

Examen de semestre – Corrigé type
06/01/2025



Documents, Calculatrice et Téléphone Portable ne sont pas autorisés;

Durée ⌚ 01h30.

Questions : (05 pts) (Voir Cours...)

Q1 : Quelles sont les formes de **localité** ? (3pts)

R1)

- **Localité spatiale** : indique que l'accès à une instruction située à une adresse va probablement être suivie d'un accès à une zone toute proche de cette adresse, copier des blocs de mots dans le cache.
- **Localité temporelle** : conserver les informations pendant quelque temps dans le cache en raison de la forte probabilité de se reproduire dans la suite du programme.

Q2 : Expliquez les quatre **algorithmes de remplacement** des blocs de la mémoire cache. (2pts)

R2)

- **Random** : Choisir un bloc candidat de manière aléatoire
- **FIFO** (First In First Out) : Choisir le plus ancien bloc du cache
- **LRU** (Least Recently Used) : Choisir le bloc le moins récemment utilisé
- **LFU** (Least Frequently Used) : Choisir le bloc le moins fréquemment utilisé

Exercice 01 : (07 pts)

A) Supposons qu'un ordinateur utilise une mémoire cache totalement associative (*Fully-associative*) avec 2^{24} mots de mémoire principale et une mémoire cache de 128 blocs, où chaque bloc contient 64 mots.

Q1 : Combien y a-t-il de blocs de mémoire principale ? (1pt)

R1) Un bloc de mémoire principale contient 64 mots. Pour calculer le nombre de blocs de mémoire principale, on divise le nombre total de mots de mémoire principale par le nombre de mots par bloc :

$$\text{Nombre de blocs de mémoire principale} = \frac{\text{Nombre total de mots}}{\text{Mots par bloc}}$$

$$\text{Nombre de blocs de mémoire principale} = \frac{2^{24}}{64} = 2^{24-6} = 2^{18} = 262144$$

Q2 : Quel est le format d'une adresse mémoire vue par la mémoire cache, c'est-à-dire quelles sont les tailles des champs « tag ou étiquette » et « mot » ? (1.5pt)

R2) Dans une mémoire cache totalement associative, une adresse mémoire est divisée en deux champs :

- Champ "tag ou étiquette" : Identifie le bloc de mémoire principale.
- Champ "mot" : Identifie la position d'un mot dans un bloc.

Taille du champ "mot" :

Chaque bloc contient 64 mots. Pour adresser ces mots, il faut :

$$\text{Nombre de bits pour le champ mot} = \log_2(2^6) = 6 \text{ bits}$$

Taille du champ "étiquette ou tag" :

L'adresse mémoire totale est de $\log_2(2^{24}) = 24$ bits

En retirant les **6 bits** du champ "mot", il reste le nombre de bits pour le champ **étiquette ou tag** = $24 - 6 = 18$ bits.

Q3 : À quel bloc de cache la référence mémoire $01D872_{16}$ sera-t-elle mappée ? (1.5pt)

R3)

Conversion de $01D872_{16}$ en binaire : $0000\ 0001\ 1101\ 1000\ 0111\ 0010_2$, une adresse mémoire sur 24 bits.

Les premiers 6 bits de faible poids représentent le champ "mot" : $110010_2 = 50$ en décimal

Le reste, soit les 18 bits les plus significatifs, représente le champ "étiquette" : $000000011101100001_2 = 1889$ en décimal

B) Supposons que cet ordinateur utilise une mémoire cache à correspondance directe (Direct-mapped).

Q1 : Quel est le format d'une adresse mémoire vue par la mémoire cache, c'est-à-dire quelles sont les tailles des champs « tag ou étiquette », « bloc » et « mot » ? (1.5pt)

R1) Dans une mémoire cache à correspondance directe, une adresse mémoire est composée de trois champs :

- Champ mot : Identifie le mot au dans un bloc.
- Champ bloc : Identifie le bloc dans la mémoire cache.
- Champ tag ou étiquette: Correspond au bloc de mémoire principale demandé.

Taille du champ mot :

Le bloc contient 64 mots. Le champ mot doit donc pouvoir adresser $64=2^6$ mots.

Taille du champ mot = 6 bits

Taille du champ bloc:

Il y a $128=2^7$ blocs en mémoire cache. Le champ bloc doit donc adresser 128 blocs.

Taille du champ bloc = 7 bits

Taille du champ tag ou étiquette :

L'adresse mémoire contient 24 bits (car 2^{24} mots en mémoire principale). La taille du champ étiquette est obtenue en soustrayant la taille des champs mot et bloc :

Taille du champ étiquette = $24 - (6 + 7) = 11$ bits

Q2 : À quel bloc de cache sera mappée la référence mémoire $01DB63_{16}$? (1.5pt)

R2)

Convertir l'adresse $01DB63_{16}$ en binaire : $0000\ 0001\ 1101\ 1011\ 0110\ 0011_2$, sur 24 bits.

- Champ mot : Les 6 bits les plus à droite (bits 0 à 5). **Mot = 100011 = en décimal : 35**
- Champ bloc : Les 7 bits suivants (bits 6 à 12). **Bloc = 1101101 en décimal : 109**
- Champ étiquette : Les 11 bits restants (bits 13 à 23). **Étiquette = 0000001110 en décimal : 14**

Exercice 02 : (04 pts)

Considérons une machine avec des instructions de 32 bits et des adresses de 12 bits

- Si la machine prend en charge 254 instructions à 2-adresses, combien d'instructions à 1-adresse peuvent être formulées? (2 pts)

R1)

- **2-address instructions (instructions à 2 adresses) :**

Deux opérandes adresses (2) : $12 + 12 = 24$ bits

Code opération (opcode) = $32 - 24 = 8$ bits pour 2-address instructions

$2^8 = 256$ instructions 2-address, la machine prend en charge 254 instructions

Alors $256 - 254 = 2$, **deux adresses disponibles non utilisées**

- **1-address instructions (instructions à 1 adresse) :**

Un opérande adresse : 12 bits

Code opération (opcode) : $32 - (8 + 12) = 12$ bits pour les instructions 1-adresse

2^{12} instructions 1-adresse multiplier par 2 adresses non utilisées :

$= 2^{12} \times 2 = 2^{13} = 8192$ instructions 1-address

- Si la machine prend en charge 254 instructions à 2-adresses et 8000 instructions à 1-adresse, combien d'instructions à zéro adresse (0-adresse) peuvent être formulées ? (2pts)

R2)

– **0-address instructions (instructions à zéro adresse)**

Nbr bits de code opération = 32 bits – (Nbr de bit 2-address + Nbr de bit 1-address)

D'après la question précédant on a 8192 instructions 1-address et tant que la machine prend en charge 8000 instructions 1-address, alors $8192 - 8000 = 192$ instructions disponibles non utilisées.

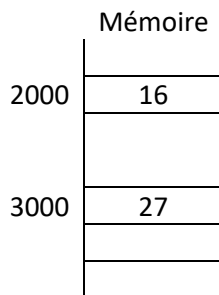
Code opération (opcode) : $32 - (8 + 12) = 12$ bits pour les instructions 0-adresse

2^{12} instructions 0-adresse multiplier par 192 adresses non utilisées :

$$= 2^{12} \times 192 = 2^{12} \times 2^5 \times 6 = 2^{17} \times 6 = 786\ 432 \text{ instructions 0-address}$$

Exercice 03 : (04 pts)

Considérez la partie de programme suivante. Ici, R1 et R2 sont des registres généraux. Supposons que le contenu de l'adresse mémoire 3000 est 27 et que celui de l'adresse 2000 est 16. Le contenu du registre R2 est 10. Tous les nombres sont exprimés en décimal. Après l'exécution de ce programme, quelle sera la valeur de R2?



	Instructions	Operations
	MOV R1, #7	$R1 \leftarrow \#7$
X:	DEC R1	$R1 \leftarrow R1 - 1$
	JNZ Y	Jump to Y if Non-Zero
	ADD R2, (3000)	$R2 \leftarrow R2 + M[3000]$
	JMP Z	Jump to Z
Y:	ADD R2, (2000)	$R2 \leftarrow R2 + M[2000]$
	JMP X	Jump to X
Z:	HALT	Stop

Registres	Itérations							
R1	7	6	5	4	3	2	1	0
R2	10	26	42	58	74	90	106	133

$$R2 = 10 + (16 \times 6) + 27 = 133$$

