

TD 4 Architecture générale

Processeurs / Mémoire

Exercice 1 : gros boutisme et petit boutisme

Considérons deux ordinateurs possédant des mots mémoire de 32 bits. L'un adopte la représentation gros-boutiste, l'autre la représentation petit-boutiste.

- a) *rappelez l'organisation des octets dans un mot mémoire avec chacune de ces deux représentations.*
- b) On suppose que ces machines stockent les informations suivantes pour un enregistrement d'une base de données :
- Nom : « Luc Duval », qui est une chaîne de caractères, chaque caractère occupant un octet en mémoire ;
 - Age : 31, qui est un entier stocké dans un mot mémoire ;
 - Numéro de service : 260, qui est aussi un entier (pour info $260 = 1 \times 256 + 4$) stocké dans un mot mémoire.

Donnez les représentations en mémoire de ces données pour chacune des deux machines (pour info : 20 octets, soit 5 mots mémoires, sont nécessaires).

- c) On effectue un transfert de ces données depuis la machine gros-boutiste vers la machine petit-boutiste. Si ce transfert est effectué octet par octet (de l'octet 0 à l'octet 19), *comment sont interprétées les valeurs reçues par la machine petit-boutiste ?*
- d) Puisque la numérotation des octets est inversée entre les deux représentations, on décide, après le transfert, d'inverser l'ordre des octets à l'intérieur d'un mot mémoire. *Quel est le résultat obtenu ?*
- e) *Quelle solution pouvez-vous proposer pour obtenir un transfert correct des données ?*

Exercice 2 : gestion de cache mémoire

Soit un processeur comportant un cache primaire de données (on ne se soucie pas des instructions) de 16 ko. La taille d'un mot mémoire est toujours de 32 bits.

- a) *pour se dérouiller les neurones : combien de mots mémoire peuvent être stockés dans ce cache ?*

La politique de gestion des transferts entre le cache et la mémoire est la suivante : si le processeur ne trouve pas une donnée dans le cache, il la rapatrie depuis la mémoire vers le cache à la place de la donnée du cache la moins récemment utilisée (algorithme LRU). Un programmeur décide d'utiliser un tableau de N entiers (chaque entier est représenté par un mot de 16 bits), et de lui appliquer une série de P traitements linéaires, depuis le début jusqu'à la fin du tableau (en clair : P boucles **for i allant de 1 à N**).

- b) le temps c d'accès au cache est de 8 ns, le temps m d'accès à la mémoire est de 40 ns. On considère que les temps de traitement des données et d'écriture dans le cache sont négligeables. *Exprimer en fonction de c , m , N et P le temps moyen d'exécution du programme, sachant qu'au départ, le cache ne contient aucune donnée du tableau.* Pour info, N a une très grande influence sur ce temps. Applications numériques :

$$N = 5\,000, P = 10 ; \text{ puis } N = 10\,000, P = 10.$$

On ajoute un cache de données secondaire de 256 ko, de temps d'accès $c' = 14$ ns. La politique de gestion de ce cache est la même que celle du cache primaire. Si le processeur ne trouve pas la donnée dans le cache primaire, il regarde dans le cache secondaire. Si la donnée s'y trouve, il la recopie dans le cache primaire selon la méthode LRU. Si elle ne s'y trouve pas, le processeur interroge la mémoire principale et recopie cette donnée dans les deux caches selon la méthode LRU.

Exprimer en fonction de c , c' , m , N et P le temps moyen d'exécution du programme, sachant qu'au départ, le cache ne contient aucune donnée du tableau. Recalculer les applications numériques précédentes ainsi que la valeur de ce temps pour $N = 200\,000$, $P = 10$.

- c) calcul moyen de temps d'accès : Le concepteur de l'ordinateur cherche à calculer les temps d'accès à une donnée avec les configurations à 1 puis à 2 caches pour préparer sa plaquette marketing. Pour ce faire, il se place dans le cas où les accès sont faits de manière totalement aléatoire dans la mémoire. *La machine comportant une mémoire principale de 1 Mo, quel est le temps d'accès moyen pour la configuration à 1 cache ? à 2 caches ? Calculez les temps théoriques d'accès aux caches permettant dans ce cas d'améliorer le temps d'accès moyen à la mémoire.*
- d) *Commentez ces résultats en regard des différents éléments vus en cours.*

Exercice 3 : Types de cache

a) A votre avis, quels signaux transitent sur le schéma suivant ?



- b) Quelles sont alors les informations que doit contenir le cache ? En déduire une organisation du cache et la méthode de recherche de données associée. A votre avis, quel est le principal inconvénient de cette solution ?
- c) Une autre organisation du cache consiste à faire en sorte qu'une adresse demandée par le processeur ne puisse être située qu'à un endroit du cache. Comment réaliser cette organisation ? Avec cette solution, plusieurs adresse peuvent être situées au même endroit du cache. Comment les différencier ?
- d) En utilisant des portes logiques et des comparateurs, dressez un schéma de ce type d'organisation.