

Architecture des microprocesseurs

Mémoire cache du processeur UltraSPARC II

Le processeur UltraSPARC II de chez SUN Microsystems est un processeur RISC 64 bits. Si les données manipulées ont une taille de 64 bits, les instructions sont codées sur 32 bits. Quant aux adresses, leur taille est fixée à 44 bits. Ce processeur possède en interne un cache de données et un cache d'instructions de type différent.

- Le cache de données est un cache de 16 KOctets à correspondance directe. Chaque ligne est composée de 32 octets (ce qui correspond à 4 mots de données).
- Le cache d'instructions est un cache de 16 KOctets, associatif à 2 groupes. Chaque ligne comprend 32 octets (ce qui représentent 8 instructions).

Le système étudié comprend une mémoire principale adressable par octet. Dans cette mémoire sont stockées les données et les instructions.

Partie A : cache de donnée

- 1) Indiquer la taille totale du cache.
Dessiner l'architecture du cache de données, en précisant tous les paramètres de l'architecture (nombre de lignes, taille des bus, ...).
- 2) On charge dans ce cache le bloc de données (32 octets) dont l'adresse mémoire de la première donnée du bloc est 0x000AB240020. Indiquer la ligne du cache qui reçoit ce bloc et le contenu du champ étiquette de la ligne du cache.

Partie B : cache d'instructions

- 3) Indiquer la taille totale du cache.
Dessiner l'architecture du cache d'instructions, en précisant tous les paramètres de l'architecture (nombre de lignes, taille des bus, ...).
- 4) On charge un bloc de 8 instructions dans ce cache dont l'adresse de la première instruction est 0x00000AB0220.
Déterminer le numéro de l'ensemble. Indiquer la ou les lignes de cache qui pourront contenir ce bloc. Quel sera alors le contenu du champ index ?

Exercice sur l'efficacité de la mémoire cache

Extrait d'un sujet d'exercices de Julien Dusser – IRISA – IFSIC – Université de Rennes 1

On considère un cache de données de 16 Ko à correspondance directe avec des lignes de 64 octets. Soit le morceau de programme suivant :

```
int a[NL][NC];
for (int i = 0; i < NL; i++)
    for (int j = 0; j < NC; j++)
        x = x + a[i][j];
```

Pour toute la suite de l'exercice, on fera les suppositions suivantes. On suppose que les variables i , j , x utilisent des registres. On supposera que l'adresse de début du tableau, $\&a[0][0]$, commence au début d'une ligne de cache. On rappelle que l'adresse $\&a[i][j]$ de l'élément (i, j) du tableau a est égale à $\&a[0][0] + i \times NC + j$, et on supposera qu'un `int` fait 4 octets.

- Q1) Le taux d'échecs est défini comme le nombre total d'échecs divisé par le nombre total d'accès. En supposant NC et NL grands, quel est le taux d'échecs du cache de données ?

On considère maintenant le morceau de programme suivant (permutation de i et j) :

```
int a[NL][NC];
for (int j = 0; j < NL; j++)
    for (int i = 0; i < NC; i++)
        x = x + a[i][j];
```

- Q2) Quel est le taux d'échecs si $NL = NC = 256$?
- Q3) Quel est le taux d'échecs si $NL = 256$ et $NC = 272$?