

TD d'architecture des ordinateurs VI

Temps d'accès aux données

Exercice 1 : Temps d'accès

On considère un ordinateur disposant d'une hiérarchie mémoire à trois niveaux M1, M2 et M3 (Voir Figure 1).

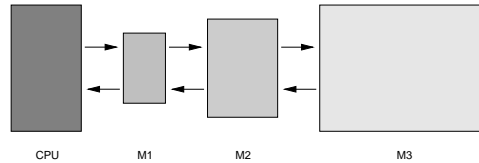


FIG. 1 – Hiérarchie mémoire

On donne les caractéristiques suivantes pour chaque mémoire :

	Capacité S_i (bit)	Coût C_i (Fr/bit)	Temps T_i d'Accès (s)	Hit Ratio
M1	1024	1.0	10^{-8}	0.9
M2	2^{16}	0.1	10^{-6}	0.9999
M3	2^{24}	0.01	10^{-3}	1.0

Remarques :

- H_i est le Hit Ratio et représente la proportion d'accès mémoire réussis au niveau i (présence de la donnée).
- Dans les temps d'accès T_i , le temps de chargement d'un bloc en cas d'échec est pris en compte.

1. Calculer le coût total de la mémoire C_T .

Réponse _____

$$C_T = \sum_{i=1}^3 S_i C_i = 1024 \times 1.0 + 2^{16} \times 0.1 + 2^{24} \times 0.01 = 175349.76 \text{ Fr}$$



2. Calculer le coût moyen par bit C_M .

Réponse _____

$$C_M = \frac{C_T}{\sum_{i=1}^3 S_i} = 0.0104 \text{ Fr/bit}$$



3. Notons P_i la probabilité qu'une donnée soit présente au niveau i et absente des niveaux j avec $j < i$. Compléter le tableau avec les valeurs de P_i .

Réponse _____

$$P_1 = H_1 = 0.9$$

$$P_2 = H_2 - P_1 = 0.9999 - 0.9 = 0.0999$$

$$P_3 = H_3 - (P_1 + P_2) = 1.0 - (0.9 + 0.0999) = 0.0001$$



4. Calculer le temps d'accès moyen T_M à une information.

Réponse _____

$$T_M = P_1 \times T_1 + P_2 \times T_2 + P_3 \times T_3 = 21 \times 10^{-8} \text{ s}$$



Exercice 2 : Temps d'accès (suite)

Soient les caractéristiques suivantes des différentes mémoires :

niveau	TA_i	P_i	TB_i
M1	10^{-6}	0.9999	
M2	10^{-5}	0.000099	0.001
M3	10^{-3}	0.000001	0.1

où TA_i est le temps d'accès à une donnée du niveau i , P_i la probabilité de trouver une donnée au niveau i et pas avant et TB_i le temps nécessaire au transfert d'un bloc du niveau i au niveau $i-1$.

- Calculer le temps d'accès moyen en supposant que :
 - en cas d'échec, le temps d'accès est la somme des temps pour savoir si le bloc est présent, pour accéder au 1er octet au niveau supérieur et pour transférer le bloc.
 - en cas d'échec, l'UC est arrêtée jusqu'à ce que le bloc soit chargé.
 - le temps de sauvegarde du bloc remplacé est nul.

Réponse _____

$$TA = P_1 \times T_1 + P_2 \times T_2 + P_3 \times T_3$$

avec :

$$T_1 = TA_1$$

$$T_2 = TA_1 + TA_2 + TB_2$$

$$T_3 = TA_1 + TA_2 + TA_3 + TB_3 + TB_2$$

$$\text{A.N. : } T_1 = 10^{-6} \text{ s}, T_2 = 1.011 \times 10^{-3} \text{ s et } T_3 = 0.102011 \text{ s.}$$

$$\text{Donc } TA = 1.202 \times 10^{-6} \text{ s.}$$



- On veut obtenir un temps TA inférieur ou égal à $1.15 \times 10^{-6} \text{ s}$ en remplaçant M3 par une mémoire M31 plus rapide offrant un temps de transfert TB_{31} dont vous devez calculer la valeur maximale.

Réponse _____

Il suffit d'isoler TB_3 dans l'expression de TA et de résoudre l'inéquation proposée.

On obtient alors une valeur de $TB_{31} \approx 5 \times 10^{-2} \text{ s}$.



Exercice 3 : Cache associatif par ensembles de blocs

On considère une architecture comportant un cache associatif par ensembles de blocs et utilisant des adresses sur 22 bits pour accéder linéairement à l'espace d'adressage. L'unité d'adressage est l'octet et le cache est structuré en 64 ensembles de 4 blocs (4-way associatif), chaque bloc contenant 32 octets.

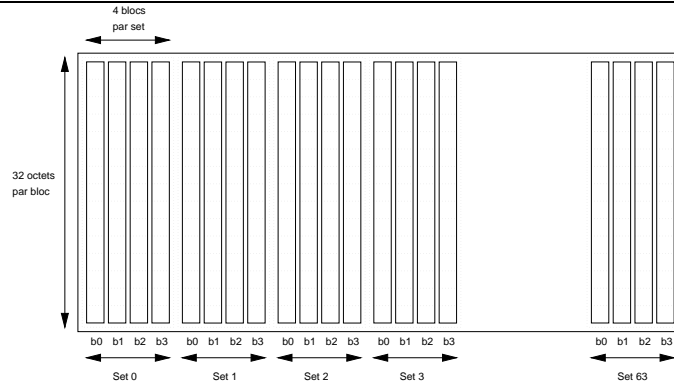
- Représenter schématiquement le cache.

Réponse _____

Voir Figure-Réponse 1.



FigRep 1 Organisation du cache



2. Expliquer comment une donnée est accédée dans le cache à partir de son adresse sur 22 bits.

Réponse _____

L'adresse peut être vue comme suit ; une partie adressant le bloc et un offset repérant le mot à l'intérieur du bloc.

Adresse du bloc	Offset
17 bits	5 bits

L'offset tient sur 5 bits pour pouvoir adresser $2^5 = 32$ mots mémoire. Il reste donc 17 bits pour repérer le bloc.

Au niveau du cache, l'unité de gestion de la mémoire doit accéder rapidement à l'ensemble contenant les blocs candidats ; il faut donc pouvoir retrouver ce numéro d'ensemble à partir de l'adresse. Puisque nous avons 64 ensembles, 6 bits sont nécessaires à ce codage ($2^6 = 64$). La dernière composante de l'adresse servira à déterminer si le bloc cherché est effectivement chargé dans l'ensemble cible, c'est le tag. D'où l'organisation de l'adresse : 11 bits de tag, 6 bits d'index pour l'ensemble et 5 bits d'offset pour un total de 22 bits organisés comme suit :

Adresse du bloc	Index	Offset
11 bits	6 bits	5 bits



3. Soit la suite de références suivante :

1AA050 1AA853 2E3057 3FF85F 1AA04E

En supposant un remplissage quelconque du cache avant le premier accès de la séquence, représenter l'emplacement de ces références dans le cache (indiquer le mode de remplissage en cas de conflit : random ou LRU).

Réponse _____

Chaque chiffre hexadécimal est codé par 4 bits, le décodage des adresses virtuelles se fait donc comme suit :

Adresse :

1	A	A	0	5	0
0001	1010	1010	0000	0101	0000
0001 1010 1010			0 000010		10000
Tag			Index		Offset

En procédant de même pour chaque référence, on constate que toutes les références désignent un mot d'un bloc du deuxième ensemble. Le mot mémoire

recherché doit se trouver dans l'ensemble 2, son offset vaut 16. Pour savoir si le bloc est effectivement présent dans le cache, l'unité de gestion de celui-ci va comparer le tag de l'adresse réelle avec chacun des tags associés aux blocs de l'ensemble 2. Si le bloc est absent, il sera cherché en mémoire principale et remplacera un des blocs de l'ensemble en suivant la stratégie de remplacement choisie.

