

Licence d'Informatique 2ème année

Architecture des ordinateurs Interrogation écrite - 12/11/2019 Corrigé

Exercice 1 (10 points)

Soit l'extrait de code en assembleur MIPS suivant :

```
L1: add $t0, $t1, $t2

L2: addi $t3, $t1, 4

L3: sub $t0, $t0, $t3

L4: beq $t0, $t4, L6

L5: addi $t3, $t3, 4

L6: sw $t3, 100($s1)
```

Pipeline

Questions:

1. Identifiez tous les aléas de données LAE, EAL, EAE et de contrôle. Donnez vos réponses sous la forme suivante :

Aléa de type XXX entre l'instruction Li et l'instruction Lj à cause de ...

Solutions (0,5 point par réponse juste):

Aléa de type LAE entre l'instruction L1 et l'instruction L3 à cause de l'écriture de t0 en t1 et sa lecture en t3

Aléa de type EAE entre l'instruction L1 et l'instruction L3 à cause de l'écriture de \$t0 en L1 et en L3

Aléa de type LAE entre l'instruction L2 et l'instruction L3 à cause de l'écriture de \$t3 en L2 et sa lecture en L3

Aléa de type LAE entre l'instruction L3 et l'instruction L4 à cause de l'écriture de \$t0 en L3 et sa lecture en L4

Aléa conditionnel de type LAE entre l'instruction L2 et l'instruction L5 à cause de l'écriture de \$t3 en L2 et sa lecture en L5

Aléa conditionnel de type EAL entre l'instruction L3 et l'instruction L5 à cause de la lecture de \$t3 en L3 et son écriture en L5

Aléa conditionnel de type EAE entre l'instruction L2 et l'instruction L5 à cause de l'écriture de \$t3 en L2 et en L5

Aléa conditionnel de type LAE entre l'instruction L5 et l'instruction L6 à cause de l'écriture de \$t3 en L5 et sa lecture en L6

Aléa conditionnel de type LAE entre l'instruction L2 et l'instruction L6 à cause de l'écriture de \$t3 en L2 et sa lecture en L6

Aléa de contrôle entre l'instruction L4 et l'instruction L5 à cause de l'instruction de branchement conditionnel en L5

2. Qu'y-a-t-il de particulier entre les instructions L2, L5 et L6 concernant leurs éventuels conflits de données?

Solution (0,5 point par réponse juste) : Les aléas entre L2 et L5 ont lieu seulement si le branchement de l'instruction L4 est non-pris. Il en est de même pour les aléas entre L5 et L6. Les aléas entre L2 et L6 ont lieu seulement si le branchement de l'instruction L4 est pris.

3. En utilisant un pipeline classique de 5 étages, donnez le tableau d'exécution depuis l'instruction L1 jusqu'à l'instruction L4 incluse (1 ligne par étage, 1 colonne par cycle), en insérant les bulles

nécessaires à la prise en compte des aléas de données. On considère que le pipeline ne dispose pas de forwarding du résultat, ni de forwarding sur chargement. Le tableau aura la forme suivante :

	1	2	3	4	5	6	7	8	9	10	11	12	
Fetch													
Decode													
Execute													
Memory													
Write													

Solution (3 points par réponse juste):

	1	2	3	4	5	6	7	8	9	10	11	12
Fetch	L1	L2	L3	L4								
Decode		L1	L2	X	X	L3	X	X	L4			
Execute			L1	L2	X	X	L3	X	X	L4		
Memory				L1	L2	X	X	L3	X	X	L4	
Write					L1	L2	X	X	L3	X	X	L4

Prédicteur de branchement

On utilise maintenant un prédicteur de branchement avec un compteur à saturation 2 bits pour prédire les actions de branchement de l'instruction L4 (**Pris** ou **Non Pris**). De plus, on considère 5 exécutions successives de l'extrait de code MIPS avec, pour chaque exécution, les valeurs de départ des registres \$t1, \$t2 et \$t4 qui sont données dans le tableau suivant :

Exec.	Valeurs de \$t1, \$t2, \$t4	Prédiction L4	Action L4	Nouvelle prédiction L4
1	1, 4, 0	NP (00)		
2	2, 6, 2			
3	3, 5, 1			
4	1, 4, 1			
5	0, 4, 0			

Le compteur à saturation 2 bits est initialisé à 00, ce qui veut dire que le prédicteur prédit que le branchement en L4 sera **Non Pris** (NP) lors de la première exécution.

Question:

5. En exécutant l'extrait de code 5 fois avec les valeurs de départ données des registres \$t1, \$t2 et \$t4, compléter les colonnes **Prédiction L4**, **Action L4** et **Nouvelle prédiction L4**, en indiquant dans les colonnes **Prédiction L4** et **Nouvelle prédiction L4**, la valeur du compteur 2 bits (00, 01, 10 ou 11) et la prédiction (P ou NP), et en indiquant dans la colonne **Action L4** si le branchement est effectivement Pris (P) ou Non Pris (NP). Combien y-a-t-il de bonnes prédictions?

Solution (2 points par réponse juste):

Exec.	Valeurs de \$t1, \$t2, \$t4	Prédiction L4	Action L4	Nouvelle prédiction L4
1	1, 4, 0	NP (00)	P	NP (01)
2	2, 6, 2	NP (01)	P	P (11)
3	3, 5, 1	P (11)	P	P (11)
4	1, 4, 1	P (11)	NP	P (10)
5	0, 4, 0	P (10)	P	P (11)

Exercice 2 (5 points)

Mémoires caches

On considère un cache associatif à 2 voies qui possède 8 lignes de cache et peut stocker 4 mots mémoire par ligne de cache.

Questions:

1. Combien y-a-t-il d'ensembles de lignes dans ce cache?

Solution (0,5 point par réponse juste):

2 voies \Leftrightarrow 2 lignes de cache par ensemble \Leftrightarrow 4 ensembles de lignes (pour 8 lignes de cache au total)

2. Si les adresses mémoire sont codées sur 8 bits, comment ces adresses sont-elles décomposées en *Tag, Index* et *Offset* afin de lire ou écrire la donnée correspondante dans le cache?

Solution (1 point par réponse juste):

- 4 mots par ligne de cache \Rightarrow 2 mots \Rightarrow 2 bits d'Offset
- -4 ensembles de lignes de cache $\Rightarrow 2^2$ ensembles $\Rightarrow 2$ bits d'Index
- 8 bits d'adresse \Rightarrow 8 2 2 = 4 bits de Tag
- 3. Soit la suite d'accès mémoire successifs suivante, dont les adresses mémoire sur 8 bits sont associées à un label A, B, C, D et E.

Label	Adresse
A	10100001
В	10100011
С	10101010
D	01011000
\mathbf{E}	11110011

Donnez les emplacements dans la mémoire cache de A, B, C, D et E après l'exécution de ces 5 accès mémoire, et en considérant que la mémoire cache est vide au départ. Utilisez un tableau pour représenter ces emplacements, en y indiquant les tags, indexes et offsets.

Solution (3,5 points par réponse juste):

index	Tag	11	10	01	00
00	1010	В		A	
00	1111	Е			
01					
01					
10	1010		С		
10	0101				D
11					
11					

Exercice 3 (5 points)

Architectures Parallèles

On considère un programme pour lequel 5% de son temps d'exécution séquentiel est pris par la partie qui ne peut pas être parallélisée.

Questions:

1. En appliquant le principe de la loi d'Amdahl, quelle est l'accélération maximum que l'on peut espérer atteindre pour ce programme en le parallélisant sur un nombre illimité de cœurs de processeurs?

3

Solution (2 points par réponse juste) :
$$Acceleration = \frac{1}{-P+1} = 1/0,05 = 20$$

2. En appliquant toujours le principe de la loi d'Amdahl, , quelle est l'accélération que l'on peut obtenir sur 20 cœurs de processeurs? De quel type est cette accélération?

Solution (2 points par réponse juste) : $Acceleration = \frac{1}{\frac{P}{N} - P + 1} = \frac{1}{\frac{0.95}{20} - 0.95 + 1} = 10,26$

3. Quelle sera alors l'efficacité obtenue sur 20 cœurs de processeurs ?

Solution (1 point par réponse juste) : 10,26/20 = 51,3%