

# Technologie mémoire, mémoire cache

M. Dubacq

S1D 2009

## 1 Les types de mémoire

**Objectif :** *Connaître les qualificatifs des mémoires, connaître les technologies existantes*

1. Pour tous les types de mémoire suivants, donnez les qualificatifs applicables ainsi que les ordres de grandeur des paramètres numériques pour la capacité (usuelle), la taille physique, la latence : DRAM, ROM, SRAM, disque dur, disquette HD, DVD, cédérom, bande magnétique.

## 2 Overclocking et mémoire

L'*overclocking* est une technique qui consiste en l'accélération de l'horloge d'un processeur pour obtenir des performances meilleures. Toutefois, la cadence du bus de données, du northbridge et de la mémoire sont toutes déterminées à partir de la cadence du processeur en divisant par un entier. Cet entier est, sur certaines cartes-mères, réglable.

Une RAM est prévue pour fonctionner à 333 MHz. La vitesse du processeur qui va avec est de 3 GHz. En fait, cette RAM est capable de fonctionner sans problèmes jusqu'à 400 MHz, et le processeur peut être accéléré jusqu'à 10% sans défauts (il peut être accéléré uniquement par paliers de 100 MHz).

1. Expliquez pourquoi accélérer modérément la cadence du processeur peut amener à des meilleures performances, et pourquoi l'accélérer énormément peut ne pas fonctionner du tout.
2. Quel est le multiplicateur utilisé pour obtenir la cadence nominale ?
3. Peut-on réaliser une accélération de la cadence du processeur qui ne crée aucun défaut sans changer le multiplicateur ? Que deviendrait la vitesse de la mémoire si on le faisait ?
4. Peut-on avoir une accélération du processeur et de la mémoire qui privilégie un peu plus la mémoire ?

## 3 Mémoire cache

On se place dans le cas d'un processeur qui utilise un adressage de 4 Gio de mémoire, avec 8 kio de mémoire cache. Cette taille de mémoire cache ne changera pas dans tout l'exercice. On utilisera des mots-mémoires de 32 bits.

1. Quelle est la taille d'une adresse (en bits) ?

2. On utilise les données suivantes :

- temps d'accès à la mémoire cache : 5 ns,
- latence d'accès à la mémoire principale : 40 ns ; mémoire,
- temps de cycle en mode page : 10 ns par mot mémoire ; supplémentaire ;

Quel est le temps nécessaire pour consulter un mot mémoire en cas de succès de cache ? Et pour seulement un octet ?

3. Si la taille d'un bloc de mémoire est fixée à 512 bits, en cas de défaut de cache, quel est le temps de pénalité ?

4. Même question si la taille d'un bloc de mémoire est fixée à 256 bits.

5. On a le choix entre trois modes de construction de cette mémoire : par cache direct de 256 bits par bloc (temps d'accès 3 ns), par cache associatif par ensemble à quatre voies de 256 bits par bloc (temps d'accès 4 ns), par cache associatif mais avec 512 bits par bloc (temps d'accès 5 ns).

Des expériences montrent que pour le genre d'applications que l'on va utiliser sur cette machine, dans la première solution, on a un taux de succès cache de 75%, dans la deuxième un taux de succès de 80% et dans la troisième un taux de succès de 87%.

Quel est la solution la plus intéressante ? Justifiez en exhibant le temps moyen d'accès à la mémoire dans les trois cas.

6. Dans le cas de la première solution, décrivez comment on décompose une adresse (nombre de bits en particulier). Dites aussi combien de blocs différentes peuvent résider en mémoire cache.

7. Dans le cas de la deuxième solution, décrivez comment on décompose une adresse (nombre de bits en particulier). Dites aussi combien de blocs différents peuvent résider en mémoire cache, ainsi que le nombre de blocs par voie.

8. Dans le cas de la troisième solution, dites comment se décompose une adresse.

9. On se place dans le cadre de la troisième solution (cache associatif). On veut accéder aux mots-mémoires qui vont de l'adresse 0x40000000 à l'adresse 0x40003FC (inclus). Dans l'hypothèse la plus pessimiste possible, dites combien il y a de défauts de cache en expliquant pourquoi.

10. Quelles sont les politiques possibles d'écriture en cache en cas de succès ?

## 4 Gestion mémoire de matrices

Une matrice est représentée en C par un tableau de taille  $N^2$  de `float` (c'est-à-dire des nombres codés selon le standard IEEE 754).

Dans l'ordinateur que nous examinons, les blocs de mémoire cache de données (cache associatif) sont de 256 mots de 4 octets, avec une taille totale de mémoire cache de 2 Mo. On veut faire une multiplication de matrices.

Rappel : le coefficient  $c_{ij}$  d'une matrice  $C = A \times B$  est

$$c_{ij} = \sum_{1 \leq k \leq N} a_{ik} b_{kj}.$$

Le temps de calcul d'un produit de matrice est donc le temps de  $N^3$  multiplications et  $N^3$  additions, ainsi que quelques opérations qui prennent assez peu de temps par rapport au reste (tests de fin de boucle).

1. Quelle est la quantité de mémoire nécessaire pour stocker une matrice de taille  $4 \times 4$ ? Et une matrice de taille  $N \times N$  en général?
2. On veut calculer  $C = A^2$ . À chaque étape de calcul, on réaffiche complètement les deux matrices (A et C, même si C n'est pas encore calculée complètement). Jusqu'à une certaine taille de matrice, on constate que le temps de calcul ressemble à  $t = k \times N^3$ , avec un certain  $k$  qui dépend de la vitesse du processeur. Mais à partir d'une certaine taille, on constate un très fort ralentissement. Comment peut-on l'expliquer? Quelle est cette taille?
3. On constate que l'affichage d'une matrice entre la taille 8 et la taille 9 est assez important. Quelle peut-être l'explication?