

**NOM:** \_\_\_\_\_**Code permanent:** \_\_\_\_\_

**Architecture des ordinateurs contemporains  
Examen Final (Hiver 2001)**

**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 le 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)**

On suppose une machine DLX où le branchement est retardé avec un créneau d'une instruction on suppose aussi que la latence entre une instruction productrice et une consommatrice sont comme suit

Instruction productrice	Instruction consommatrice	Latence
Op UAL Virgule flottante	Op UAL V.F.	2
Op UAL V.F.	Stockage double en mémoire	1
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
        NOP
```

1. Récrire le code en explicitant les suspensions qui existent dans le code.
2. Combien de fois faut-il dérouler la boucle afin de permettre un ordonnancement qui élimine toutes les suspensions (nombres de copies d'itérations)? On suppose que les tests intermédiaires sur R1 sont ignorés.
3. Dérouler la boucle autant de fois que nécessaire afin de l'ordonnancer sans aucune suspension. Les tests intermédiaires sur R1 doivent être ignorés.

**NOM:** \_\_\_\_\_ **Code permanent:** \_\_\_\_\_

**NOM:** \_\_\_\_\_**Code permanent:** \_\_\_\_\_**Question 2 (20)**

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

Dérouler la boucle de la question 1, autant de fois que nécessaire, et l'ordonner sur le DLX super-scalaire de manière optimale (en minimisant les suspensions). 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.

**NOM:** \_\_\_\_\_ **Code permanent:** \_\_\_\_\_

**NOM:** \_\_\_\_\_ **Code permanent:** \_\_\_\_\_

**Question 3 (20)**

Soit le fragment de code suivant (On suppose qu’au départ R3 et R1 contiennent zéro):

```

L0:      BNEZ   R1, L1      -- branchement bI si R1 ≠ 0
        ADDI   R3, R3, #1
        SUBI   R4, R3, #2
        BNEZ   R4, L0      -- branchement bII si R3 ≠ 2
        ADD    R1, R0, R3   -- mettre R3 dans R1
        J      L0          -- aller à L0
L1:      SUB    R1, R1, R1
        SUBI   R3, R3, R3
        J      L0
    
```

On suppose qu’on a un bit de prédiction, remplir une table similaire à la table 4.17 du livre, où chaque ligne représente les valeurs de R1, R3 et R4 ainsi que les prédictions et les actions prises à l’exécution de chacun des branchements bI ou bII.

R1	R3	R4	Prédiction bI	Action bI	Prédiction bII	Action bII
0	0		NT	NT		
0	1	-1			NT	T

Remplir la même table avec un prédicteur (1,1), au départ tous les prédicteurs sont à NT.

R1	R3	R4	Prédiction bI	Action bI	Prédiction bII	Action bII
0	0		NT/NT			
0	1	-1				

**NOM:** \_\_\_\_\_**Code permanent:** \_\_\_\_\_**Question 4 (20)**

Un système a un espace d'adressage sur 32 bits. Il possède au premier niveau, une cache d'instructions et une cache de donnée séparées. Il possède aussi un deuxième niveau de cache, cette deuxième cache est unifiée. Le premier niveau de la cache de données utilise la correspondance directe, il a une politique "write-through" et "write-allocate", et possède un tampon d'écriture parfait (ne provoque jamais de suspension). Cette première cache a une taille de 32K octets et possède des blocs de 8 octets. La cache d'instructions a une taille de 8K octets et possède des blocs de 8 octets aussi. La cache du deuxième niveau est associative avec des ensembles de taille 2, elle utilise une politique de "write-back" et "write-allocate". Elle est de 512 K octets et ses blocs ont 32 octets chacun.

1. Pour chacune des caches donner le nombre de bits du champ de l'adresse qui sert d'index ainsi que celui qui sert d'étiquette (tag) et enfin celui qui sert de déplacement à l'intérieur du même bloc.

La cache d'instructions a un taux d'échec (miss rate) de 1%, La cache de données a un taux d'échec de 25%. La cache unifiée du deuxième niveau a un taux d'échec de 10%. On suppose que 35% des instructions accèdent la mémoire: 70% sont des lectures et 30% des écritures en mémoire. On suppose qu'en tout temps 30% des blocs dans la cache du deuxième niveau sont modifiés (dirty). Il faut 10 cycles pour accéder à un bloc au deuxième niveau quand il est manquant au premier niveau de la cache. Le temps d'accès à la mémoire principale est de 100 cycles.

2. Quel est le pourcentage d'accès à la mémoire qui provoque une recherche en mémoire principale (donner l'équation puis le résultat numérique).

**NOM:** \_\_\_\_\_

**Code permanent:** \_\_\_\_\_

3. Combien de cycles au maximum peut-on dépenser pour accéder à une donnée ?  
Expliquer quand est-ce que cela se produit.

4. Quel est le temps moyen d'accès à la mémoire ?



**NOM:** \_\_\_\_\_ **Code permanent:** \_\_\_\_\_

**Question 5 (20)**

Soit un processeur qui permet un ordonnancement dynamique des instructions à l'aide de l'algorithme de Tomasulo. L'exécution des accès-mémoires prend 2 cycles, elle est de 3 cycles pour l'addition et la soustraction, elle est de 8 pour la multiplication et la division.

1. Modifier la figure ci-dessous (qui représente le cycle 6 de la machine) pour indiquer l'état de la machine au cycle 9.

**Statut des instructions V.F.**

	j	k	Emission	Exécution complétée	Ecriture résultat		Occ.	Adr.	
LD	F6	34	(R2)	1	3	4	ld1	Non	
LD	F2	45	(R3)	2	4	5	ld2	Non	
MULD	F0	F2	F4	3			ld3	Non	
SUBD	F8	F6	F2	4					
DIVD	F10	F0	F6	5					
ADDD	F6	F8	F2	6					
SUBD	F12	F10	F0						

**Stations de réservation**

Temps restant	Nom	Occupé	Op	S1: Vj	S2: Vk	RS pour Qj	RS pour Qk
1 ?	Add1	Oui	SUBD	M(34+R2)	M(45+R3)		
	Add2	Oui	ADDD		M(45+R3)	Add1	
	Add3	Non					
4 ?	Mult1	Oui	MULTD	M(45+R3)	R(F4)		
	Mult2	Non?	?		M(34+R2)	Mult1	

Corriger si nécessaire là où il y a des points d'interrogation

**Statut de registres**

Horloge	Unité Fonct.	F0	F2	F4	F6	F8	F10	F12	...
6		Mult1	M(45+R3)		Add2	Add1	Mult2		

2. Remplir complètement le tableau suivant en donnant le temps de fin d'exécution et d'écriture des résultats de chaque instruction.

	j	k	Emission	Exécution complétée	Ecriture résultat	
LD	F6	34	(R2)	1	3	4
LD	F2	45	(R3)	2	4	5
MULD	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		
SUBD	F12	F10	F0			

**NOM:** \_\_\_\_\_

**Code permanent:** \_\_\_\_\_