i/o escrits.

L/E	adreça (hex)	índex MC	encert (e)/ fallada (f)	bytes de MP	
				llegits	escrits
E	00010000				
L	00010008				
E	00010016				
L	00F40316				
E	03100004				
L	00010024				
L	03200308				
L	00F403A8				

Considerem un computador amb un processador MIPS funcionant a una freqüència de 500Mhz, i que dissipa una potència de 20 W. Suposem que la cache d'instruccions és ideal (sempre encerta), i que la cache de dades té un temps de servei en cas d'encert $t_h = 1$ cicle. El temps necessari per copiar un bloc de memòria principal a cache és $t_{block} = 59$ cicles. Els CPI dels diversos tipus d'instruccions (en absència de fallades) és:

	Salts	Loads	Resta d'instruccions
CPI	8	3	1

A través de simulacions amb un programa de test hem mesurat una taxa de fallades de cache de $m = 2,5\%$. Totes les referències a memòria són lectures. El nombre d'instruccions executades és:

	Salts	Loads	Resta d'instruccions
n. instr.	$3 \cdot 10^9$	$6 \cdot 10^9$	$20 \cdot 10^9$

a) (0,5 pts) Calcula el temps d'accés mitjà a memòria dels loads per a aquest programa, en cicles

$$t_{am} = \boxed{} \text{ cicles}$$

b) (0,5 pts) Calcula el temps d'execució del programa (incloent-hi fallades de cache), en segons

$$t_{exe} = \boxed{} \text{ s}$$

c) (0,1 pts) Calcula l'energia total consumida durant l'execució del programa, en Joules

$$E = \boxed{} \text{ J}$$

Considera que tenim un processador de 32 bits amb una memòria cache de dades de 1KB, on cada bloc té 32 bytes. Suposem que executem els següents programes.

```
// programa A
```

```
int M[8][64];
void main() {
    int i, j; //en registres
    for (i=0; i<8; i++)
        for (j=0; j<64; j++)
            M[i][j]++;
}
```

```
// programa B
```

```
int M[8][64];
void main() {
    int i, j; //en registres
    for (j=0; j<64; j++)
        for (i=0; i<8; i++)
            M[i][j]++;
}
```

Considera a més que la memòria cache és inicialment buida i l'adreça base de la matriu M és 0:

- a) Suposant que la cache és de correspondència directa i amb política d'escriptura retardada amb assignació, calcula la taxa de fallades per a cada programa (m_A i m_B):

$m_A =$

$m_B =$

- b) Suposant que la cache és associativa per conjunts (algorisme de reemplaçament LRU) i amb política d'escriptura immediata sense assignació, indica per a cada programa el mínim nombre de blocs per conjunt que minimitza la taxa de fallades i quina seria aquesta taxa de fallades mínima.

programa A:

blocs/conjunt:

taxa de fallades:

programa B:

blocs/conjunt:

taxa de fallades: