

NOM: _____

Code permanent: _____

**IFT3380/6380 : Architecture des ordinateurs contemporains
Examen final (Hiver 2005)**

Professeur : El Mostapha Aboulhamid

Directives

- Toute documentation est permise.
 - Une calculatrice (même programmable) est permise.
 - Le barème 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.*
-

Question 1 (25)

Nous disposons d'un système monoprocesseur, dont l'accès à la mémoire principale est de 200 nanosecondes. On a constaté qu'une instruction de chargement à la mémoire (« load ») prend un temps d'exécution variable. Les mesures suivantes ont été prises, elles sont données en ordre croissant.

Pour chacun de ces temps d'exécution, donner l'explication **la plus plausible** qui justifierait ces durées. On suppose que tous les caches ont une politique de réécriture simultanée (« write through »).

a) 10 nanosecondes

b) 50 nanosecondes

c) 200 nanosecondes

d) 400 nanosecondes

e) 20 millisecondes (20×10^6 nanosecondes).

f) Donner 3 détails sur l'organisation et l'implémentation de ce monoprocesseur que vous pourriez déduire des mesures précédentes.

Question 2 (25)

Soit l'énoncé de branchement suivant:

```
if (x>0) call PRINT;
```

Supposons que x prenne les valeurs suivantes de manière consécutive : -1, -3, 2, 3, -5, 6, 6, 7, -1, 0, 1. Nous ne connaissons pas la valeur de x dans l'étape de recherche de l'instruction donc nous ne pourrons pas savoir de manière exacte le résultat du branchement à cette étape. Chaque fois qu'on rencontre un branchement pour la première fois on suppose qu'il n'a pas été pris, on utilisera par la suite une table de prédiction de branchement.

- Remplir la prédiction plus bas dans les 2 cas suivants :
 - a) on utilise une table (BHT) de 1 bit (se servant du résultat précédent du branchement comme indication du branchement présent)
 - b) On utilise une table 2-bit BHT

Données	-1	-3	2	3	-5	6	6	7	-1	0	1
Résultat du test											
Prédiction (1 bit)											
Prédiction (2 bits)											

- Combien de fois le schéma 1-bit prédit-il bien ? _____
- Même question avec 2-bit _____
- Est-ce qu'un prédicteur avec corrélation donnerait un meilleur résultat dans ce cas particulier?

Question 3 (25)

a) Un multiprocesseur ayant deux processeurs possède une mémoire partagée centrale. L'accès à cette mémoire s'effectue à travers un bus partagé. La mémoire est initialisée à 0, et 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 à travers les contrôleurs de cache et du bus. Indiquer aussi les valeurs stockées en mémoire centrale. Les adresses A1 et A2 correspondent au même bloc des mémoires caches, mais des blocs différents de la mémoire.

P1 écrit 11 dans A1

P1écrit 21 dans A2

P2 lit A2

P1 lit A1

P1 écrit 31 dans A1

P2 écrit 41 dans A2

P1 lit A1

b) Supposons que le multiprocesseur possède plutôt une mémoire partagée accessible à travers un 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)**.

Opération	P1			P2			Bus				Mém	
	Etat	Adr	Val	Etat	Adr	Val	Act.	CPU	adr	Val	Adr	Val
bus-espion												

Table 1: Opérations utilisant un bus espion (snoopy bus)

Question 4 (25)

On suppose une machine DLX super-scalaire où on peut émettre deux opérations indépendantes dans le même cycle, y compris deux opérations arithmétique simples.

Instruction productrice	Instruction consommatrice	Latence
Op UAL Virgule flottante	Op UAL V.F.	2
Op UAL V.F.	Stockage double en mémoire	2
Chargement double de la mémoire	Op UAL V.F.	1
Chargement double	Stockage double	2

Soit le fragment de code suivant:

```
BOUCLE: LD      F0, 0(R1)
         MULTD  F0, F0, F2
         LD      F4, 0(R2)
         ADDD   F0, F0, F4
         SD     0(R2), F0
         SUBI   R2, R2, 8
         SUBI   R1, R1, 8
         BNEZ  R1, BOUCLE
```

Dérouler la boucle 4 fois (4 copies de la boucle) et l'ordonnancer sur le DLX super-scalaire de manière optimale (en minimisant les calages). Pour chaque opération indiquer le cycle où elle commence ainsi que le cycle où une autre instruction qui en dépend peut commencer.

On suppose que les unités en virgule flottante sont des unités pipelines, qu'il y a une unité addition en virgule flottante et une unité de multiplication, et qu'il y a deux unités arithmétiques simples.

Donner le temps que prend chaque itération avec votre nouvel ordonnancement.

