TD 7-8 4MMARCA, Ensimag 2A/S6, ISI/SEOC

Caches

Architecture avancée, 2018-2019

Les caches permettent de cacher la latence d'accès à la mémoire. Les exercices qui suivent servent d'une part à quantifier le gain qu'il y a à utiliser un cache et d'autre part à en faire l'implantation.

Exercice 1 : étude quantitative

On fait l'hypothèse qu'une machine pipeline idéalement capable d'exécuter une instruction par cycle possède un cache de premier niveau. On ignore dans un premier temps les échecs sur lecture d'instructions, et les écritures sont non bloquantes. Le processeur et la mémoire fonctionnent à 500 MHz (ordre de grandeur classique pour les processeurs embarqués dans les téléphones portables, appareils photo numériques, etc). La lecture dans le cache se fait dans le cycle, et la latence d'accès à la mémoire est de 8 cycles. Pour un programme donné, 25% des instructions sont des lectures mémoires, et 10% de ces lectures tapent en dehors du cache.
Question 1 – Donnez le nombre de cycles moyen d'accès à la mémoire pour ce programme.

Le programme met 10 secondes à s'exécuter sur une machine idéale (sans aucun échec d'accès au cache).
Question 2 – Donnez le temps d'exécution sur la machine réelle.

On ajoute un cache de niveau 2 pour minimiser la latence dans un système mémoire pour lequel la fréquence du processeur est très supérieure à celle de la mémoire. L'inspection des caractéristiques d'un cpu pour l'embarqué donne :

```
model name           : Intel(R) Atom(R) E processor 1.70GHz
cpu MHz              : 597.698
host bus clock speed : 99.0579 MHz.
```

En cas de *hit*, le cache de niveau 1 a une latence de 1 cycle processeur, celui de niveau 2 une latence de 5 cycles processeur. L'accès à la mémoire en cas de *miss* L2, la latence de la mémoire est de 5 cycles mémoire.
Question 3 – En faisant l'hypothèse que le taux de *miss* du cache de niveau 1 est de 7% et celui du cache de niveau 2 de 12%, donnez le CPI de cette architecture.

Exercice 2 : implantation

On veut implanter un cache à correspondance directe dont la capacité totale est de 8 KBytes. On choisit une organisation en 512 lignes de 4 mots. On rappelle par ailleurs qu'une mémoire ne peut être accédée qu'une fois par cycle, que ce soit en lecture ou en écriture.

Question 4 – Donnez la structure de l'adresse du point de vue du cache.

On s'intéresse tout d'abord à la lecture.

Question 5 – Dessinez un schéma bloc de la partie lecture du cache qui ne prend en compte que les informations liés aux adresses.

Question 6 – Quelle précaution faut-il prendre lors de ma mise sous tension de la machine ? Comment suggérez vous d'implanter cette solution ?

Question 7 – Quelle doit être la chaîne combinatoire la plus longue de cette architecture ?

Question 8 – Quantifiez le nombre de bits de mémoire et le nombre de comparateurs (et leurs tailles) nécessaire à l'implantation cette implantation.

On veut réaliser à présent un cache instruction de capacité identique, mais associatif à 4 voies.

Question 9 – Donnez la structure de l'adresse du point de vue du cache.

Question 10 – Définissez les opérations à effectuer pour la détection du *miss*. En déduire un diagramme d'implantation.

On veut réaliser à présent un cache instruction de capacité identique, mais totalement associatif.

Question 11 – Donnez la structure de l'adresse du point de vue du cache.

Question 12 – Définissez les opérations à effectuer pour la détection du *miss*. En déduire un diagramme d'implantation. Quel problème majeur rend problèmatique l'usage d'un cache totalement associatif au premier niveau ?

Question 13 – Quantifiez le nombre de bits de mémoire et le nombre de comparateurs (et leurs tailles) nécessaire à l'implantation et les comparer à ceux de la solution à correspondance directe.

On ajoute à présent l'écriture.

Question 14 – Quels sont les 2 évènements possibles lors d'une écriture ? Quel problème cela pose-t-il ? Comment suggérez vous d'y remédier ?

Question 15 – Inserez le matériel nécessaire à la gestion de l'écriture. Donnez la chaîne combinatoire la plus longue de cette architecture.

Exercice 3 : étude des accès (Exercice complémentaire)

Soit le code C suivant :

```
int32_t x[1024][64];
for (j = 0; j < 64; j++)
   for (i = 0; i < 1024; i++)
      x[i][j] = 2 * x[i][j];
```

Question 16 – Donnez l'adresse de la case `x[i][j]` en mémoire comme une fonction de x, i et j.

Question 17 – En supposant que le cache soit à correspondance directe, de taille 2Kb organisé comme

64 lignes de 8 mots, et que l'adresse du tableau `x` soit alignée sur une frontière de 2Kb, donnez le nombre de *miss* pour l'exécution de ce code.

Question 18 – Même question en permutant les 2 boucles.

Soit le code C suivant qui effectue une multiplication/accumulation de valeurs contenues dans des matrices :

```
for (i = 0; i < n; i++)
  for (j = 0; j < m; j++)
    for (k = 0; k < p; k++)
       C[i][j] = C[i][j] + A[i][k] * B[k][j];
```

Question 19 – En supposant que l'on dispose d'un cache, a-t-on intérêt à permuter les boucles d'indice j et k ?

```
for (ii = 0; ii < SIZE; ii += BLOCK_SIZE)
  for (kk = 0; kk < SIZE; kk += BLOCK_SIZE)
    for (jj = 0; jj < SIZE; jj += BLOCK_SIZE)
      for (i = ii; i < ii + BLOCK_SIZE && i < SIZE; i++)
        for (k = kk; k < kk + BLOCK_SIZE && k < SIZE; k++)
          for (j = jj; j < jj + BLOCK_SIZE && j < SIZE; j++)
             C[i][j] = C[i][j] + A[i][k] * B[k][j];
```

Question 20 – En quoi la modification ci-dessus est-elle intéressante ?