

GIF-17455 – Architecture des microprocesseurs :
Solutions de l'examen partiel #1

Lundi le 16 octobre 2006

Durée: 9h30-11h20

Question 1 (35 pts)

- a) Faux. La performance (temps d'exécution) dépend aussi du CPI et # instructions.
- b) Vrai – il faut que ces instructions prennent plus de 62.5% du temps d'exécution (avant amélioration)
- c) Vrai. La latence des instructions (en nombre de cycles) augmente avec le nombre d'étages, ce qui augmente le nombre de cycles de suspension causés par les aléas de donnés.
- d) Faux. Il y a aussi un accès à la mémoire pour chaque instruction (pour aller la lire).
- e) Faux. Comme les exceptions peuvent survenir à différents étages du pipeline, il est possible qu'une instruction génère une exception avant une instruction qui a été lancée plus tôt.
- f) Vrai. Comme le nombre de cycles d'exécution des instructions n'est pas constant, il est possible qu'une instruction se termine et écrive dans un registre avant une autre instruction lancée plus tôt et qui a le même registre de destination.
- g) Faux. La réalisation en pipeline permet d'éliminer les aléas de type structurel sur cette unité, mais pas les aléas de données (il faut quand même attendre la fin de l'exécution pour obtenir la valeur).

Question 2 (20 pts)

Avant l'ajout des instructions de délai :

- 0 cycles de suspension pour les branchements non pris
- 2 cycles de suspension pour chaque branchement pris
- Contribution au CPI : $0.65 \times 2 = 1.3$

Après l'ajout des instructions de délai :

- Quand l'instruction de délai est utilisée (A% du temps)
 - 0 cycles de suspension pour les branchements non pris
 - 1 cycle de suspension pour les branchements pris
- Quand l'instruction de délai n'est pas utilisée
 - 1 cycle de suspension pour les branchements non pris
 - 2 cycles de suspension pour les branchements pris
- Contribution au CPI : $A \times 0.65 \times 1 + (1-A) \times 0.35 \times 1 + (1-A) \times 0.65 \times 2$

Il faut donc que $A \times 0.65 \times 1 + (1-A) \times 0.35 \times 1 + (1-A) \times 0.65 \times 2 \leq 1.3$ ce qui donne $A \geq 35\%$

Question 3 (25 pts)**Séquence 1**

Cycles	1	2	3	4	5	6
LD...	IF	ID	EX	MEM	WB	
ADD....		IF	ID	Susp.	EX	MEM

Au cycle 3, test si (IF/ID.IR[rs ou rt] == ID/EX.IR[rd]) et (IF/ID.IR[opcode] == Op. ALU) et (ID/EX.IR[opcode] == LD); dans ce cas, le test est positif et on a suspension du pipeline pour l'instruction ADD... et les suivantes.

Au cycle 5, test si (ID/EX.IR[rs ou rt] == MEM/WB.IR[rd]) et les deux opcodes; dans ce cas, le test est positif et le registre MEM/WB.IR[LMD] est envoyé à l'entrée de l'ALU pour l'opération.

Séquence 2

Cycles	1	2	3	4	5	6
LD...	IF	ID	EX	MEM	WB	
SD....		IF	ID	EX	MEM	

Au cycle 5, test si (EX/MEM.IR[rs] == MEM/WB.IR[rd]) et (EX/MEM.IR[opcode] == Op. SD) et (MEM/WB.IR[opcode] == LD); dans ce cas, le test est positif et le contenu du registre MEM/WB.IR[LMD] est rangé (écrit) en mémoire.

Question 4 (20 pts)

Avant le changement :

- Les seules suspensions sont causées par les instructions arithmétiques qui suivent un LD fournissant un des opérandes.
 - Contribution au CPI : $70\% \text{ des LD} \times 30\% \text{ instructions} \times 1 = 0.21$
- Total : $1 \times (1 + 0.21) = 1.21$

Après le changement :

- Avec l'inversion des étages EX et MEM, le cas précédent ne génère plus de suspensions.
- Pour chaque LD & SD avec offset, il y a maintenant une instruction de plus (ces instructions ne causent pas d'aléas – donnée du problème)
 - $45\% \times 40\% = 18\%$ d'instructions supplémentaires sans aléas
- Avec l'inversion, le calcul de l'adresse des LD et SD avec offset se fait au 4^{ème} étage (EX), et le LD ou SD qui suit doit être suspendu 1 cycle pour attendre l'adresse
 - Contribution au CPI des LD & SD avec offset : $45\% \times 40\% = 0.18$
- Total : $0.18 \text{ instructions} \times 1 + 1 \times (1 + 0.18) = 1.36$