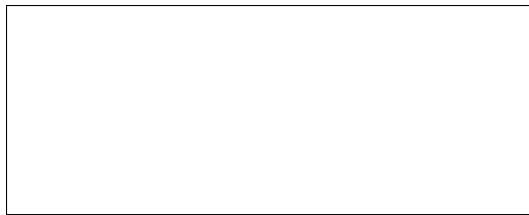


EXAMEN PARTIEL #1

Instructions : – Une feuille aide-mémoire recto verso manuscrite est permise ;
– Utilisez le verso des pages pour les brouillons ;
– Durée de l'examen : 1 h 50.

Pondération : Cet examen compte pour 25% de la note finale.



Signature : _____

Question 1 (14 points sur 100)

Soit deux systèmes suivants, avec les performances associées en calculs et en accès mémoire.

Performance	Calculs	Accès mémoire
Système A	7628	2953
Système B	11 210	3501

Les valeurs des deux mesures peuvent être directement comparées et les valeurs élevées sont meilleures que les valeurs faibles. On suppose que pour une application particulière, un processeur passe 25 % du temps pour effectuer des accès mémoire et le temps restant à faire des calculs.

- (4) (a) Calculez le gain offert par l'utilisation de système B relativement au système A.

Solution:

Performance pondérée du système A : $\text{Perf}_A = 0,75 \times 7628 + 0,25 \times 2953 = 6459,25$.

Performance pondérée du système B : $\text{Perf}_B = 0,75 \times 11\,210 + 0,25 \times 3501 = 9282,75$.

Gain système B relativement au système A : $\text{gain}_B = \frac{\text{Perf}_B}{\text{Perf}_A} = \frac{9282,75}{6459,25} = 1,4371$.

- (4) (b) Supposons maintenant qu'il existe une version du système A à deux cœurs de sorte que la performance en calculs de votre application peut être parallélisée à 50 %. Les performances des accès de mémoire restent inchangées. Calculez le gain en performance offert par ce système à deux cœurs relativement à la version à un cœur.

Solution:

$$\text{Loi d'Amdahl : gain} = \frac{1}{\frac{\text{frac}_{\text{seq}}}{\text{gain}_{\text{par}}} + \frac{\text{frac}_{\text{par}}}{\text{gain}_{\text{par}}}} = \frac{1}{(1 - (0,5 \times 0,75)) + \frac{0,5 \times 0,75}{2}} = 1,23.$$

- (6) (c) Déterminez le niveau de parallélisme nécessaire au niveau des calculs de l'application pour faire en sorte que le système A à deux cœurs ait le même niveau de performance que le système B.

Solution:

On cherche à obtenir un gain de 1,4371 relativement au système A à un cœur (réponse du point a). On trouve le gain avec la loi d'Amdahl :

$$\text{gain} = \frac{1}{(0,25 + 0,75(1 - x)) + \frac{0,75x}{2}} = 1,4371,$$

où x est le niveau de parallélisme. En brassant l'équation on obtient :

$$x = -\frac{8}{3} \left(\frac{1}{1,4371} - 1 \right) = 0,811.$$

Donc les calculs devraient être parallèles à 81,1 %.

Question 2 (20 points sur 100)

Supposons que vous faites partie d'une organisation fabriquant des panneaux d'affichage variable formés d'une matrice de 8192 diodes électroluminescentes (DEL) (64 lignes par 128 colonnes). Chaque DEL est une composante discrète et indépendante des autres, ayant un temps moyen avant défaillance (MTTF) de 10^8 heures d'activités. On suppose que les DEL formant un panneau d'affichage sont en activités pendant 2000 heures en moyenne par année.

- (10) (a) Calculez le temps moyen avant défaillance d'une DEL parmi toutes celles formant un panneau d'affichage.

Solution:

Comme toutes les composantes sont indépendantes, on doit d'abord calculer la probabilité que le panneau tombe en panne durant une heure.

$$\begin{aligned} 1 - p_{\text{panne}} &= \left(1 - \frac{1}{10^8}\right)^{8192} \\ p_{\text{panne}} &= 1 - (1 - 10^{-8})^{8192} = 8,1917 \times 10^{-5} \end{aligned}$$

Le taux de panne est l'inverse de cette probabilité :

$$\text{taux}_{\text{panne}} = \frac{1}{p_{\text{panne}}} = \frac{1}{8,1917 \times 10^{-5}} = 1,2208 \times 10^4.$$

Le temps moyen avant la défaillance d'une DEL d'un panneau est donc de 12 208 heures.

- (10) (b) Supposons qu'une garantie est offerte sur une période d'un an à savoir qu'un panneau d'affichage est remplacé si deux DEL ou plus sont défectueuses. Déterminez le taux de retour espéré pour cette garantie.

Solution:

Il faut d'abord déterminer la probabilité d'une panne d'une DEL durant une année :

$$p_{\text{DEL}} = \frac{1}{10^8} \times 2000 = 2 \times 10^{-5}.$$

On calcule la probabilité de ne pas avoir de panne durant l'année :

$$p_{0\text{panne}} = (1 - p_{\text{DEL}})^{8192} = 0,8489.$$

On calcule ensuite la probabilité qu'une DEL tombe en panne durant l'année, ce qui correspond à la probabilité d'une panne parmi les 8192 DEL multipliée par la probabilité que les 8191 DEL restantes ne tombent pas en panne :

$$p_{1\text{panne}} = \left(1 - (1 - p_{\text{DEL}})^{8192}\right) \times (1 - p_{\text{DEL}})^{8191} = 0,1283$$

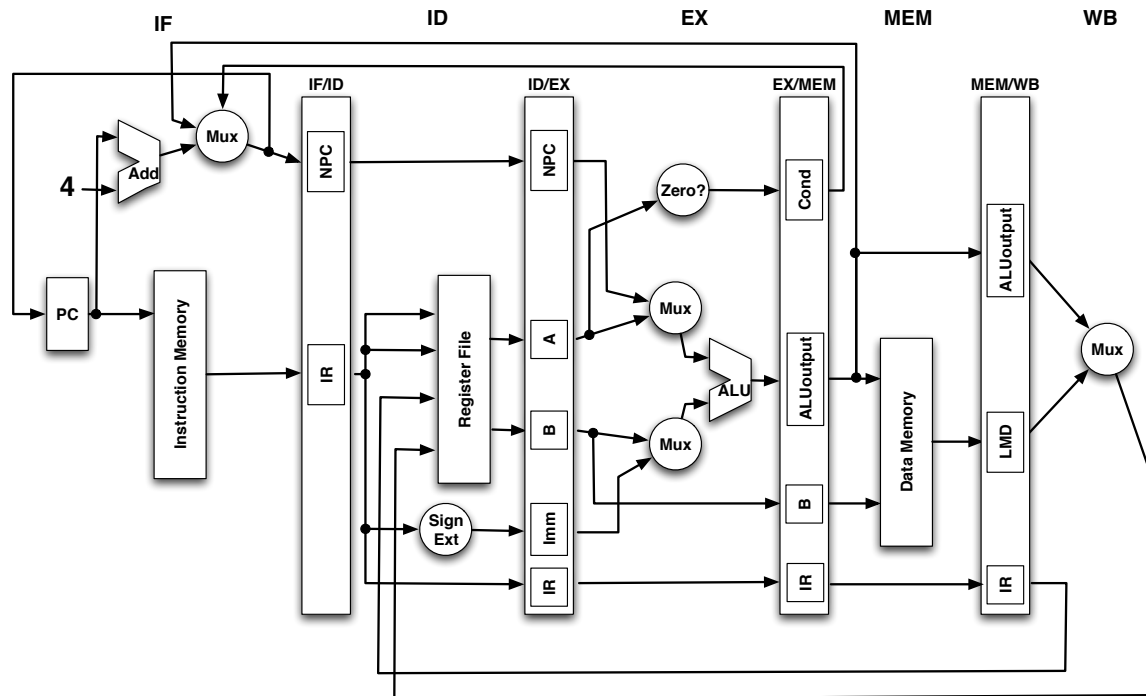
Le taux de retour correspond à la probabilité que plus qu'une DEL tombe en panne durant l'année :

$$p_{\text{retour}} = 1 - (p_{0\text{panne}} + p_{1\text{panne}}) = 0,0228.$$

Il devrait donc y avoir un taux de retour de 2,28 %.

Question 3 (20 points sur 100)

Soit la figure suivante, qui présente la micro architecture du MIPS (32 bits) implantée en pipeline, **sans envoi** de données, tel que vu en classe.



Supposons que ce processeur en pipeline exécute les instructions suivantes.

			t	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$t+7$	$t+8$
i	ADDU	R3, R3, R4	IF	ID	EX	MEM	WB				
$i+1$	ADDIU	R2, R2, #-1		IF	ID	EX	MEM	WB			
$i+2$	ADDU	R4, R5, R4			IF	ID	EX	MEM	WB		
$i+3$	SW	R3, 8 (R1)				IF	ID	ID	EX	MEM	WB
$i+4$	SW	R4, 16 (R1)					IF	bulle	ID	ID	EX
$i+5$	BEQZ	R2, Loop							IF	bulle	ID

La bulle au cycle $t+5$ est causée par une dépendance de données entre les instructions i et $i+3$ via le registre R3, alors que la bulle au cycle $t+7$ est causée par une dépendance de données entre les instructions $i+2$ et $i+4$ via le registre R4.

- (4) (a) Donnez la valeur du registre ID/EX.B au début du cycle $t+2$.

Solution:

La valeur du registre ID/EX.B au cycle $t+2$ correspond à la valeur du deuxième opérande de l'instruction ADDU R3, R3, R4, soit R4.

- (4) (b) Donnez la valeur du registre IF/ID.IR au début du cycle $t+7$.

Solution:

La valeur du registre IF/ID.IR au cycle $t+7$ correspond au code machine de l'instruction $i+5$, soit BEQZ R2, Loop.

- (4) (c) Donnez la valeur du registre EX/MEM.ALUoutput au début du cycle $t+4$.

Solution:

Le registre EX/MEM.ALUoutput au cycle $t+4$ contient le calcul l'addition de l'instruction $i+1$, soit R2-1.

- (4) (d) Donnez la valeur du registre IF/ID.NPC au début du cycle $t + 7$.

Solution:

Le registre IF/ID.NPC au cycle $t + 7$ contient l'adresse de l'instruction immédiatement consécutive à l'instruction $i + 5$ dans le programme (et non l'adresse du branchement).

- (4) (e) Indiquez s'il est possible d'améliorer la performance de ce code dans le pipeline, en réduisant le nombre de cycles nécessaires à son exécution dans le pipeline, seulement en inversant l'ordre d'exécution des instructions, en prenant soin de respecter la rectitude du programme. Si c'est le cas, indiquez les modifications à effectuer à l'ordre d'exécution des instructions et le gain résultant.

Solution:

Il est possible de réduire d'un cycle d'horloge l'exécution du programme en inversant les instructions $i + 1$ et $i + 2$. Cette modification permet d'éliminer la bulle au cycle $t + 7$.

Question 4 (14 points sur 100)

Soit le code MIPS suivant.

```
1  Loop:  LD      R2, 512(R1)
2          LD      R3, 1024(R1)
3          DMUL    R3, R2, R3
4          SD      R3, 2048(R1)
5          DADDI   R1, R1, #-4
6          BNE     R1, R4, Loop
```

Donnez l'exécution dans un pipeline MIPS pour deux itérations de la boucle, jusqu'au cycle d'horloge $t + 17$, dans le tableau de la page suivante. Expliquez également la cause (l'aléa à la source) de chaque bulle insérée dans le pipeline.

Supposez que le pipeline supporte un **envoi de données complet** entre les étages du pipeline. Supposez également que vous utilisez une version optimisée du pipeline MIPS, avec condition et adresse de branchement connues à la fin de l'étage ID. Finalement, supposez que la condition de l'instruction de branchement à la ligne 6 (BNE R1, R4, Loop) est vraie, soit que R1 n'est pas égal à R4.

Solution:

			t	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$
i	LD	R2, 512 (R1)	IF	ID	EX	MEM	WB	
$i+1$	LD	R3, 1024 (R1)		IF	ID	EX	MEM	WB
$i+2$	DMUL	R3, R2, R3			IF	ID	bulle	EX
$i+3$	SD	R3, 2048 (R1)				IF	bulle	ID
$i+4$	DADDI	R1, R1, #-4						IF
$i+5$	BNE	R1, R4, Loop						
$i+6$	LD	R2, 512 (R1)						
$i+7$	LD	R3, 1024 (R1)						
$i+8$	DMUL	R3, R2, R3						
$i+9$	SD	R3, 2048 (R1)						
$i+10$	DADDI	R1, R1, #-4						
$i+11$	BNE	R1, R4, Loop						
			$t+6$	$t+7$	$t+8$	$t+9$	$t+10$	$t+11$
i	LD	R2, 512 (R1)						
$i+1$	LD	R3, 1024 (R1)						
$i+2$	DMUL	R3, R2, R3	MEM	WB				
$i+3$	SD	R3, 2048 (R1)	EX	MEM	WB			
$i+4$	DADDI	R1, R1, #-4	ID	EX	MEM	WB		
$i+5$	BNE	R1, R4, Loop	IF	bulle	ID	EX	MEM	WB
$i+6$	LD	R2, 512 (R1)			bulle	IF	ID	EX
$i+7$	LD	R3, 1024 (R1)					IF	ID
$i+8$	DMUL	R3, R2, R3						IF
$i+9$	SD	R3, 2048 (R1)						
$i+10$	DADDI	R1, R1, #-4						
$i+11$	BNE	R1, R4, Loop						
			$t+12$	$t+13$	$t+14$	$t+15$	$t+16$	$t+17$
i	LD	R2, 512 (R1)						
$i+1$	LD	R3, 1024 (R1)						
$i+2$	DMUL	R3, R2, R3						
$i+3$	SD	R3, 2048 (R1)						
$i+4$	DADDI	R1, R1, #-4						
$i+5$	BNE	R1, R4, Loop						
$i+6$	LD	R2, 512 (R1)	MEM	WB				
$i+7$	LD	R3, 1024 (R1)	EX	MEM	WB			
$i+8$	DMUL	R3, R2, R3	ID	bulle	EX	MEM	WB	
$i+9$	SD	R3, 2048 (R1)	IF	bulle	ID	EX	MEM	WB
$i+10$	DADDI	R1, R1, #-4			IF	ID	EX	MEM
$i+11$	BNE	R1, R4, Loop				IF	bulle	ID

Explications des bulles introduites dans le pipeline :

- Les bulles au cycle $t+4$ et au cycle $t+13$ sont causées par un aléa de données de type RAW via R3 entre les instructions $i+1$ et $i+2$ au cycle $t+4$ et les instructions $i+7$ et $i+8$ au cycle $t+13$, la lecture de la valeur en mémoire n'étant disponible qu'à la fin de l'étage MEM.
- Les bulles aux cycles $t+7$ et $t+16$ sont causées par un aléa de données de type RAW via R1 entre les instructions $i+5$ et $i+6$ (cycle $t+7$) et les instructions $i+10$ et $i+11$ (cycle $t+16$), la valeur étant nécessaire à l'étage ID par le branchement conditionnel, étant donné l'optimisation du pipeline pour les branchements.
- La bulle au cycle $t+8$ est causée par un aléa de contrôle dû au branchement conditionnel de l'instruction $i+5$, la direction et l'adresse de branchement étant connues à la fin de l'étage ID.

Question 5 (32 points sur 100)

Répondez aussi brièvement et clairement que possible aux questions suivantes.

- (4) (a) Deux facteurs principaux influençant le coût de composantes électroniques ont été présentés en classe. Donnez ces deux facteurs ainsi qu'une brève explication de chacun d'eux.

Solution:

Le temps (courbe d'apprentissage) est un facteur influençant le coût des composantes électroniques. Avec le temps, les procédés de fabrication sont améliorés, de sorte que les coûts devraient baisser. Également, avec le temps il y a un ajustement de l'offre et la demande, en plus d'une augmentation de la compétition, ce qui a généralement un effet à la baisse sur le coût.

L'augmentation du volume est le deuxième facteur majeur influençant le coût des composantes électroniques. Avec un plus grand volume, la courbe d'apprentissage est passée plus rapidement. Également, avec plus de volume, la capacité d'achat est meilleure, alors que les coûts fixes, incluant les coûts de développement, sont amortis sur une plus grande quantité de composantes.

- (4) (b) Indiquez pourquoi une suite de benchmarks est de loin préférable à l'utilisation d'un seul benchmark pour l'évaluation des performances de systèmes informatiques.

Solution:

Une suite de benchmark est préférable, car elle permet de capturer un ensemble de cas d'utilisation variés, alors qu'un benchmark représente généralement un cas particulier. De plus, il est dangereux de concevoir un système informatique en utilisant un seul benchmark pour en évaluer les performances, car le concepteur peut alors spécialiser son architecture sur le benchmark, consciemment ou inconsciemment, pour avoir de bonnes performances avec ce cas spécifique, mais sans nécessairement bien performer dans les autres situations. Cette surspécialisation est beaucoup plus difficile, voire impossible, lorsqu'on utilise une suite de benchmarks représentant les usages typiques et variés du système.

- (4) (c) Expliquez clairement en quoi consistent la bande passante et la latence.

Solution:

La bande passante est une mesure de débit correspondant à la quantité de travail effectué ou à la quantité d'information transmise.

La latence est une mesure de délai entre le début et la fin d'un événement et correspond au temps nécessaire pour accéder à une donnée ou le temps nécessaire pour transmettre une information.

- (4) (d) Expliquez pourquoi les architectures mémoire-mémoire sont maintenant désuètes.

Solution:

Les architectures mémoire-mémoire offrent des instructions-machine où plusieurs opérandes correspondent à des accès mémoire. Ces instructions étaient pratiques pour les programmeurs lorsque les applications étaient faites en langage assembleur. Cependant, elle implique une complexité accrue en terme d'implantation matérielle, car une instruction peut correspondre à plusieurs accès mémoire suivis d'une opération arithmétique. Cette complexification rend difficile l'utilisation des avancées architecturales développées depuis les années 1980, telles que les pipelines.

- (4) (e) Pourquoi dit-on qu'un pipeline ne réduit pas la latence d'exécution dans un microprocesseur ?

Solution:

La latence d'exécution dans un microprocesseur correspond au nombre de cycles d'horloge nécessaire pour compléter l'exécution d'une instruction. En ce sens, un pipeline ne réduit pas la latence d'exécution, car une instruction prend autant de temps à s'exécuter dans un microprocesseur pipeliné que dans un microprocesseur équivalent non pipeliné.

- (4) (f) Expliquez en quoi consiste un aléa de contrôle lors de l'exécution d'un programme.

Solution:

Un aléa de contrôle correspond au fait que la direction d'un branchement conditionnel n'est pas encore connue, de sorte que l'on ne peut s'engager dans l'exécution des instructions suivantes.

- (4) (g) Expliquez en quoi consiste le maintien d'exceptions précises.

Solution:

Le maintien d'exceptions précises consiste à assurer que lorsque qu'une exception survient dans un microprocesseur, toutes les instructions précédents l'instruction ayant causé l'exception sont complétées, alors qu'aucune des instructions suivant celle qui a causé l'exception sont complétées ou ont modifié l'état du programme.

- (4) (h) Des architectures RISC ont été proposées au moment où les langages de programmation compilés de plus haut niveau, tels que le langage C, ont gagné en popularité. Expliquez la relation entre ces deux technologies et la synergie entre celles-ci qui leur a permis de s'imposer dans les systèmes informatiques.

Solution:

Les architectures RISC sont conçues pour des systèmes où le code machine est généralement produit par des compilateurs, à partir de programmes faits en des langages de haut niveau. Ceci est en opposition avec les architectures CISC, qui visent souvent à offrir un environnement de programmation en assembleur plus convivial pour les programmeurs, au risque de complexifier grandement l'architecture matérielle. L'arrivée des architectures RISC a permis des améliorations rapides et soutenues des systèmes informatiques, rendant encore plus attrayante la programmation d'applications complexes en langage de plus haut niveau, applications qui auraient été autrement très complexes à développer en assembleur. Un autre élément à considérer est la disponibilité de systèmes d'exploitation programmés en langage C, qui ont permis de proposer des architectures novatrices pour lesquelles l'effort nécessaire pour fournir une infrastructure logicielle de soutien est raisonnable.

Résultats

Question:	1	2	3	4	5	Total
Points:	14	20	20	14	32	100
Score:						