

## CORRECTION

### Exercice 3 (10 points)

1. Capacité =  $2 \times 8 \times 32 = 512$  octets – 24 bits
2. Etiquette = 16 bits, Ligne = 3 bits et Déplacement = 5 bits
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 octets dans le même ordre.
  - a. après lecture des 513 octets

512..543	256..287
32..63	288..319
...	...
...	...
...	...
...	...
...	...
124..255	480..511

Après le premier passage le cache est ainsi. Au second passage tous les octets sont déjà dans le cache sauf le premier bloc, le contenu devient donc

256..287	512..543
32..63	288..319
...	...
...	...
...	...
...	...
...	...
124..255	480..511

- b. Pour le premier passage : chaque case du cache se remplit suite à un échec et ensuite viennent 31 succès. Le temps d'accès pour une case est donc  $(10 + 31 \times 1) = 41$  ns. Comme il y a 16 cases, cela fait  $16 \times 41 = 656$  ns. Il faut y ajouter l'échec de l'accès à l'octet 512 donc  $656 + 10 = 666$  ns. Pour le second

passage il n'y a que des succès sauf pour les blocs de la première ligne où il y a 3 échecs, le temps d'accès pour le second passage est donc  $510 * 1\text{ns} + 3 * 10 = 540\text{ns}$ .

4. a- tab est stocké en mémoire depuis l'adresse 0 jusqu'à l'adresse 399. Les adresses accédées sont donc 0, 3, 7, ... ,397. Elles sont accédées chacune 1 fois en lecture et 2 fois en écriture.

b- en Write through , il y a un accès mémoire pour chaque adresse accédée en écriture donc  $100 * 2 = 200$  accès en écriture.

En Write back : il y a un accès mémoire pour chaque bloc modifié en écriture. Or tab est stocké sur  $400/32 = 13$  blocs donc il y a 13 accès en écriture.

Bilan : on économise  $200 - 13 = 187$  accès en écriture en mémoire centrale