

GIF-17455 – Architecture des microprocesseurs

Examen Partiel

Mardi le 25 octobre 2005

Durée: 13h30-15h20

Vous devez montrer vos calculs ou justifier vos réponses. Une réponse, même bonne, mais sans autre information, ne vaut aucun point.

Question 1

a) Dans ce code, identifiez tous les aléas LAE, EAE, et EAL (RAW, WAW, WAR).

L.D.	F0, 0(R1)	% MEM[R1] → F0
MULT.D	F2, F0, F6	% F0 * F6 → F2
L.D.	F6, 8(R1)	% MEM[8+R1] → F6
MULT.D	F6, F6, F2	% F6 * F2 → F6
S.D.	F6, 16(R1)	% F6 → MEM[16 + R1]
DADDI	R1, R1, #-32	% R1 - 32 → R1

b) Parmi les aléas identifiés en a), identifiez ceux qui ***pourraient*** causer une suspension de pipeline sur un processeur avec ordonnancement statique (pipeline statique comme vu à l'annexe A).

Note : Votre réponse doit être indépendante de la réalisation (nombre d'étages du pipeline, envoi de données ou non, délai d'accès mémoire, etc.) du processeur.

c) Pour un processeur avec ordonnancement dynamique (dynamic scheduling), les obstacles de type WAW et WAR ne peuvent être ignorés. Pourquoi ?

Question 2

- a) Quelle est la différence entre un prédicteur de branchement statique et un prédicteur de branchement dynamique ?
- b) Quelle est la différence entre les prédicteurs de branchement à corrélation (1,1) et (2,1) ?
- c) Pourquoi le tampon de destination de branchement (branch target buffer) a-t-il été introduit ?
- d) Pouvez-vous décrire une modification du tampon de destination de branchement qui permet de « gagner » un cycle lors d'un branchement correctement prédit comme suivi ?

Pour les questions e) et f) :

Le code suivant est exécuté sur un processeur avec prédicteur de branchement dynamique :

```
for (i = 0 ; i < 100 ; i++){
    for (j = 0 ; j < 10 ; j++){
        ...
        B1: retour à la boucle « for » de j
    }
    ...
    B2: retour à la boucle « for » de i
}
```

B1 et B2 sont respectivement les instructions de branchement des deux boucles « for ».

- e) B1 a sa propre entrée dans le tampon de prédiction de branchement (branch prediction buffer) Combien de fois B1 sera-t-il bien prédit si
 - 1) on utilise un prédicteur de 1 bit initialisé à l'état « suivi » ?
 - 2) on utilise un prédicteur de 2 bits initialisé à l'état « 10 » (i.e. suivi) ?
- f) B1 et B2 partagent la même entrée dans le tampon de prédiction de branchement. Combien de fois B1 sera-t-il bien prédit si on utilise un prédicteur de 1 bit initialisé à l'état « suivi » ?

Question 3 Les questions a) b) et c) sont indépendantes

a) Votre entreprise développe un appareil électronique pour une **application très spécialisée et unique**. Vous avez comparé la performance de nombreux processeurs en effectuant vos propres tests, tests ciblés pour votre application, et vous avez retenu le processeur X.

À la dernière minute, votre superviseur vous demande de considérer le nouveau processeur de la compagnie « P.A. Semi » qui, selon des tests effectués sur des benchmarks standards et connus, performe mieux que le processeur X. Comme le processeur est nouveau, vous ne pouvez pas en obtenir un pour effectuer vos propres tests de performance. Vous devez donc vous en remettre aux résultats des tests publiés par « P.A. Semi ». Quelle mise en garde devrez-vous faire à votre superviseur lorsque vous lui présenterez les résultats de votre comparaison entre le processeur X et ce nouveau processeur ?

b) Votre produit a été développé et vous devez réduire le temps d'exécution de 10%. L'analyse de l'exécution de votre application vous indique que de nombreuses instructions causent des suspensions de pipeline, suspensions qui pourraient être éliminées ou réduites en utilisant un meilleur compilateur (le compilateur déplace des instructions pour éviter les aléas).

Si vous voulez atteindre votre objectif, quel est le % du temps d'exécution que les instructions à déplacer doivent prendre (avant le changement de compilateur) ? Faites l'hypothèse que le CPI des instructions déplacées sera réduit de 40%.

c) Vous constatez que l'unité fonctionnelle de calcul en point flottant, qui n'est pas implantée en pipeline, est responsable de plusieurs suspensions de pipeline. Vous contactez le fabricant du processeur qui vous propose deux options :

1) ajouter une 2^{ème} unité fonctionnelle identique à la première

2) réaliser l'unité fonctionnelle de calcul en point flottant en pipeline. La durée du cycle augmentera cependant de 2%.

Le CPI de l'application (avant modification) est de 1.9, 10% des instructions sont des instructions à point flottant, et il faut 4 cycles pour en exécuter une. Vous observez les distributions suivantes pour les instructions point flottantes (PF) :

30% des instructions sont en groupe de trois : instr. entières, PF, PF, PF, instr. entières

20% des instructions sont en groupe de deux : instr entières, PF, PF, instr. entières

50% des instructions sont isolées : instr entières, PF, instr. entières

Quelle option offerte par le fabricant devez-vous choisir ?

Question 4

Un processeur avec ordonnancement dynamique a les caractéristiques suivantes :

- Les instructions de lecture ou écriture en mémoire utilisent *l'unique* UAL (ALU) entière pour le calcul des adresses. Le calcul d'une adresse prend 1 cycle d'exécution.
- Les accès à la mémoire (écriture et lecture) prennent un cycle.
- Il n'y a qu'un port d'accès à la mémoire.
- Il y a autant de stations de réservation que nécessaire, ainsi que de tampons d'écriture et de lecture.
- Quand une instruction passe à l'étape WB au cycle x , une instruction qui attendait pour la même unité fonctionnelle (à cause d'un obstacle structurel) peut démarrer son exécution au cycle x .
- Une seule instruction peut diffuser (écrire) une valeur sur le bus commun des données (CDB) pendant un cycle. Les instructions de branchement et d'écriture en mémoire n'utilisent pas le CDB.
- Quand il y a un conflit entre instructions pour une ressource (incluant le CDB), l'instruction qui précède les autres a la priorité.
- Une instruction qui attend une donnée sur le CDB peut démarrer son exécution au cycle *suivant* la diffusion (broadcast) de la valeur sur le CDB.
- Il y a *une unité fonctionnelle de multiplication en point flottant*, et elle n'est pas implantées en pipeline. La multiplication prend 5 cycles.
- L'instruction qui suit le branchement est lancée au cycle suivant le lancement du branchement. Le processeur n'est pas spéculatif.
- Pour les instructions de branchement, l'évaluation de la condition et le calcul de l'adresse prennent un cycle au total dans une unité d'exécution dédiée aux instructions de branchement.

Voici le code d'une boucle qui est exécutée à 10 reprises par un processeur avec ordonnancement dynamique avec les caractéristiques suivantes :

Boucle :	L.D.	F0, 0(R1)	% MEM[R1] \rightarrow F0
	MULT.D	F2, F0, F6	% F0 * F6 \rightarrow F2
	L.D.	F6, 8(R1)	% MEM[8+R1] \rightarrow F6
	MULT.D.	F6, F6, F2	% F6 * F2 \rightarrow F6
	S.D.	F6, 16(R1)	% F6 \rightarrow MEM[16 + R1]
	DADDI	R1, R1, #-32	% R1 - 32 \rightarrow R1
	BNEZ	R1, boucle	% Si R1 \neq 0, retourne à « Boucle »

NOM : _____

Matricule _____

Voici le tableau d'exécution des premières instructions. Ce tableau est complété jusqu'à la fin du cinquième cycle

	Lancement (IS)	Exécution (EX)	Accès mémoire (MEM)	Diffusion sur le bus (WB)	Justification de délai
L.D. F0, 0(R1)	1	2	3	4	
MULT.D F2,F0,F6	2	5 -			
L.D. F6, 8(R1)	3	4	5		
MULT.D F6,F6,F2	4				
S.D. F6, 16(R1)	5				
DADDI R1,R1, #-32					
BNEZ R1, boucle					
L.D. F0, 0(R1)					

a) Donnez le contenu des stations de réservations et le statut des registres à la fin du cinquième cycle (utilisez la feuille ci-jointe pour votre solution)

b) Complétez le tableau ci haut et justifiez tous les délais (répondez sur cette feuille)

c) Avec un processeur spéculatif, est-ce que chaque itération serait exécutée plus rapidement ? Est-ce que le temps total d'exécution des dix itérations serait plus court?

NOM : _____

Matricule _____

Nom station	Instruction	Vj	Vk	Qj	Qk	Commentaires
Station 1						
Station 2						
Station 3						
Station 4						
Station 5						
Station 6						

Statut des registres							
	F0	F2	F4	F6	F8	R1	R3
Qi							