



Examen – Architecture des ordinateurs

(Durée 1h30mn)

Le 13 /02 /2022

Remarques :

1. La partie QCM est à retourner avec la copie double sans oublier de mentionner le nom, le prénom, la section et le groupe.
2. L'utilisation de la calculatrice est interdite
3. La présentation de la copie est prise en considération.

Exercice 1 :

I/ Soit une mémoire possède **32 lignes d'adresses** et dont l'**accumulateur** égale à **16 bits**.

- 1- Calculer la **taille de cette mémoire** (en KOctets).
- 2- Est-ce que cette mémoire peut avoir **2^{30} cellules mémoire** ? justifier votre réponse.

II/ Soit une mémoire cache de niveau L1 ayant les caractéristiques suivantes :

- 64 mots par ligne (mots de 4 octets) - Taille de 256 Ko - L1 et L2 sont inclusifs - 8-associatifs - Remplacement LRU - Association par poids faible - Taille de bus d'adresse : 16 bits

- 1- Combien y-a-t-il de lignes dans cette mémoire cache ?
- 2- Combien y-a-t-il de blocs associatifs dans cette mémoire cache ?
- 3- Si la mémoire cache de niveau L2 a une taille de 2 Mo, combien y-a-t-il de blocs de la mémoire cache L2 par bloc de la mémoire cache L1 ?
- 4- Si la mémoire fait 1Go, combien d'adresses correspondront à un bloc du cache L1 ?

III/ A partir des performances du tableau ci-dessous calculer le temps d'exécution moyen d'une instruction pour chaque niveau sachant que durée d'un cycle horloge est **T**.

Niveaux	Temps d'accès succès (ns)	Taux de succès	Pénalité d'échec (Cycles)	Taille
Cache L1	3	80%	5	256 Ko
Cache L2	6	90%	10	2 Mo

Exercice 2 :

- 1- Ecrire un programme (en utilisant le minimum d'instruction possible) qui calcule l'expression **Z** dans une machine possédant **un opérande (machine à une (1) adresse)** et dont le mode d'adressage est **immédiat**.

$$Z = (A - B * C) + (D / E)$$

- 2- En supposant que l'architecture de la machine est à **deux adresses**, donner le code pour évaluer la même expression et ce en utilisant le minimum d'instruction possible.
- 3- Un processeur à 1000 MHz a été utilisé pour exécuter un programme de référence avec les instructions et le nombre de cycles d'horloge suivants :

Type d'instruction	Nombre d'instructions	Cycles d'horloge
Arithmétiques	50%	2
Point flottant	10%	4
Transfert de contrôle	10%	4
Transfert de données	30%	4

Calculer le **CPI** (Cycles Par Instruction) et le nombre d'instructions qu'il est capable de traiter par seconde (**MIPS**).

Exercice 3 :

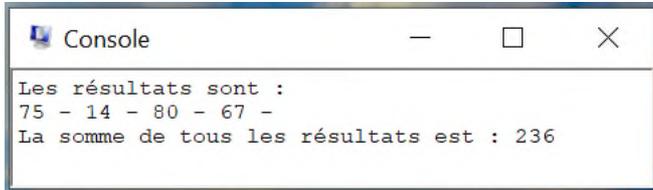
Compléter le code MIPS, ci-dessous, qui calcule et affiche les résultats de : $x+y+z - x+4 - x*8 - x+y+(z/2)$ tels que x, y, z sont des variables déclarées sur **8 bits** et leurs valeurs sont respectivement : **10, 50, 15**.

Pour l’affichage il faut avoir : (Voir l’image de la console ci-dessous)

Message1- **Les résultats sont** : retour à la ligne

Résultat des calculs : **résultat1 - résultat2 - résultat 3 - résultat 4 -**

Message 2 - **La somme de tous les résultats est** : *Résultat de la somme*



Code MIPS :

```
1  .data
2  x: .byte 10
3  y: .byte 50
4  z: .byte 15
5  [ ] 1
6  separateur:.asciiz "-"
7  retour:.asciiz "\n"
8  message2 : [ ] 2
9  .text
10 afficheSeparateur:
11 li $v0,1
12 syscall
13 li$v0,4
14 [ ] 3
15 syscall
16 jr $ra
17 main:
18 li$v0,4
19 la $a0,message1
20 syscall
21 lb $t1,x
22 lb $t2,y
23 [ ] 4
24 add $a0,$t1,$t2
25 add $a0,$a0,$t3
26 move $t0,$a0
27 jal afficheSeparateur
28 [ ] 5
29 move $t5,$a0
30 jal afficheSeparateur
31 li $t4,8
32 mul $a0,$t1,$t4
33 move $t6,$a0
34 jal afficheSeparateur
35 add $a0,$t1,$t2
36 li $t4,2
37 div $t4,$t3,$t4
38 add $a0,$a0,$t4
39 move $t7,$a0
40 [ ] 6
41 li$v0,4
42 la $a0,retour
43 syscall
44 li$v0,4
45 la $a0,message2
46 syscall
47 add $t4,$t0,$t5
48 add $a0,$t4,$t6
49 add $a0,$a0,$t7
50 li $v0,1
51 syscall
52 li $v0,10
53 syscall
```

1 →

2 →

3 →

4 →

5 →

6 →

Nom : Prénom : Sec/Gr :

Partie QCM : Choisir une ou plusieurs réponses pour chaque question :

1	Un programme source est généralement écrit en	<input type="checkbox"/> C++	<input type="checkbox"/> Basic	<input type="checkbox"/> Assembleur	<input type="checkbox"/> Langage machine
2	Ce sont les éléments de mémoire les plus rapides. Ils sont situés au niveau du processeur et servent au stockage des opérandes et des résultats intermédiaires.	<input type="checkbox"/> Mémoires caches	<input type="checkbox"/> Mémoires secondaires	<input type="checkbox"/> Registres	<input type="checkbox"/> BIOS
3	Le temps minimal entre 2 accès mémoire s'appelle	<input type="checkbox"/> Cycle mémoire	<input type="checkbox"/> Temps d'accès	<input type="checkbox"/> Temps d'exécution	<input type="checkbox"/> Temps optimal
4	C'est une mémoire morte qui peut être programmée une seule fois par l'utilisateur. La programmation est réalisée par un programmeur spécifique.	<input type="checkbox"/> RAM	<input type="checkbox"/> SDRAM	<input type="checkbox"/> PROM	<input type="checkbox"/> DDRAM
5	Un programme informatique qui convertit un programme en langage machine est appelé	<input type="checkbox"/> Interpréteur	<input type="checkbox"/> Compilateur	<input type="checkbox"/> Simulateur	<input type="checkbox"/> Convertisseur
6	Le mode d'adressage dans un processeur définit la manière	<input type="checkbox"/> d'exécution	<input type="checkbox"/> de lecture de l'opérande	<input type="checkbox"/> de calcul	<input type="checkbox"/> d'accès à l'opérande
7	Un aléa (problème) dans une architecture pipeline peut être	<input type="checkbox"/> structurel	<input type="checkbox"/> de données	<input type="checkbox"/> de panne	<input type="checkbox"/> de contrôle
8	Les instructions simples sont utilisées dans l'architecture	<input type="checkbox"/> Harvard	<input type="checkbox"/> RISC	<input type="checkbox"/> Von Newman	<input type="checkbox"/> CISC
9	la puissance de traitement du microprocesseur est représentée en	<input type="checkbox"/> PTM	<input type="checkbox"/> PUIS	<input type="checkbox"/> MIPS	<input type="checkbox"/> MIMS
10	La politique de remplacement lors d'un défaut de cache est	<input type="checkbox"/> LRU	<input type="checkbox"/> SIR	<input type="checkbox"/> ROM	<input type="checkbox"/> LFU



Cette partie est à retourner avec la copie double

Bon courage

Corrigé Examen – Architecture des ordinateurs (du 13 /02 / 2022)

Exercice 1 : (6.5 points)

I/ Soit une mémoire possèdent **32 lignes d'adresses** et dont l'**accumulateur** égale à **16 bits**.

1- Calculer la taille de cette mémoire (en KOctets). **(1 point)**

Capacité = $2^{\text{lignes d'@}} \times \text{nbre de lignes de données et accumulateur} = \text{nbre de lignes de données}$ (0.5 pt)

Capacité = $2^{32} \times 16 = 2^{36} \text{ bits} = 2^{36} / 2^{10} * 2^3 = 2^{23} \text{ KOctets}$ (0.5 pt)

2- Est-ce que cette mémoire peut avoir 2^{30} cellules mémoire ? justifier votre réponse. **(0.5 point)**

Oui c'est possible (0.25 pt) car $2^{\text{lignes d'@}} > \text{nbre de cellules}$ (0.25 pt)

II/ Soit une mémoire cache de niveau L1 ayant les caractéristiques suivantes :

- 64 mots par ligne (mots de 4 octets) - Taille de 256 Ko - L1 et L2 sont inclusifs - 8-associatifs - Remplacement LRU - Association par poids faible - Taille de bus d'adresse : 16 bits

1- Combien y-a-t-il de lignes dans cette mémoire cache ? **(1 point)**

Nombre de lignes = Taille cache / (Taille mot * nombre de mots par ligne) (0.5 pt)

Nombre de lignes (L1) = $256 \text{ KOct} / 64 * 4 \text{ Oct} = 2^8 * 2^{10} / 2^6 * 2^2 = 2^{10} = 1024 \text{ Lignes}$ (0.5 pt)

2- Combien y-a-t-il de blocs associatifs dans cette mémoire cache ? **(1 point)**

Nombre de blocs = Nombre de lignes / Nombre de lignes par bloc (0.5 pt)

= $2^{10} / 8 = 2^{10} / 2^3 = 2^7 = 128 \text{ blocs}$ (0.5 pt)

3- Si la mémoire cache de niveau L2 a une taille de 2 Mo, combien y-a-t-il de blocs de la mémoire cache L2 par bloc de la mémoire cache L1 ? **(0.5 point)**

Taille cache L2 / Taille cache L1 (0.25 pt) = $2 \text{ MOct} / 256 \text{ KOct} = 2^{21} / 2^{18} = 2^3 = 8$ (0.25 pt)

4- Si la mémoire fait 1Go, combien d'adresses correspondront à un bloc du cache L1 ? **(1 point)**

Taille mémoire / (Nombre de blocs * Taille mot * Taille lignes d'adresse) (0.5 pt)

= $1 \text{ GOct} / (128 * 4 \text{ Oct} * 16) = 2^{30} / 2^7 * 2^2 * 2^4 = 2^{17}$ (0.5 pt)

III/ A partir des performances du tableau ci-dessous calculer le temps d'exécution moyen d'une instruction pour chaque niveau sachant que durée d'un cycle horloge est **T**. **(1.5 point)**

Niveaux	Temps d'accès succès (ns)	Taux de succès	Pénalité d'échec (Cycles)	Taille
Cache L1	3	80%	5	256 Ko
Cache L2	6	90%	10	2 Mo

temps d'accès mémoire moyen = temps d'accès succès + (1 - taux de succès) x pénalité d'échec (0.5 pt)

temps d'accès mémoire moyen Cache L1 = $3 + (1-80\%) \times 5 = 4 \text{ T}$ (0.5 pt)

temps d'accès mémoire moyen Cache L2 = $6 + (1-90\%) \times 10 = 7 \text{ T}$ (0.5 pt)

Exercice 2 : (5.5 points)

1- Ecrire un programme (en utilisant le minimum d'instruction possible) qui calcule l'expression **Z** dans une machine possédant **un opérande (machine à une (1) adresse)** et dont le mode d'adressage est **immédiat**. **Z = (A-B*C) + (D/E)** **(2.5 points (10*0.25))**

Machine a 1@ Utilisation de l'accumulateur	
LOAD D;	=> [Acc] ← D
DIV E;	=> [Acc] ← D/E
STORE Z;	=> Z ← [Acc] = D/E
LOAD B;	=> [Acc] ← B
MPY C;	=> [Acc] ← B*C
STORE Y;	=> Y ← [Acc] = B*C
LOAD A;	=> [Acc] ← A
SUB Y;	=> [Acc] ← A- B*C
ADD Z;	=> [Acc] ← (A- B*C) + (D/E)
STORE Z;	=> Z ← [Acc] ← (A- B*C) + (D/E)

2- En supposant que l'architecture de la machine est à **deux adresses**, donner le code pour évaluer la même expression et ce en utilisant le minimum d'instruction possible. **(1.5 point)**

Machine a 2@ Avec un seul registre	Machine a 2@ Avec des registres
MOVE B, R1 MPY C, R1 SUB A, R1 MOVE E, R2 DIV D, R2 ADD R1, R2 STORE R2, Z	MOVE B, R1 MOVE C, R2 MPY R1, R2 MOVE A, R3 SUB R3, R2 MOVE D, R4 MOVE E, R5 DIV R4, R5 ADD R2, R5 STORE R5, Z

3- Un processeur à 1000 MHz a été utilisé pour exécuter un programme de référence avec les instructions et le nombre de cycles d'horloge suivants :

Type d'instruction	Nombre d'instructions	Cycles d'horloge
Arithmétiques	50%	2
Point flottant	10%	4
Transfert de contrôle	10%	4
Transfert de données	30%	4

Calculer le **CPI** (Cycles Par Instruction) et le nombre d'instructions qu'il est capable de traiter par seconde (**MIPS**). **(1.5 point)**

$$\text{CPI} = \sum_i (IC_i) (CC_i) / IC \text{ (0.5 pt)}$$

$$\text{CPI} = (50*2+10*4+10*4+30*4) / (50+10+10+30) = 3 \text{ (0.5 pt)}$$

MIPS (Millions d'Instructions Par Seconde) = fréquence du processeur / CPI.

$$\text{MIPS} = 1000/3 = 333.33 \text{ MIPS (0.5 pt)}$$

Exercice 3 : ((3 pts)

Code MIPS à compléter

```
1  .data
2  x: .byte 10
3  y: .byte 50
4  z: .byte 15
5  
6  separateur:.asciiz " - "
7  retour:.asciiz "\n"
8  message2 : 
9  .text
10 afficheSeparateur:
11 li $v0,1
12 syscall
13 li$v0,4
14 
15 syscall
16 jr $ra
17 main:
18 li$v0,4
19 la $a0,message1
20 syscall
21 lb $t1,x
22 lb $t2,y
23 
24 add $a0,$t1,$t2
25 add $a0,$a0,$t3
26 move $t0,$a0
27 jal afficheSeparateur
28 
29 move $t5,$a0
30 jal afficheSeparateur
31 li $t4,8
32 mul $a0,$t1,$t4
33 move $t6,$a0
34 jal afficheSeparateur
35 add $a0,$t1,$t2
36 li $t4,2
37 div $t4,$t3,$t4
38 add $a0,$a0,$t4
39 move $t7,$a0
40 
41 li$v0,4
42 la $a0,retour
43 syscall
44 li$v0,4
45 la $a0,message2
46 syscall
47 add $t4,$t0,$t5
48 add $a0,$t4,$t6
49 add $a0,$a0,$t7
50 li $v0,1
51 syscall
52 li $v0,10
53 syscall
```

- 1 → message1: .asciiz "Les résultats sont : \n" (0,5 pt)
- 2 → .asciiz "La somme de tous les résultats est : " (0,5 pt)
- 3 → la \$a0,separateur (0,5 pt)
- 4 → lb \$t3,z (0,5 pt)
- 5 → addi \$a0,\$t1,4 (0,5 pt)
- 6 → jal afficheSeparateur (0,5 pt)

Partie QCM : (5 pts) → 0,5 par bonne réponse

1	Un programme source est généralement écrit en	<input type="checkbox"/> C++	<input type="checkbox"/> Basic	<input checked="" type="checkbox"/> Assembleur	<input type="checkbox"/> Langage machine
2	Ce sont les éléments de mémoire les plus rapides. Ils sont situés au niveau du processeur et servent au stockage des opérandes et des résultats intermédiaires.	<input type="checkbox"/> Mémoires caches	<input type="checkbox"/> Mémoires secondaires	<input checked="" type="checkbox"/> Registres	<input type="checkbox"/> BIOS
3	Le temps minimal entre 2 accès mémoire s'appelle	<input checked="" type="checkbox"/> Cycle mémoire	<input type="checkbox"/> Temps d'accès	<input type="checkbox"/> Temps d'exécution	<input type="checkbox"/> Temps optimal
4	C'est une mémoire morte qui peut être programmée une seule fois par l'utilisateur. La programmation est réalisée par un programmeur spécifique.	<input type="checkbox"/> RAM	<input type="checkbox"/> SDRAM	<input checked="" type="checkbox"/> PROM	<input type="checkbox"/> DDRAM
5	Un programme informatique qui convertit un programme en langage machine est appelé	<input type="checkbox"/> Interpréteur	<input checked="" type="checkbox"/> Compilateur	<input type="checkbox"/> Simulateur	<input type="checkbox"/> Convertisseur
6	Le mode d'adressage dans un processeur définit la manière	<input type="checkbox"/> d'exécution	<input type="checkbox"/> de lecture de l'opérande	<input type="checkbox"/> de calcul	<input checked="" type="checkbox"/> d'accès à l'opérande
7	Un aléa (problème) dans une architecture pipeline peut être	<input checked="" type="checkbox"/> structurel	<input checked="" type="checkbox"/> de données	<input type="checkbox"/> de panne	<input checked="" type="checkbox"/> de contrôle
8	Les instructions simples sont utilisées dans l'architecture	<input type="checkbox"/> Harvard	<input checked="" type="checkbox"/> RISC	<input type="checkbox"/> Von Newman	<input type="checkbox"/> CISC
9	la puissance de traitement du microprocesseur est représentée en	<input type="checkbox"/> PTM	<input type="checkbox"/> PUIS	<input checked="" type="checkbox"/> MIPS	<input type="checkbox"/> MIMS
10	La politique de remplacement lors d'un défaut de cache est	<input checked="" type="checkbox"/> LRU	<input type="checkbox"/> SIR	<input type="checkbox"/> ROM	<input checked="" type="checkbox"/> LFU