



Examen – Architecture des ordinateurs

(Durée 1h30mn)

Le 02 /02 / 2020

Questions de cours (5,5 pts)

1. Enumérer brièvement les différentes politiques de remplacement possibles lors d'un défaut de cache. Quelle est la plus utilisée en pratique et quelle est la plus facile à implémenter ?
2. Actuellement l'architecture des microprocesseurs se composent de deux grandes familles :
 - L'architecture CISC (Complex Instruction Set Computer)
 - L'architecture RISC (Reduced Instruction Set Computer)
 Citer quatre (4) différences entre l'architecture CISC et l'architecture RISC.
3. Expliquer brièvement le principe de la mémoire cache.

Exercice 1 : (5 points)

Soit une machine dotée d'une mémoire centrale de 1024 K mot de 32 bits.

- 1- Combien de bits, d'octets, de Kilo octets et de Méga octets contient cette mémoire
- 2- Combien de valeur différente peut prendre un mot de cette mémoire ?
- 3- Déterminer la plage d'adressage de cette mémoire (en hexadécimale/ base 16)
- 4- On veut stocker sur cette mémoire des nombres réels où chaque nombre est représenté sur 64 bits. Calculer l'adresse du 9^{ème} nombre sachant que le premier est stocké à l'adresse FF₍₁₆₎.

Exercice 2 : (5.5 points)

L'ordinateur dont il est question ici possède une architecture dont les instructions machines possèdent un **seul opérande (machine à une adresse)** et dont le mode d'adressage est **immédiat**. Les mnémoniques à considérer, pour les instructions arithmétiques, sont ADD- SUB- MUL et DIV, et pour les instructions de manipulation de données LOAD et STORE.

- 1- Expliquer l'étape de recherche de l'opérande et d'exécution de l'instruction suivante : 5 SUB 22. (Phase 2 des étapes d'exécution de l'instruction)
- 2- Donner le code nécessaire pour évaluer l'expression : $Z = (A+B/C) * (D+E)$ en utilisant le minimum d'instruction possible.
- 3- En supposant que l'architecture de la machine est à **pile**, donner le code pour évaluer la même expression et ce en utilisant le minimum d'instruction possible.

Exercice 3 : (4 points)

Les instructions exécutées sur un processeur peuvent être classées en trois catégories selon le nombre de cycles d'horloge (CCi) nécessaires pour leur exécution.

Catégorie	Nombre de cycles d'horloge CCi
A	1
B	2
C	3

Deux compilateurs différents génèrent, pour un même programme, les mélanges d'instructions suivantes. Ce nombre d'instructions pour chaque catégorie est indiqué dans le tableau suivant :

Compilateur	Catégorie A	Catégorie B	Catégorie C
IC(X)	5.000.000.000	1.000.000.000	1.000.000.000
IC(Y)	10.000.000.000	1.000.000.000	1.000.000.000

- 1- En supposant que la fréquence d'horloge du processeur est de 500 MHz, donner la valeur du CPI pour chaque cas.
- 2- Lequel des deux mélanges va s'exécuter le plus rapidement?
- 3- Quelle est la performance en MIPS (Millions d'Instructions Par Seconde) dans les deux cas?

Bon courage

Corrigé Examen –Architecture des ordinateurs

(du 02 /02 / 2020)

Questions de cours (5,5 pts)

1. Réponse 1 : (2,5 points (0.5*4 + 0.25*2))

Si le cache est plein et que le processeur a besoin d'un bloc qui n'est pas dans le cache, il faut remplacer un des blocs du cache. Diverses stratégies sont employées, principalement :

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

Les stratégies concernant l'utilisation (LFU, LRU) sont les plus efficaces (vient ensuite la stratégie aléatoire). Les stratégies aléatoires et FIFO sont plus faciles à implanter.

2. Réponse 2 : La différence entre les architectures CISC et RISC : (2 points (0.25*8))

Architecture RISC	Architecture CISC
instructions simples ne prenant qu'un seul cycle	instructions complexes prenant plusieurs cycles
instructions au format fixe	instructions au format variable
décodeur simple (câblé)	décodeur complexe (microcode)
beaucoup de registres	peu de registres
seules les instructions LOAD et STORE ont accès à la mémoire	toutes les instructions sont susceptibles d'accéder à la mémoire
peu de modes d'adressage	beaucoup de modes d'adressage
compilateur complexe	compilateur simple

3. Réponse 3 : Explication du principe de la mémoire cache : (1 point (0.5*2))

Le principe de cache est très simple :

- Soit la donnée ou l'instruction requise est présente dans le cache et elle est alors envoyée directement au microprocesseur. On parle de succès de cache. (a)
- Soit la donnée ou l'instruction n'est pas dans le cache, et le contrôleur de cache envoie alors une requête à la mémoire principale. Une fois l'information récupérée, il la renvoie au microprocesseur tout en la stockant dans le cache. On parle de défaut de cache. (b)

Exercice 1 : (5 points)

Soit une machine dotée d'une mémoire centrale de 1024 K mot de 32 bits.

1- Combien de bits, d'octets, de Kilo octets et de Méga octets contient cette mémoire (1.5 point)

Capacité (bits) = Nombre de mots * taille du mot = 1024 K * 32 = $2^{10} * 2^5 = 2^{25}$ bits (0.75)

Cap(octs) = $2^{25} / 2^3 = 2^{22}$ octs (0.25) || Cap(Kocts) = $2^{22} / 2^{10} = 2^{12}$ octs (0.25) ||

Cap(Mocts) = $2^{12} / 2^{10} = 2^2$ Moct (0.25)

2- Combien de valeur différente peut prendre un mot de cette mémoire ? **(0.5 point)**

2^{32} valeurs

3- Déterminer la plage d'adressage de cette mémoire (en hexadécimale/ base 16) **(1.5 point)**

Nombre de mots = $2^{\text{nombre de lignes d'adresses}} = 1024K = 2^{10} * 2^{10} = 2^{20} \rightarrow$ nombre de lignes d'adresse = 20
(0.5pt)

adresse minimale (0000000000..) sur 20 bits = 00000(16)(0.5)

et adresse maximale ($2^{20} - 1$) = 11111111 ... 1111 sur 20 bits = FFFFF(16) (0.5 pt)

4- On veut stocker sur cette mémoire des nombres réels et chaque nombre est représenté sur 64 bits.
Calculer l'adresse du 9^{ème} nombre sachant que le premier est stocké à l'adresse FF(16) **(1.5 point)**

Rep : chaque nombre dans la mémoire prend 2 mots mémoire (taille du nombre réel / taille du mot = 64/32 = 2 mot)

@n^{ème} nombre = @1^{er} nombre + (nombre de mots de chaque nombre) * (n-1)

@1^{er} nombre = FF(16) = (15*16 + 15)₍₁₀₎ = 255₍₁₀₎

@9^{ème} nombre = @1^{er} nombre + (8 * 2) = 255₍₁₀₎ + 16₍₁₀₎ = 271₍₁₀₎ = 10F(16)

Exercice 2 : (5.5 points)

L'ordinateur dont il est question ici possède une architecture dont les instruction machines possèdent un **seul opérande (machine a une adresse)** dans le mode d'adressage est **immédiat**. Les mnémoniques à considérer, pour les instructions arithmétiques, sont ADD- SUB- MUL et DIV, et pour les instructions de manipulation de données LOAD et STORE.

1- Expliquer l'étape de recherche de l'opérande et d'exécution de l'instruction suivante : 5 SUB 22. (Phase 2 des étapes d'exécution de l'instruction) (1 pt)

Phase 2 : Traitement de l'instruction (soustraction du contenu de l'accumulateur avec la valeur)

- Transfert de l'opérande (valeur contenue dans le RI) dans l'UAL c'est-à-dire

$UAL \leftarrow (RI).ADR$ ou $UAL \leftarrow (ADOP)$ (0.5)

- Commande de l'exécution de l'opération (soustraction) : $ACC \leftarrow (ACC) + \text{Valeur}$ (0.5)

2- Donner le code nécessaire pour évaluer l'expression : $Z = (A+B/C) * (D+E)$ en utilisant le moins d'instruction possible. **(2 points (8*0.25))**

LOAD B; [Acc] \leftarrow B

DIV C; [Acc] \leftarrow B/C

ADD A; [Acc] \leftarrow B/C + A

STORE Z; [Z] \leftarrow A + B/C

LOAD D; [Acc] \leftarrow D

ADD E; [Acc] \leftarrow D + E

MUL Z; [Acc] \leftarrow (D + E) * (A + B/C)

STORE Z; [Z] \leftarrow (D + E) * (A + B/C)

3- En supposant que l'architecture de la machine est à **pile**, donner le code pour évaluer la même expression en utilisant le minimum d'instruction possible. **(2.5 points (10*0.25))**

PUSH A; Pile = { A }
 PUSH B; Pile = { A; B }
 PUSH C; Pile = { A; B; C }
 DIV; Pile = { A; B/C }
 ADD; Pile = { A+(B/C) }
 PUSH D; Pile = { A+(B/C); D }
 PUSH E; Pile = { A+(B/C); D; E }
 ADD; Pile = { A+(B/C); D+E }
 MUL; Pile = { (A+(B/C))*(D+E) }
 POP Z; Pile = { } et [Z] ← (A+B/C)*(D+E)

Exercice 3 : (4 points)

R1. Calcul du CPI :

$$CPI = \frac{\sum_i (IC_i)(CC_i)}{IC} \dots\dots\dots(0.50 \text{ pt})$$

$$CPI(X) = \frac{5 \times 10^9 \times 1 + 1 \times 10^9 \times 2 + 1 \times 10^9 \times 3}{5 \times 10^9 + 1 \times 10^9 + 1 \times 10^9} = \frac{5+2+3}{7} = \frac{10}{7} = 1,428 \dots\dots\dots(0.25 \text{ pt})$$

$$CPI(Y) = \frac{10 \times 10^9 \times 1 + 1 \times 10^9 \times 2 + 1 \times 10^9 \times 3}{10 \times 10^9 + 1 \times 10^9 + 1 \times 10^9} = \frac{10+2+3}{12} = \frac{15}{12} = 1,25 \dots\dots\dots(0.25 \text{ pt})$$

R2. Pour donner le mélange qui va s'exécuter plus rapidement, nous devons tout d'abord calculer le temps du cycle horloge pour chaque compilateur.

F= 500Mhz : $500 * 10^6 \text{ op} \rightarrow 1 \text{ seconde}$

1 op → T cycle

⇒ T cycle= $1 / 500 * 10^6 = 2 * 10^{-9}$ secondes(0.25pt)

• $T_X = (5 * 1 * 2 * 10^{-9}) + (1 * 2 * 2 * 10^{-9}) + (1 * 3 * 2 * 10^{-9}) = 20. 10^{-9}$ secondes(0.50 pt)

• $T_Y = (10 * 1 * 2 * 10^{-9}) + (1 * 2 * 2 * 10^{-9}) + (1 * 3 * 2 * 10^{-9}) = 30. 10^{-9}$ secondes(0.50 pt)

C'est le mélange généré par le compilateur X qui va être exécuté plus rapidement. ... (0.25pt)

R3. La performance en MIPS (Millions d'Instructions Par Seconde) dans les deux cas.

Cas du compilateur X :

7.000.000.000 instructions → $20 * 10^{-9}$ secondes

Per_X → 1 seconde

⇒ Perf_X = $7 * 10^3 / 20 * 10^{-9} = 350 \text{ MIPS}$ (0.75 pt)

Cas du compilateur Y :

12.000.000.000 instructions → $30 * 10^{-9}$ secondes

Per_Y → 1 seconde

⇒ Perf_Y = $12 * 10^3 / 30 * 10^{-9} = 400 \text{ MIPS}$(0.75 pt)