

NOM: _____

Code permanent: _____

**Architecture des ordinateurs contemporains
Examen Final (Hiver 2003)**

Professeur: El Mostapha Aboulhamid

Directives

- Toute documentation est permise.
 - Une calculatrice (même programmable) est permise.
 - Le barême est indiqué entre parenthèses.
 - Inscrire votre nom et code permanent sur la première page du questionnaire.
 - Si vous ne comprenez pas exactement une question, faites une interprétation, expliquez la et continuez. Veillez à la clarté de vos réponses et justifiez les.
 - *Les questions sont indépendantes et leur ordre n'indique pas leur degré de difficulté.*
-

NOM: _____**Code permanent:** _____**Question 1 (20)**

La loi d'Amdahl donne l'accélération d'une application typique étant donnée une amélioration partielle effectuée sur une architecture. L'objectif est de récrire cette loi pour le cas suivant :

Supposer que si cette tâche est effectuée sur un monoprocesseur, elle prendrait un temps T . Supposer avoir une fonction de la forme $f(i, p)$ qui donne la fraction du temps T où exactement i processeurs, parmi un total de p processeurs disponibles, pourraient être utilisés pour effectuer la tâche. Ainsi nous avons l'équation (1) suivante :

$$\sum_{i=1}^p f(i, p) = 1 \quad (1)$$

1. Donner une expression qui donne l'accélération générale d'une telle application en fonction de la distribution f .
2. Quelle serait cette accélération si : $f(i, p) = \frac{1}{p} \quad \forall i \in [1, p]$.

NOM: _____ **Code permanent:** _____

Question 2 (20)

Considérer l'exécution de la boucle suivante sur un processeur à émission unique d'instructions ("single issue processor"), d'abord avec l'ordonnancement dynamique ensuite avec la spéculation:

```

Loop:    LD      R2, 0(R1)
         DADDI  R2, R2, #1
         SD     R2, 0(R1)
         DADDI  R1, R1, #4
         BNEZ  R2, LOOP
    
```

Supposer qu'il y a différentes unités de traitement d'entiers pour le calcul de l'adresse effective, pour les opérations UAL, ainsi que pour l'évaluation des conditions de branchement. Créer une table similaire à la figure 3.27 du livre (3^{ème} édition, et reproduite ci-après) pour les trois premières itérations de la boucle, et ce pour chacun des deux cas cités précédemment (dynamique puis spéculation). On suppose qu'au plus deux instructions peuvent effectuer un "commit" par coup d'horloge.

Iteration number	Instructions	Issues at	Executes	Memory access at	Write CDB at	Comment
1	L.D F0,0(R1)	1	2	3	4	First issue
1	ADD.D F4,F0,F2	1	5		8	Wait for L.D
1	S.D F4,0(R1)	2	3	9		Wait for ADD.D
1	DADDIU R1,R1,#-8	2	3		4	Executes earlier
1	BNE R1,R2,Loop	3	5			Wait for DADDIU
2	L.D F0,0(R1)	4	6	7	8	Wait for BNE complete
2	ADD.D F4,F0,F2	4	9		12	Wait for L.D
2	S.D F4,0(R1)	5	7	13		Wait for ADD.D
2	DADDIU R1,R1,#-8	5	6		7	Executes earlier
2	BNE R1,R2,Loop	6	8			Wait for DADDIU
3	L.D F0,0(R1)	7	9	10	11	Wait for BNE complete
3	ADD.D F4,F0,F2	7	12		15	Wait for L.D
3	S.D F4,0(R1)	8	10	16		Wait for ADD.D
3	DADDIU R1,R1,#-8	8	9		10	Executes earlier
3	BNE R1,R2,Loop	9	11			Wait for DADDIU

Figure 3.27 The clock cycle of issue, execution, and writing result for a dual-issue version of our Tomasulo pipeline with separate functional units for integer ALU operations and effective address calculation, which also uses a wider CDB. The extra integer ALU allows the DADDIU to execute earlier, in turn allowing the BNE to execute earlier, and, thereby, starting the next iteration earlier.

NOM: _____ **Code permanent:** _____

NOM: _____

Code permanent: _____

NOM: _____**Code permanent:** _____**Question 3 (20)**

- a. Implémenter la barrière de la figure 6.39 du livre (reproduite ci-après) en utilisant les loquets à file (queuing locks).

```
local_sense =! local_sense; /* toggle local_sense */
lock (counterlock); /* ensure update atomic */
count=count+1; /* count arrivals */
if (count==total) { /* all arrived */
    count=0; /* reset counter */
    release=local_sense; /* release processes */
}
unlock (counterlock); /* unlock */
spin (release==local_sense); /* wait for signal */
}
```

- b. Comparer la performance par rapport au « spin lock barrier ».

NOM: _____

Code permanent: _____

NOM: _____**Code permanent:** _____**Question 4 (20)**

a) Supposons qu'un multiprocesseur ayant deux processeurs possède une mémoire partagée centrale accédée via un bus partagé, que la mémoire est initialisée à 0, et que le bus est géré par un protocole espion (snoopy algorithm).

Décrivez dans le tableau ci-joint (Table 1) la séquence des requêtes et des réponses, et les valeurs transmises via les contrôleurs de cache et du bus. Indiquer aussi les valeurs stockées en mémoire centrale. Les adresses A1 et A2 réfèrent au même bloc de la mémoire cache mais des blocs différents de la mémoire.

P1 lit A1

P2 lit A2

P1 écrit A1

P2 écrit A2

P1 lit A1

b) Supposons que multiprocesseur possède plutôt une mémoire partagée accédée via un bus ou une autre réseau d'interconnexion, on utilise alors un protocole de cohérence de cache basé sur des répertoires. Remplir alors la Table 2 pour indiquer les différentes activités engendrées par les lectures/écritures de la question a).

	P1			P2			Bus				Répertoire			Mémoire
OPÉRATIONS	Etat	Adr	Val.	Etat	Adr	Val.	Action	Proc	Adr	Val	Adr	Etat	{Procs}	Val.

Table 2: Opérations utilisant un répertoire

Question 5 (20)

Un système contient un processeur qui roule 1 GHz et qui a un CPI de 0.80 excluant les accès mémoire. Les seules instructions qui lisent ou écrivent en mémoire sont des « load » (15% de toutes les instructions) et des « store » (5% de toutes les instructions). On suppose que toutes les instructions ainsi que les registres sont sur 32 bits.

Le système mémoire est composé d'une cache à deux niveaux. Le premier niveau (L1) est séparé (split) et n'a aucune pénalité en cas de succès (hit). La cache d'instructions (I-cache) ainsi que la cache des données (D-cache) sont toutes deux à correspondance directe (direct map) et sont chacune de 32 KB. La I-cache possède des blocs de 32 octets et un taux d'échec (miss rate) de 2%. La D-cache, qui est « write through », possède des blocs de 16 octets et un taux d'échec de 5%. Il y a un tampon d'écriture pour la D-cache qui permet d'éliminer les suspensions (stalls) sur 95% des écritures.

Le niveau 2 de la cache (L2) est une cache unifiée de 512KB et qui est « write back », avec un temps d'accès de 15 ns. Chaque bloc de L2 est de 64 octets. Elle est connectée à L1 par un bus qui roule à 266 MHz et qui peut transférer 128 bits par cycle du bus. Parmi toutes les références envoyées à L2, 80% sont satisfaites sans accéder à la mémoire principale. 50% des blocs remplacés sont « dirty ».

La mémoire principale a une largeur de bande de 128 bits et un temps de latence de 60 ns, après quoi une séquence quelconque de mots de 128 bits peut être transférée à raison d'un mot de 128 bits par cycle sur un bus qui roule à 133 MHz.

1. Quel est le temps moyen d'accès mémoire pour la recherche des instructions ?
2. Quel est le temps moyen d'accès mémoire pour la lecture des données ?
3. Quel est le temps moyen d'accès mémoire pour l'écriture des données ?
4. Quel est le CPI global incluant l'accès mémoire ?

