

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

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

## Architecture des ordinateurs contemporains Examen Intra (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.

Question 1 (25)	
Question 2 (30)	
Question 3 (20)	
Question 4 (25)	

**Question 1 (25)**

Soit une machine pipeline, ayant 7 étages (IF1, IF2, ID, EX, M1, M2, WB). Les branchements inconditionnels sont résolus à la fin de l'étage 3 alors que les branchements conditionnels ne le sont qu'à la fin de l'étage 4. On suppose la distribution suivantes: 20% des instructions sont des branchements conditionnels (dont 60% sont pris) et 5% sont des sauts inconditionnels ou des appels de sous-programmes.

1. Combien de suspensions (cycles perdus) on a
  - a. pour un branchement conditionnel? \_\_\_\_\_
  - b. pour un branchement inconditionnel? \_\_\_\_\_
2. Calculer le CPI (cycles par instruction) de cette machine, si on ne fait ni prédiction ni remplissage du créneau d'attente.
3. On suppose que les branchements, qu'ils soient conditionnels ou non, sont différés avec un créneau d'attente. On suppose que ce créneau est rempli correctement pour 30% des branchements. Calculer le nouveau CPI ?
4. Dans le cas des branchements conditionnels on suppose qu'en plus du créneau d'attente, on prédit que le branchement ne sera pas pris. Quel est le nouveau CPI?

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

**Question 2 (30)**

On considère une architecture DLX où le branchement est résolu à l'étage du décodage et où on a:

- un pipeline F-D-X-M-W (Fetch, Decode, Execute, Memory, Write Back) pour le calcul entier,
- un pipeline F-D-Y<sup>3</sup>-M-W pour la multiplication en virgule flottante
- et un pipeline F-D-A<sup>2</sup>-M-W pour l'addition en virgule flottante.
- Les étages F,D, M,W sont communs au trois pipelines alors qu'il y a un étage X, 3 étages Y et 2 étages A distincts.

Soit le code suivant:

```

ADDI    R2,R0,#1600 -- mettre 1600 dans R2
BOUCLE: LD      F0, 0(R2)
        LD      F4, 8(R2)
        MULTD   F0,F0,F4
        SD      F0, 8(R2) -- écrire F0 en mémoire à 8(R2)
        ADDD    F2,F0,F2
        SD      F2, 0(R2)
        SUBI    R2,R2,#16
        BNEZ    R2, BOUCLE
    
```

1. Compléter les indications d'aléas potentiels RAW sur le code précédent (en tirant un trait entre l'origine de l'aléa et sa destination).
2. Compléter la table ci-joint afin de donner le timing de chaque instruction, et en tenant compte des aléas, il faudra utiliser les courts-circuits nécessaires afin de minimiser les suspensions. Notez qu'il y a une seule instruction à la fois qui peut récrire les registres dans l'étage WB. S'il y a deux candidates, la deuxième dans le pipeline est suspendu après son décodage.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
LD F0, 0(R2)	F	D	X	M	W																		
LD F4, 8(R2)																							
MULTD F0,F0,F4																							
SD F0, 8(R2)																							
ADDD F2,F0,F2																							
SD F2, 0(R2)																							
SUBI R2,R2,#16																							
BNEZ R2, BOUCLE																							

Quel est le temps d'exécution de ce programme en nombre de cycles?



**NOM:** \_\_\_\_\_**Code permanent:** \_\_\_\_\_**Question 3 (20)**

On propose trois améliorations d'une nouvelle architecture résultant en une accélération respective de  $A_1 = 40$ ,  $A_2 = 30$  et  $A_3 = 20$ . Seulement une des trois améliorations est utilisable à un moment donné: on ne peut pas utiliser deux améliorations simultanément.

1. Calculer l'accélération globale obtenue si on utilise seulement l'amélioration 1, cette amélioration est utilisable 20% du temps.
2. Une généralisation de la formule d'Amdahl pour tenir compte de plusieurs améliorations comme spécifié au début de la question est:

$$A_G = \left[ \left( 1 - \sum_i F_i \right) + \sum_i \frac{F_i}{A_i} \right]^{-1}$$

Les  $A_i$  désignent les accélérations et les  $F_i$  désignent la fraction de temps où cette amélioration est utilisable.  $A_G$  désigne l'amélioration globale du système.

Si les améliorations 1 et 2 sont utilisables dans 35% et 25% des cas, quelle est la fraction  $F_3$  où l'amélioration 3 doit être utilisable afin d'atteindre une accélération globale de 10?

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

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

**NOM:** \_\_\_\_\_**Code permanent:** \_\_\_\_\_**Question 4 (25)**

## 1. Ecrire un modèle VHDL:

Ecrire en VHDL une entité avec son architecture. L'architecture est composée de 2 processus ou plus (si nécessaire).

Le premier processus appelé DEMANDEUR génère un entier aléatoire entre 0 et 999. Il attend une réponse possible du 2<sup>ème</sup> processeur appelé FOURNISSEUR, cette réponse peut être ATTENDRE ou CONTINUER. Dans le cas d'attente, le processus DEMANDEUR attend 1000 ns avant de formuler une nouvelle demande. Dans le cas de continuation, le demandeur procède après une attente de 10 ns seulement.

Chaque fois qu'il reçoit un nouveau nombre N, le processus FOURNISSEUR calcule les 2 quantités suivantes:

$$Q := N \text{ div } 10$$

$$R := N \text{ mod } 10$$

Il dispose d'un tableau C[0..9] qui a les valeurs initiales suivantes: 0, 11, 22, 33, 44, 55, 66, 77, 88, 99.

Il envoie la réponse CONTINUER si  $Q=C[R]$  sinon, il renvoie ATTENDRE et effectue  $C[R]:=Q$ .

## 2. Discussion:

Quand vous écrivez votre code VHDL, supposez que les nombres générés par le processus DEMANDEUR sont tous différents, vous n'avez pas besoin de coder le test pour ne pas trop alourdir le code.

Discuter l'impact d'avoir deux nombres aléatoires identiques qui se succèdent. Discuter toute autre hypothèse que vous avez prise pour simplifier votre code, si besoin est.

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

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