

## CONTRÔLE TERMINAL – L3 INFO : ARCHITECTURE DES ORDINATEURS

Durée =1h30 ; aucun document autorisé

### Exercice 1 (4 points)

1. Expliquer **pourquoi** la mémoire cache permet d'améliorer les performances d'un ordinateur.
2. Expliquer le fonctionnement de l'algorithme LRU
3. Quel est l'inconvénient d'un cache à associativité totale ?
4. Dans une case de la mémoire cache, pourquoi stocke-t-on l'étiquette plutôt que le numéro de bloc ?

### Exercice 2 (6 points)

Un ordinateur a une mémoire principale constituée de 1 Mo. Il possède un cache direct de 8 lignes. La taille d'un bloc est de 16 octets. On suppose que le temps d'accès au cache est de 1 ns alors que le temps d'accès à la RAM est de 10 ns.

1. Quelle est la capacité du cache ? Quel est le nombre de bits nécessaires pour décrire une adresse de la mémoire principale ?
2. Calculer le nombre de bits pour les champs Etiquette, Ligne et Déplacement de l'adresse d'un mot de la mémoire principale
3. Le cache est initialement vide. Le processeur lit 256 octets à partir des adresses 0, 1, 2...255 dans cet ordre, puis relit ces mêmes 256 octets mais dans l'ordre inverse.
  - a. Décrire le contenu du cache après lecture des 512 octets.
  - b. Calculer le temps nécessaire à la lecture des 512 octets.

## Exercice 3 (10 points)

Un ordinateur a une mémoire principale constituée de 16 Mo. Il possède un cache associatif par ensemble de 8 lignes, avec 2 entrées par ensemble. La taille d'un bloc est de 32 octets. On suppose que l'algorithme LRU est utilisé pour le remplacement d'un bloc, et que le temps d'accès au cache est de 1 ns alors que le temps d'accès à la RAM est de 10 ns.

1. Quelle est la capacité du cache ? Quel est le nombre de bits nécessaires pour décrire une adresse de la mémoire principale ?
2. Calculer le nombre de bits pour les champs *Etiquette*, *Ligne* et *Deplacement* de l'adresse d'un mot de la mémoire principale
3. Le cache est initialement vide. Le processeur lit **513** octets à partir des adresses 0, 1, 2, ..., 512 dans cet ordre, puis relit ces mêmes **513** octets dans le même ordre.
  - a. Décrire le contenu du cache après lecture des 513 premiers octets et calculer le temps nécessaire à leur lecture.
  - b. Décrire le contenu du cache après lecture des 513 octets suivants et calculer le temps nécessaire à leur lecture.
4. On considère le programme C suivant :

```
for (i=0;i<100;i++) {tab[i]=i; tab[i]=2*tab[i];}
```

  - a. On suppose que le tableau d'entiers `tab` est stocké en mémoire à partir de l'adresse 0 et qu'un entier est stocké sur 4 octets. Indiquer quelles sont les adresses accédées par ce programme et si elles sont accédées en lecture ou en écriture.
  - b. On suppose que l'on fait du WriteBack pour la gestion du cache. Combien d'écritures en mémoire centrale ont été évitées par rapport à la solution Write Through ?