

Examen de semestre – Corrigé type
17/01/2026

 **Documents, Calculatrice et Téléphone Portable ne sont pas autorisés;** Durée  **01h30.**

Exercice 01 : (05 pts)

Considérons deux processeurs différents, P1 et P2, exécutant le même jeu d'instructions. P1 a une fréquence de 3 GHz et un CPI de 1,5. P2 a une fréquence de 2,5 GHz et un CPI de 1,0.

Q1 : Quel processeur a la performance la plus élevée exprimée en instructions par seconde ? (1pt)
R1)

Pour mesurer la performance d'un processeur (le nombre d'instructions exécutées par seconde), nous partons de l'équation du temps d'exécution :

$$\text{Temps total} = (\text{CPI} \times \text{nombre d'instructions}) / \text{Fréquence du processeur}$$

Puisque la performance est le nombre d'instructions divisé par le temps, en simplifiant l'équation, nous obtenons la formule directe suivante :

$$\begin{aligned}\text{Performance} &= \text{Nombre d'instructions} / \text{Temps total} \\ &= \text{Fréquence du processeur} / \text{CPI}\end{aligned}$$

Appliquons cette formule aux deux processeurs :

$$\text{Performance P1} = 3 \times 10^9 / 1.5 = 2 \times 10^9 \text{ instructions/second}$$

$$\text{Performance P2} = 2.5 \times 10^9 / 1.0 = 2.5 \times 10^9 \text{ instructions/second}$$

Le processeur P2 est plus performant que le processeur P1

Q2 : Si chaque processeur exécute un programme en 10 secondes, trouvez le nombre de cycles et le nombre d'instructions. (3pts)

R2)

Pour mesurer le nombre de cycles et d'instructions, nous partons aussi de l'équation du temps d'exécution :

$$\text{Temps total} = (\text{CPI} \times \text{Nombre d'instructions}) / \text{Fréquence du processeur}$$

Alors nous obtenons les deux formules directes suivantes :

$$\begin{aligned}\text{Nombre total de cycles} &= \text{CPI} \times \text{Nombre d'instructions} \\ &= \text{Temps total} \times \text{Fréquence du processeur}\end{aligned}$$

$$\text{Nombre d'instructions} = \text{Nombre total de cycles} / \text{CPI}$$

– Pour le processeur P1 :

$$\text{Nombre total de cycles} = 10 \times (3 \times 10^9) = 30 \times 10^9 \text{ cycles}$$

$$\text{Nombre d'instructions} = 30 \times 10^9 / 1.5 = 20 \times 10^9 \text{ Instructions}$$

- Pour le processeur P2 :

$$\text{Nombre total de cycles} = 10 \times (2.5 \times 10^9) = 25 \times 10^9 \text{ cycles}$$

$$\text{Nombre d'instructions} = 25 \times 10^9 / 1.0 = 25 \times 10^9 \text{ Instructions}$$

Q3 : Nous essayons de réduire le temps d'exécution de 50%. Quelle est la fréquence nécessaire pour les deux processeurs pour obtenir cette réduction du temps ? (1pt)

R3)

Pour réduire le temps d'exécution de 50 %, le nouveau temps d'exécution doit être la moitié (1/2) du temps initial. Puisque le Temps est inversement proportionnel à la Fréquence (Temps = 1/Fréquence), pour réduire le temps de moitié, nous devons doubler la fréquence initiale.

$$\text{Nouveau temps} = \text{Temps initial} / 2$$

En remplaçant le temps dans l'équation, nous obtenons :

$$\text{CPI} \times \text{Nombre d'instructions} / \text{Nouveau Fréquence} = \text{CPI} \times \text{Nombre d'instructions} / 2 \times \text{Fréquence initiale}$$

- Pour le processeur P1 :

$$\text{Nouveau Fréquence} = 2 \times \text{Fréquence initiale} = 2 \times 3 \text{ GHz} = 6 \text{ GHz}$$

- Pour le processeur P2 :

$$\text{Nouveau Fréquence} = 2 \times \text{Fréquence initiale} = 2 \times 2.5 \text{ GHz} = 5 \text{ GHz}$$

NB :

$$\text{Temps total} = (\text{CPI} \times \text{nombre d'instructions}) / \text{Fréquence du processeur}.$$

Exercice 02 : (06 pts)

Supposons qu'une machine adressable par mots utilise une mémoire cache associative par ensembles à 2 lignes (2-way set-associative), avec 8 GB de mémoire principale et une mémoire cache de 128 KB, dont le bloc de la mémoire cache contient 64 mots de 32 bits.

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

R1)

Pour calculer le nombre de blocs de la mémoire cache, on divise la taille de cache par la taille de bloc.

$$\text{Nombre de bloc de mémoire cache} = \text{la taille de cache} / \text{la taille de bloc}$$

$$= \text{Nombre total de mots de mémoire cache} / \text{Mots par bloc}$$

bloc

$$\text{La taille de cache} = 128 \text{ KB}, \text{ et comme un mot} = 2^2 \text{ bytes}$$

Alors :

$$\text{La taille de cache} = 2^7 \times 2^{10} / 2^2 = 2^{15} \text{ mots}$$

$$\text{La taille de bloc} = 64 \text{ mots} = 2^6 \text{ mots}$$

$$\text{Nombre de bloc de mémoire cache} = 2^{15} / 2^6 = 2^9 \text{ blocs}$$

Q2 : Combien y a-t-il de blocs de mémoire principale ? (1pt)

R2)

De la même manière, nous divisons la taille de la mémoire principale par la taille d'un bloc.

Nombre de blocs de mémoire principale = Nombre total de mots de mémoire principale / Mots par bloc

$$\begin{aligned}\text{La taille de la mémoire principale} &= 8 \text{ GB} = 2^3 \times 2^{30} / 2^2 \\ &= 2^{31} \text{ mots}\end{aligned}$$

$$\text{La taille de bloc} = 64 \text{ mots} = 2^6 \text{ mots}$$

Alors,

$$\text{Nombre de bloc de mémoire principale} = 2^{31} / 2^6 = 2^{25} \text{ blocs}$$

Q3 : 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 », « ensemble ou Set » et « mot » ? (3pts)

R3)

Dans une mémoire cache associative par ensembles à 2 lignes (2-way set-associative), une adresse mémoire est divisée en trois champs:

- "tag ou étiquette" : Identifie le bloc de mémoire principale.
- "ensemble ou Set" : Identifier l'ensemble des lignes du cache
- "mot" : Identifie la position d'un mot dans un bloc mémoire.

Taille du champ "mot" :

Un 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 "Set" :

On a Nombre de bloc de mémoire cache = 2^9 blocs et quand la mémoire cache associative par ensembles à 2 lignes on divise le nombre de bloc par 2

$$\text{Nombre des ensembles} = 2^9 / 2 = 2^8 \text{ ensemble ou Set}$$

$$\text{Nombre de bits pour le champ Set} = \log_2 (2^8) = 8 \text{ bits}$$

Taille du champ "étiquette ou tag" :

La taille de la mémoire principale déjà calculer en nombre de mots = 2^{31}

L'adresse mémoire totale est de $\log_2(2^{31}) = 31$ bits

En retirant les 6 bits du champ "mot", et 8 bits du champ Set, il reste le nombre de bits pour le champ étiquette ou tag = $31 - (6+8) = 17$ bits.

Q4 : À quel bloc de cache sera mappée la référence mémoire : $(012A53)_{16}$? (1pt)

R4)

Conversion de $(012A53)_{16}$ en binaire sur une adresse mémoire de 31 bits.

$$= (000\ 0000\ 0000\ 0001\ 0010\ 1010\ 0101\ 0011)_2$$

Interprétation des champs :

Les premiers 6 bits de faible poids représentent le champ "mot" : 01 0011 = 19 en décimal

Les 8 bits après les bits de champ mot représentent le champ "Set" : 10 1010 01 = 169 en décimal

Le reste, soit les 17 bits les plus significatifs, représente le champ "étiquette" :
000 0000 0000 0001 00 = 4 en décimal

Exercice 03 : (04 pts)

Soit l'expression suivante : $A = (B + C) \times (D - E)$

Ecrire le code assembleur de cette expression en une suite d'instructions en format à :

Trois (03) adresses, Deux (02) adresses, Une (01) adresse et Zéro (0) adresse.

R)

(03) adresses	(02) adresses	(01) adresse	(0) adresse
Add R1, B, C	Move R1, B	Load B	Push B
Sub R2, D, E	Add R1, C	Add C	Push C
Mult A, R1, R2	Move R2, D	Store Temp	Add
	Sub R2, E	Load D	Push D
	Mult R2, R1	Sub E	Push E
	Store A, R2	Mult Temp	Sub
		Store A	Mult
			Pop A

NB : Utiliser le jeu d'instructions : Load, Move, Add, Sub, Mult, Store, Push, Pop

Exercice 04 : (05 pts)

Considérons une machine dotée d'une mémoire principale adressable par mot de 32 bits. Son jeu d'instructions (2-adresses), comprend 250 opérations différentes. Chaque instruction comporte une partie code d'opération (opcode) et deux parties adresse (ne permettant que des instructions à deux opérandes). Chaque instruction occupe un mot mémoire.

Q1. Combien de bits sont nécessaires pour l'opcode ? (1pt)

R1)

Pour coder 250 opérations distinctes, il faut trouver la puissance de 2 supérieure ou égale à 250.

On a $2^8 = 256$ suffisant pour les 250 instructions

Donc, il faut 8 bits pour l'opcode.

Q2. Combien de bits restent-ils pour l'adresse (operand)? (1pt)

R2)

Chaque instruction occupe un mot mémoire, la taille totale de l'instruction fait 32 bits. Nous devons soustraire la taille de l'opcode, puis diviser le reste par deux (car c'est une instruction à 2 adresses)

Alors : $32 - 8 = 24$ bits pour les deux opérandes (adresses)

Le nombre de bits pour le champ adresse = $24/2 = 12$ bits.

Q3. Quelle est la taille maximale de cette mémoire ? (1pt)

R3)

La taille de la mémoire est déterminée par la taille du champ d'adresse. Avec 12 bits d'adresse, on peut adresser 2^{12} mots ou emplacements mémoire différents.

Alors : la taille maximale de la mémoire est de 2^{12} mots (4K mots) = 16 KB

Q4. Donner l'architecture réduite de cette mémoire (c-à-d taille de: CO, MAR, MBR, Registre d'instruction, Bus d'adresse, Bus de données). (1pt)

R4)

Nous devons définir la taille des registres et des bus en fonction de la taille du mot (données) et de la taille de l'adresse et d'instruction.

- Pour le CO, il faut 12 bits, parce que il est lié à l'adresse.
- MAR (Memory Address Register) : 12 bit, il est lié à l'adresse
- Bus d'adresse : 12 bits
- MBR (Memory Buffer Register) : 32 bits, il est lié au mot de données en mémoire.
- Bus de données : 32 bits, il est lié au mot de données
- RI (Registre d'Instruction) : 32 bits, Chaque instruction occupe un mot mémoire.

Q5. Quel est le plus grand nombre binaire non signé pouvant être stocké dans un mot mémoire?

(1pt)

R5)

Pour un mot mémoire de 32 bits, le plus grand nombre binaire non signé est atteinte lorsque tous les 32 bits sont à 1.

Le plus grand nombre binaire non signé = $2^{32} - 1$

