Université Larbi ben M'hidi -Oum el Bouaghi-

Faculté des SESNV | Département : MI

Matière : Architecture des ordinateurs — *1h30m*

Niveau : (L2/S3) — *Le : 15/01/2024*

Examen de la session normale -> *Calculatrice non autorisée*

Questions de cours

1. Que signifie le concept : ***mémoire non volatile*** ? (**1 pt**)
2. C'est quoi l'architecture pipeline ? (**1 pt**)
   Quel est le gain en ***performance*** (en fonction de nombre d'instructions et de nombre d'étages) de cette architecture par rapport à l'architecture classique (séquentielle). (**1 pt**)
3. Expliquer les concepts : **Fréquence** et **CPI** et ses ***relations*** avec la performance du processeur. (**2 pts**)
4. Étant donné un bus d'adresse d'une fréquence de **`100`** Mhz qui possède un débit égal à **`400`** MO/s. Calculer le nombre de bits que ce bus peut transmettre simultanément. (**1 pt**)
5. Sachant qu'une instruction machine occupe **2** octets consécutifs et que la première instruction d'un programme donnée est chargée à l'adresse **`0x00A4`**, donner l'adresse de chargement des instructions n°**2**, **4** et **10** du même programme. (**1.5 pt**)
6. Coder l'instruction suivante : **`BLEZ $10, target`** (**3 pts**)
   (Avec ***`target`*** = **`0x0040003C`**, PC = **`0x00400020`** et **`$10 = -1`**).

Exercice

Un compilateur C a transformé un code en une séquence d'instructions en langage assemleur MIPS ci-contre (le programme P1) :

1. Extraire de ce programme les instructions qui réalisent une **lecture** de la mémoire ? supposons que le tableau se trouve à l'adresse (**`0x10010000`**) (**1 pt**)
2. Quelle est le mode d'adressage utilisé dans l'instruction : **`lw $t0, 0($t1)`**, expliquer que fait cette instruction. (**1.5 pt**)
3. Quelle est la valeur du contenu du registre **`$s4`** après exécution du programme ? (**2 pt**)
4. Au deuxième tour de l'exécution de la boucle, quelle est la valeur du registre **`$t1`** ? (**1 pt**)
5. Donner la valeur du regitre **`$t0`** à la fin de l'exécution du programme, expliquer. (**2 pt**)
6. Quel est le role de l'instruction **`add $t1, $s1, $s1`**, expliquer avec ***précision***. (**2 pt**)

Programme P1 :

```
.data
      tab: .word 10 20 30 40 50 60 70
.text
        addi $s1, $0, 0
        addi $s3, $0, 3
        la $s4, tab
Loop:   add $t1, $s1, $s1
        add $t1, $t1, $t1
        add $t1, $t1,$s4
        lw $t0, 0($t1)
        bgt $s1, $s3, Exit
        addi $s1, $s1, 2
        j Loop
Exit:   li $v0,10
        Syscall
```

بالتوفيق والسداد لكل طالب مجتهد

Université Larbi ben M'hidi -Oum el Bouaghi-

Faculté des SESNV | Département : MI

Matière : Architecture des ordinateurs — *1h30m*

Niveau : (L2/S3) — *Le : 15/01/2024*

Corrigé Type

1. Mémoire non volatile : une mémoire qui **NE PERD PAS** son contenu lorsque l'alimentation électrique est **INTERROMPUE**. **1 point**
2. Le gain en performance en fonction de nombre d'instructions et de nombre d'étages **1 point**

$$Gain = \frac{nbrI * nbrCycles}{nbrI + nbrCycles - 1}$$

Où :

- **nbrI** : le nombre d'instructions exécutées.
- **nbrCycles** : le nombre de cycles d'horloge par instruction (= le nbr d'étage sachant que chaque étage prend un cycle d'horloge)

L'architecture pipeline

- Le pipeline est un mécanisme permettant D'ACCROÎTRE LA VITESSE D'EXÉCUTION des instructions dans un micro-processeur.
- La technique du pipeline améliore LE DÉBIT DES INSTRUCTIONS
- La technique du pipeline exploite LE PARALLÉLISME entre instructions d'un flot séquentiel d'instructions.
- Permet à plusieurs instructions de SE CHEVAUCHER pendant l'exécution

1 point (deux parmi 4 sont suffisants pour la réponse, chacune sur 0.5 point)

3. **Fréquence** : la fréquence d'horloge représente le nombre de cycles par secondes, F= 1/P (P = temps de cycle). 0.25 point
   CPI : Cycle Per Instruction : le nbr moyen de cycles d'horloge nécessaires pour l'exécution d'une instruction 0.25 point

On a un Pgm = **nbrI** →

- Temps d'exécution = nbrCycle * P = nbrCycle * 1/F →0.5 points
- nbrCycle = nbrI * CPI →0.5 point
- Temps d'exécution = nbrI * CPI * P
- Performance = 1/Temps Exécution → 0.5 point

4. Débit maximale d'un bus : largeur * fréquence → 0.5 point
   → Largeur = Débit maximale / fréquence = 400 MO/s / 100 Mhz = 4 Octets = **32 bits** 0.5 point

5. Un programme commence à l'@ : 00A4h (sachant qu'une instruction occupe 2 Octets)
   On suppose que ***TMM = 1 Octet*** **(*ACCEPTABLE*)**

بالتوفيق والسداد لكل طالب مجتهد

Ins1	00A4 (l@ du 1er octet dans la 1ere instruction) 00A5(l@ du 2eme octet dans la 1er instruction)
Ins2	**00A6**, 00A7 0.5 point
Ins3	00A8, 00A9
Ins4	**00AA**, 00AB 0.5 point
Ins5	00AC, 00AD
…	…
Ins10	**00B6**, 0.5 point

Si **TMM = 2 Octets → l'@ pointe sur 2 octets (*ACCEPTABLE*)**

Ins1	00A4
Ins2	**00A5**, 0.5 point
Ins3	00A6
Ins4	**00A7** 0.5 point
…	…
Ins10	**00AD**, 0.5 point

6. Codage

`BLEZ $10, target` (*avec target* = **`0x0040003C`** et PC = **`0x00400020`**).

Format I :

Syntaxe : ***BLEZ $rs, etiquette***

Format I :	opcode	rs	rt	imm16

- COP BLEZ = **000110 (0.5 point)**
- Rs = $10 = **01010 (0.5 point)**
- Rt = **00000 (0.5 point)**

target = etiquette = (PC+4)+ Imm16 * 4 ***puisque $10 = -1 < 0*** **(0.5 point)**

→ imm16 = (target – (PC + 4)) / 4

→ imm16 = (**0x0040003C – (0x00400020 + 4)) / 4**
→ imm16 = (0x0040003C - 0x00400024)/ 4 → imm16 = (0x00000018 / 4) = 6 (0.5 point)

Codage est *ACCEPTABLE* soit en binaire soit en hexadécimal (0.5 point)

- ❒ Code **binaire** : **0001 1001 0100 0000 0000 0000 0000 0110**
- ❒ Code **hexadécimale : 0x19400006**

Université Larbi ben M'hidi -Oum el Bouaghi-

Faculté des SESNV | Département : MI

Matière : Architecture des ordinateurs — *1h30m*

Niveau : (L2/S3) — *Le : 15/01/2024*

Exercice

1. L'instruction qui réalise la lecture de la mémoire : **la $s4, tab**, **lw $t0, 0($t1)** **1 point**
2. Le mode d'adressage : **relatif** **0.5 point**
   Explication : **$t0 <- mémoire [0 + $t1]** **1 point**
3. Après exécution **: $s4 = 0x10010000** **2 points**
4. Au deuxième tour de la boucle la valeur de **$t1** = **0x10010008** **1 point**
5. A la fin de l'exécution la valeur de **$t0 = 50** **1 point**
   Le programme lit un élément et passe au suivant du suivant, les éléments lus : 10, 30 et 50 donc le dernier élément lu est **50**. **1 point**
6. Le role de l'instruction **add $t1, $s1, $s1** est de passer d'un élément du tableau à l'élément suivant en ajouter à chaque fois 4 (puisque les éléments du tableau sont de type Word = 32 bits = 4 octets. Équivalente à (PC = PC + 4) pour les instructions. **2 points**

بالتوفيق والسداد لكل طالب مجتهد