

# IFT 1215 H21 - Introduction aux systèmes informatiques

## Démonstration Semaine 10 - Solutions

Blanche Mongeon

01 avril 2021

### Exercice 1

Si l'adresse de la mémoire physique est de 32 bits, si la cache (à correspondance directe) est de 256 KiB (1 KiB =  $2^{10}$  octets), et si les blocs sont de 128 octets, quel est le nombre de bits des champs « étiquette », « index » et « offset » de l'adresse mémoire ?

Dans une cache à accès direct, les adresses mémoires sur  $X$  bits sont séparés ainsi :

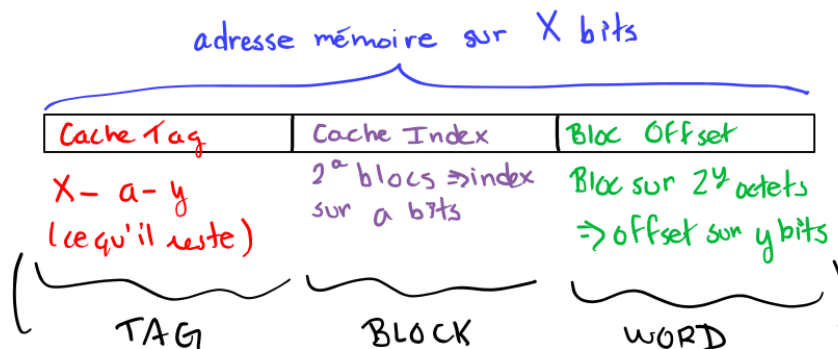


FIGURE 1 – Adresse mémoire

Ici, les blocs sont de  $128 = 2^7$  octets ; le champ offset est donc sur 7 bits. De plus, la mémoire cache est de 256 KiB =  $256 \cdot 2^{10}$  octets. On a donc

$$\frac{256 \cdot 2^{10} \text{ octets}}{2^7 \text{ octets/bloc}} = 256 \cdot 2^3 \text{ blocs} = 2^8 \cdot 2^3 \text{ blocs} = 2^{11} \text{ blocs}$$

Le champs index prends donc 11 bits. Finalement, puisque l'adresse mémoire physique est de 32 bits, alors le champs étiquette prends  $32 - 7 - 11 = 14$  bits.

### Exercice 2

On a une mémoire dont le temps de latence, i.e. le temps moyen pour trouver le premier mot, est de 40 ns ; le temps de transfert des mots suivants est 10 ns chacun.

- a) Quel est le temps (espéré) de chargement d'une ligne (le temps pour copier une ligne dans le cache) si le cache a des lignes de 16 mots ?

Le premier mot nécessitera 40 ns, et les 15 mots suivants nécessitent chacun 10 ns. Ainsi, le temps espéré de chargement d'une ligne est de  $40\text{ns} \cdot 1 + 10\text{ns} \cdot 15 = 190\text{ns}$ .

- b) Quel est le temps de chargement si les lignes sont de 8 mots ?

Le premier mot nécessitera 40 ns, et les 7 mots suivants nécessitent chacun 10 ns. Ainsi, le temps espéré de chargement d'une ligne est de  $40\text{ns} \cdot 1 + 10\text{ns} \cdot 7 = 110\text{ns}$ .

### Exercice 3

Un programme consiste en deux boucles imbriquées. Soit la structure générale suivante :

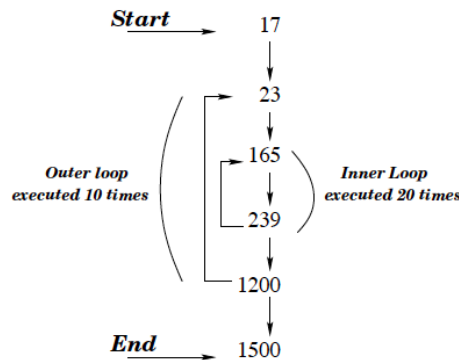


FIGURE 2 – Structure générale d'un certain programme avec deux boucles imbriquées

Les adresses mémoires sont en décimal. À l'exception des deux branches aux adresses 239 et 1200, toutes les instructions utilisent le *straight-line sequencing* et chaque adresse contient une instruction. On exécute le programme sur un ordinateur avec les paramètres suivants :

Taille de la mémoire principale	64 K mots
Taille de la mémoire cache	1 K mots
Taille d'un bloc	128 mots

Un cycle dans la mémoire principale prends  $10\tau s$  et un cycle dans la mémoire cache prends  $1\tau s$  ( $\tau = 10^{-9}$  seconde).

- a) Déterminer le nombre de bits dans les champs TAG, BLOCK et WORD (TAG, INDEX et OFFSET) d'une adresse mémoire.

Un bloc a  $128 = 2^7$  mots, ce qui implique que le champ offset/WORD a 7 bits. Aussi, si la cache contient  $1K = 2^{10}$  mots et que chaque bloc est de  $2^7$  mots, alors on a

$$\frac{2^{10} \text{ mots}}{2^7 \text{ mots/bloc}} = 2^3 \text{ blocs}$$

On a donc besoin de 3 bits pour la champ index/BLOCK. Finalement, puisque la mémoire principale est de  $64K \text{ mots} = 2^6 \cdot 2^{10} \text{ mots} = 2^{16} \text{ mots}$ , alors on a des adresses sur 16 bits. Ainsi, le champ TAG est sur  $16 - 3 - 7 = 6$  bits.

- b) Calculer le nombre de temps total nécessaire pour l'accès aux instructions (*instruction fetching*) durant l'exécution du programme.

Pour faire ce numéro, il est important de comprendre que la cache contient **8 blocs**. Ainsi, à partir du début jusqu'à la fin de la première itération de la boucle externe, on va remplir les 8 blocs pour les adresses  $0 - 1023$ , puis on va ensuite transférer les adresses  $1024 - 1151$  dans le bloc 0 puis les adresses  $1152 - 1279$  dans le bloc 1. À chacune des 9 autres itérations de la boucle externe, les blocs 0 et 1 vont chacun changer 2 fois (au début pour les adresses  $0 - 127$  et  $128 - 255$ , et à la fin pour les adresses  $1024 - 1151$  et  $1152 - 1279$ ). Finalement, à la fin du programme, on va transférer les adresses  $1280 - 1407$  dans le bloc 2 et les adresses  $1408 - 1535$  dans le bloc 3. Ainsi, le temps pris pour transférer des blocs de la mémoire principale à la cache est

$$(10 + 9 \cdot 4 + 2) \text{ transferts de blocs} \cdot \frac{128 \text{ mots}}{\text{blocs}} \cdot 10\tau s = 61440\tau s$$

Maintenant, si on regarde la boucle externe sans la boucle interne, on a  $[1200 - 23 + 1 - (239 - 165 + 1)]$  accès à la mémoire cache par itération (il y a 10 itérations) et qui prenne chacun  $1\tau s$ . Ainsi, on a un temps de  $11030\tau s$ .

Pour la boucle interne, celle-ci comprends  $239 - 165 + 1$  opérations par itérations, et il y a 20 itérations par itération de la boucle externe. Ainsi, on a un temps de  $(239 - 165 + 1) \cdot 20 \cdot 10\tau s = 15000 \text{ secondes}$ .

Finalement, au début on a les  $22 - 17 = 5$  accès à la mémoire cache, puis les  $1500 - 1200 + 1 = 301$  accès à la fin, ce qui prends en tout  $306\tau s$ . Le temps total est donc :

$$306\tau s + 11030\tau s + 15000\tau s + 61440\tau s = 87776\tau s$$