

M1 SESI – Architecture des Processeurs et Optimisation

TD5 – Cache – Hiérarchie des mémoires

Exercice 1

On étudie dans cette première partie un système contenant un cache d'instructions et un cache de données à correspondance directe. Les deux caches sont indépendants. Les adresses sont sur 32 bits. La stratégie d'écriture en mémoire est une stratégie Write Through : toute requête d'écriture émise par le processeur se traduit par une écriture en mémoire, et le cache n'est mis à jour que si le bloc est présent (Hit) dans le cache (stratégie Write Non-Allocate). Un tampon d'écritures postées permet d'éviter de geler le processeur lors des instructions d'écriture tant qu'il n'y a pas un trop grand nombre d'écritures consécutives.

Question 1 – Structure des adresses

On considère des caches à correspondance directe possédant une capacité de 8 lignes de 4 mots.

a - Donner la structure de l'adresse physique vue par le cache en précisant le nombre de bits dans chaque champ. Quel est le nombre total de lignes de cache dans l'espace adressable ?

b - Donner un schéma du cache mettant en évidence l'utilisation de chaque champ.

Question 2 – Analyse d'une application

Soit le programme assembleur ci-dessous, qui réalise l'addition vectorielle de deux vecteurs : $C = A + B$. Les 3 vecteurs **A**, **B** et **C** possèdent 20 composantes chacun. Ces vecteurs sont représentés par des tableaux d'entiers, stockés dans le segment **data**. Avant d'entrer dans la boucle, on charge l'adresse de l'élément **A[0]** dans le registre **\$8**, et l'adresse de la première case mémoire libre après le tableau **A** dans le registre **\$9**.

```
.data
.align      2
A : .word   1,   2,   3,   4,   5,   6,   7,   8,   9,  10
     .word  11,  12,  13,  14,  15,  16,  17,  18,  19,  20
     .space  48
B : .word  101, 102, 103, 104, 105, 106, 107, 108, 109, 110
     .word  111, 112, 113, 114, 115, 116, 117, 118, 119, 120
     .space  48
C : .word   0,   0,   0,   0,   0,   0,   0,   0,   0,   0
     .word   0,   0,   0,   0,   0,   0,   0,   0,   0,   0

.text
.align      2
.globl      main
.ent        main

.set        noreorder

main :
    lui      $8, 0x1000    # $8 <= address A
    addi     $9, $8, 80    # $9 <= address A + 80

loop :
    lw       $10, 0($8)    # $10 <= A[i]
    lw       $11, 128($8)  # $11 <= B[i]
    addi     $8, $8, 4      # i <= i + 1
    add      $10, $10, $11  # $10 <= $10 + $11
    bne      $8, $9, loop  # fin de boucle ?
    sw       $10, 252($8)  # C[i] <= $10
```

```

exit:
    ori    $2 , $0 , 10  # code appel système exit
    syscall

```

a - L'adresse de base du segment **code** est **0x00400000**. La première instruction rangée à cette adresse est donc l'instruction associée à l'étiquette **main**. Quelle est l'adresse de la première instruction de la boucle (instruction correspondant à l'étiquette **loop**) ?

b - L'adresse de base du segment **data** est **0x10000000**. Quelles sont les adresses des éléments **A[0]**, **B[0]**, et **C[0]** des trois tableaux ?

Question 3 – Analyse du fonctionnement du cache d'instructions

a - Représenter dans le tableau ci-dessous l'état du cache d'instructions à la fin de la première itération de la boucle en supposant que le cache est vide avant l'exécution de la première instruction du programme.

TAG	V	DATA3	DATA2	DATA1	DATA0

b - Quelles instructions ont déclenché un MISS sur le cache d'instructions pour atteindre cet état ? Quel est l'état du cache à la fin de la 2ème itération ? A la fin de la 20ème itération ?

c - Calculer le taux de miss après la 20e itération.

d - Le coût d'un MISS instruction (nombre de cycles de gel du processeur) est de 10 cycles, en déduire la durée d'exécution du programme (entre le branchement à la première instruction du programme principal et la fin de la dernière itération), si on néglige le coût des MISS sur le cache de données.

Question 4 – Analyse du fonctionnement du cache de données

a - En supposant que le cache de données est initialement "vide" (c'est-à-dire que les 8 cases ont un contenu "non valide"), donnez le contenu du cache de données à la fin de la première itération de la boucle, en précisant quelles instructions entraînent un MISS sur le cache de données et donc un gel du processeur.

TAG	V	DATA3	DATA2	DATA1	DATA0

b - Même question à la fin de la deuxième itération.

c - Combien y a-t-il de MISS par itération ? En déduire le nombre total de cycles de gel du processeur liés aux miss de données pour l'exécution complète du programme, en tenant compte du fait qu'un MISS sur le cache de données coûte deux cycles de plus qu'un MISS sur le cache d'instructions.

d - Quelle est finalement la durée totale d'exécution du programme, en prenant en compte tous les MISS sur les deux caches.

Question 5 – Optimisation du fonctionnement du cache de données

a - Pour cette application, quel pourcentage du temps d'exécution les cycles de gel du processeur dus aux MISS de conflit sur le cache de données représentent-ils ?

b - Comment peut-on réduire le taux de MISS sur le cache de données, sans modifier l'architecture matérielle et sans modifier la structure du programme ?

c - Recalculer la durée moyenne d'une itération avec la proposition faite à la question précédente.

Exercice 2

On considère un cache de 1024 octets avec une taille de bloc (ou ligne) de 16 octets. La mémoire est adressée par octet et les adresses sont sur 32 bits. Le cache est géré en écriture allouée ou *write-allocate* (un défaut en écriture entraîne le chargement du bloc correspondant), et en écriture différée ou ré-écriture ou *write-back* (une donnée n'est modifiée en mémoire principale que lorsque la ligne de cache correspondante est évincée du cache). La politique de remplacement est la politique LRU (Least Recently Used).

Question 1

a - Donner la structure de l'adresse physique vue par le cache en précisant le nombre de bits dans chaque champ dans le cas d'un cache à correspondance directe, d'un cache associatif à 2 voies (ou par ensemble de 2 blocs, ou encore 2-way associatif) et d'un cache associatif à 4 voies.

b - Donner un schéma du cache mettant en évidence l'utilisation de chaque champ dans le cas d'un cache associatif à 2 voies. Quelle est l'utilité des deux niveaux d'associativité ?

Question 2

Soient **X**, **Y** et **Z** des tableaux de réels double-précision (8 octets). **X** est implanté à partir de l'adresse **0x10010**, **Y** à partir de l'adresse **0x20210** et **Z** à partir de l'adresse **0x020410**.

On étudie la boucle suivante :

```
for (i = 0; i < N; i++) {
    Z[i] = X[i] + Y[i];
}
```

En supposant que $N = 3$, indiquer pour chaque accès mémoire (lecture et écriture), si c'est un succès ou un échec (les accès sont faits dans l'ordre **X[i]**, **Y[i]**, **Z[i]**) :

a - Dans le cas d'un cache à correspondance directe en donnant pour chaque référence le

numéro de la ligne de cache où elle est stockée.

b - Dans le cas d'un cache associatif à 2 voies en donnant pour chaque référence le numéro de l'ensemble où elle est stockée.

c - Dans le cas d'un cache associatif à 4 voies en donnant pour chaque référence le numéro de l'ensemble où elle est stockée.

d - Pour une valeur de N quelconque, quel est le taux de miss pour les 3 configurations de cache considérées ?

On étudie maintenant la boucle suivante et on considère uniquement le cas d'un cache associatif à 2 voies :

```
for (i = 0; i < N; i++) {  
    Z[i] = X[i + 1] + Y[i + 1];  
}
```

e - Pour chaque accès mémoire, indiquer s'il s'agit d'un succès ou d'un échec et donner pour chaque référence le numéro de l'ensemble où elle est stockée en supposant $N = 4$. Quel est le taux de miss pour une valeur de N quelconque ?

Exercice 3

On s'intéresse à une machine monoprocesseur construite autour du processeur MIPS pipeline à 5 étages étudié en cours. On souhaite étudier l'influence de la mémoire sur la performance globale du système. Le nombre moyen de cycles par instruction (CPI) dans l'hypothèse d'un système mémoire idéale (les écritures et les lectures se font en 1 cycle) est de 1.1 cycle par instruction. Les instructions de chargement (**load**) représentent 20% des instructions exécutées et les instructions de rangement (store), 10%.

La mémoire physique du système a une capacité de 64 Méga-octets. Elle est adressée par octets. Le nombre de cycles nécessaire pour une lecture est de 10 cycles processeur, il est de 4 cycles pour une écriture.

Question 1

a - Par rapport au système idéal, quelle est l'augmentation du nombre moyen de cycles par instruction due aux écritures (DW) ? Quelle est l'augmentation du nombre moyen de cycles par instruction due aux lectures (DR) ?

b - Quel est le nombre moyen de cycles par instruction (CPI) ?

Pour minimiser la dégradation de performance due à la lenteur de la mémoire, on introduit un cache à deux niveaux d'associativité (associatif à 2 voies) d'une capacité de 16 Kilo-octets (cache commun pour les données et les instructions). Le temps d'accès est de 1 cycle. La taille du bloc est de 16 octets. On utilise une stratégie d'écriture simultanée dans la mémoire et dans le cache (*write-through*) sans mise à jour du cache si la donnée écrite n'est pas présente dans le cache (pas de Miss sur écriture). Il n'y a pas de buffer d'écriture. La durée d'une lecture en cas de Miss est de 23 cycles (22 cycles pour le transfert d'une ligne de cache depuis la mémoire jusqu'au cache puis 1 cycle pour envoyer la donnée au processeur)

Question 2

On suppose que le taux de Miss est de 5%. En cas de Miss, le processeur ne reçoit la donnée demandée qu'après le chargement complet du bloc manquant dans le cache. On cherche à calculer le nombre moyen de cycles par instruction.

a - Donner la taille des champs TAG, INDEX et OFFSET.

b - Par rapport au système idéal, quelle est l'augmentation du nombre moyen de cycles par instruction due aux écritures (DW) ? Quelle est l'augmentation du nombre moyen de cycles par

instruction due aux lectures (DR) ?

c - Quel est le nombre moyen de cycles par instruction (CPI) ?

Question 3

Pour limiter la pénalité introduite par les Miss en lecture, on introduit un cache secondaire (un cache de niveau 2) entre le cache et la mémoire. Ce cache secondaire a une capacité de 512 Kilo-octets. La taille du bloc est de 16 octets. C'est un cache à correspondance directe. Tous les blocs contenus dans le cache primaire sont contenus dans le cache secondaire (caches inclusifs).

Le taux de Miss dans le cache secondaire est le pourcentage de requêtes émises par le cache primaire qui font Miss dans le cache secondaire. Il est de 15%. Le temps d'accès au cache secondaire est de 3 cycles (idem pour le transfert d'un bloc du cache secondaire dans le cache primaire).

a - Du point de vue du processeur, quelle est la durée d'une lecture en cas de Miss dans le cache primaire ?

b - Par rapport au système idéal, quelle est l'augmentation du nombre moyen de cycles par instruction due aux écritures (DW) ? Quelle est l'augmentation du nombre moyen de cycles par instruction due aux lectures (DR) ?

c - Quel est le nombre moyen de cycles par instruction (CPI) ?

Question 4

On cherche enfin à diminuer la pénalité introduite par les écritures. Quelle solution suggérez-vous ? Quel bénéfice peut-on espérer sur le nombre moyen de cycles par instruction ?