

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 (15 points sur 100)

Soit le code MIPS suivant.

```
1  Loop:   LD      R2, 0(R1)
2          LD      R4, 0(R3)
3          DADDU   R6, R2, R4
4          SD      R6, 0(R5)
5          DADDIU  R1, R1, #8
6          DADDIU  R3, R3, #8
7          DADDIU  R5, R5, #8
8          BNE     R5, R7, Loop
```

Donnez l'exécution dans un pipeline MIPS 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 8 (`BNE R5, R7, Loop`) est vraie, soit que R5 n'est pas égal à R7.

Solution:

			t	$t + 1$	$t + 2$	$t + 3$	$t + 4$	$t + 5$
i	LD	R2, 0 (R1)	IF	ID	EX	MEM	WB	
$i + 1$	LD	R4, 0 (R3)		IF	ID	EX	MEM	WB
$i + 2$	DADDU	R6, R2, R4			IF	ID	bulle	EX
$i + 3$	SD	R6, 0 (R5)				IF	bulle	ID
$i + 4$	DADDIU	R1, R1, #8						IF
$i + 5$	DADDIU	R3, R3, #8						
$i + 6$	DADDIU	R5, R5, #8						
$i + 7$	BNE	R5, R7, Loop						
$i + 8$	LD	R2, 0 (R1)						
$i + 9$	LD	R4, 0 (R3)						
$i + 10$	DADDU	R6, R2, R4						
$i + 11$	SD	R6, 0 (R5)						
$i + 12$	DADDIU	R1, R1, #8						
$i + 13$	DADDIU	R3, R3, #8						
			$t + 6$	$t + 7$	$t + 8$	$t + 9$	$t + 10$	$t + 11$
i	LD	R2, 0 (R1)						
$i + 1$	LD	R4, 0 (R3)						
$i + 2$	DADDU	R6, R2, R4	MEM	WB				
$i + 3$	SD	R6, 0 (R5)	EX	MEM	WB			
$i + 4$	DADDIU	R1, R1, #8	ID	EX	MEM	WB		
$i + 5$	DADDIU	R3, R3, #8	IF	ID	EX	MEM	WB	
$i + 6$	DADDIU	R5, R5, #8		IF	ID	EX	MEM	WB
$i + 7$	BNE	R5, R7, Loop			IF	bulle	ID	EX
$i + 8$	LD	R2, 0 (R1)					bulle	IF
$i + 9$	LD	R4, 0 (R3)						
$i + 10$	DADDU	R6, R2, R4						
$i + 11$	SD	R6, 0 (R5)						
$i + 12$	DADDIU	R1, R1, #8						
$i + 13$	DADDIU	R3, R3, #8						
			$t + 12$	$t + 13$	$t + 14$	$t + 15$	$t + 16$	$t + 17$
i	LD	R2, 0 (R1)						
$i + 1$	LD	R4, 0 (R3)						
$i + 2$	DADDU	R6, R2, R4						
$i + 3$	SD	R6, 0 (R5)						
$i + 4$	DADDIU	R1, R1, #8						
$i + 5$	DADDIU	R3, R3, #8						
$i + 6$	DADDIU	R5, R5, #8						
$i + 7$	BNE	R5, R7, Loop	MEM	WB				
$i + 8$	LD	R2, 0 (R1)	ID	EX	MEM	WB		
$i + 9$	LD	R4, 0 (R3)	IF	ID	EX	MEM	WB	
$i + 10$	DADDU	R6, R2, R4		IF	ID	bulle	EX	MEM
$i + 11$	SD	R6, 0 (R5)			IF	bulle	ID	EX
$i + 12$	DADDIU	R1, R1, #8					IF	ID
$i + 13$	DADDIU	R3, R3, #8						

Explications des bulles introduites dans le pipeline :

- Les bulles au cycle $t + 4$ et au cycle $t + 15$ sont causées par un aléa de données de type RAW via R4 entre les instructions $i + 1$ et $i + 2$ au cycle $t + 4$ et les instructions $i + 9$ et $i + 10$ au cycle $t + 15$, la lecture de la valeur en mémoire étant disponible à la fin de l'étage MEM.
- La bulle au cycle $t + 9$ est causée par un aléa de données de type RAW via R5 entre les instructions $i + 6$ et $i + 7$, 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 $t + 10$ est causée par un aléa de contrôle dû au branchement conditionnel de l'instruction $i + 7$, la direction et l'adresse de branchement étant connues à la fin de l'étage ID.

Question 2 (17 points sur 100)

Supposons que l'on a le choix entre deux systèmes informatiques à microprocesseurs pour effectuer un profil de traitement particulier présenté dans le tableau suivant.

Type d'instructions	Fréquence	CPI μ proc 1	CPI μ proc 2
Lecture en mémoire	0,22	8,0	10,0
Écriture en mémoire	0,09	15,0	10,0
Opérations ALU	0,23	1,0	1,0
Opérations à virgule flottante	0,19	4,8	8,7
Sauts et branchements conditionnels	0,20	7,0	4,2
Autres instructions	0,07	2,5	2,3

- (10) (a) Le microprocesseur 1 offre de meilleures performances pour ce profil de traitement que le microprocesseur 2. Déterminez le gain du microprocesseur 1 comparativement au deuxième microprocesseur.

Solution: Il faut d'abord calculer le CPI global de chaque microprocesseur. Ensuite, on fait le ratio de ces CPI pour déterminer le gain du microprocesseur 1 relativement à l'autre.

$$\begin{aligned}
 \text{CPI} &= \sum_i \frac{NI_i}{NI} \times \text{CPI}_i \\
 \text{CPI}_{\mu\text{proc1}} &= (0,22 \cdot 8,0) + (0,09 \cdot 15,0) + (0,23 \cdot 1,0) \\
 &\quad + (0,19 \cdot 4,8) + (0,20 \cdot 7,0) + (0,07 \cdot 2,5) = 5,827 \\
 \text{CPI}_{\mu\text{proc2}} &= (0,22 \cdot 10,0) + (0,09 \cdot 10,0) + (0,23 \cdot 1,0) \\
 &\quad + (0,19 \cdot 8,7) + (0,20 \cdot 4,2) + (0,07 \cdot 2,3) = 5,984 \\
 \text{gain}_{1\text{vs}2} &= \frac{\text{CPI}_{\mu\text{proc2}}}{\text{CPI}_{\mu\text{proc1}}} = \frac{5,984}{5,827} = 1,0269
 \end{aligned}$$

Donc, le microprocesseur 1 offre un gain de 2,69 % comparativement au microprocesseur 2.

- (7) (b) Supposons que l'on veut optimiser les opérations à virgule flottante du microprocesseur 2. Déterminez le CPI moyen pour ces opérations nécessaire afin que les performances du microprocesseur 2 soient comparables au microprocesseur 1 avec ce profil d'utilisation.

Solution:

$$\begin{aligned}
 \text{CPI}_{\mu\text{proc2mod}} &= (0,22 \cdot 10,0) + (0,09 \cdot 10,0) + (0,23 \cdot 1,0) \\
 &\quad + (0,19 \cdot x) + (0,20 \cdot 4,2) + (0,07 \cdot 2,3) \\
 &= 0,19x + 4,3310 = \text{CPI}_{\mu\text{proc1}} = 5,827 \\
 x &= \frac{5,827 - 4,3310}{0,19} = 7,87
 \end{aligned}$$

Avec un CPI moyen de 7,87 pour les opérations à virgule flottante, le microprocesseur 2 aurait une performance comparable au microprocesseur 1.

Question 3 (20 points sur 100)

Supposons un système informatique formé de 250 unités de traitement indépendantes, chaque unité comprenant un processeur, deux disques durs de 500 Go chacun et une infrastructure de communication. Ce système sert essentiellement au stockage de données dans un centre informatique. Supposons également que les données sont organisées en blocs de 1 Go, les blocs étant répartis sur l'ensemble des unités de traitement. Le système fait usage de redondance, de sorte que chaque bloc se retrouve copié sur deux unités de traitement distinctes et que chaque disque dur partage au moins un bloc avec chacun des autres disques. De plus, le système comporte trois serveurs d'indexation distincts, qui sont utilisés pour déterminer dans quel bloc se retrouve une donnée particulière lors d'une requête.

- (10) (a) Supposons que chaque disque dur a un temps moyen avant une panne (MTTF) de 200 000 heures et que les pannes sont indépendantes. Dans le cas d'une panne d'un disque dur, les blocs de ce disque sont recopiés sur les autres disques du système en 6 minutes, ce qui correspond au temps moyen de réparation (MTTR). Calculez le temps moyen en heures avant qu'un bloc de données soit définitivement perdu avec ce système. Donnez également le taux de pannes en FIT correspondant.

Solution: Une perte définitive de données survient lors d'une double faute, c'est-à-dire lorsqu'un deuxième disque dur tombe en panne alors que l'on répare un premier.

$$\begin{aligned}
 \text{MTTF}_{\text{sys}} &= \frac{\text{MTTF}_{\text{disque}}}{\text{nb. disques}} = \frac{200\,000}{2 \times 250} = 400 \text{ h} \\
 \text{proba}_{2\text{fautes}} &= (\text{nb. disques} - 1) \times \frac{\text{MTTR}_{\text{disque}}}{\text{MTTF}_{\text{disque}}} = (2 \times 250 - 1) \times \frac{0,1}{200\,000} \\
 &= 2,495 \times 10^{-4} \\
 \text{MTTF}_{2\text{fautes}} &= \frac{\text{MTTF}_{\text{sys}}}{\text{proba}_{2\text{fautes}}} = \frac{400}{2,495 \times 10^{-4}} = 1,6032 \times 10^6 \text{ h} \\
 \text{taux}_{2\text{fautes}} &= \frac{1}{\text{MTTF}_{2\text{fautes}}} = \frac{1}{1,6032 \times 10^6} = 6,2375 \times 10^{-7} \text{ h} = 623,75 \text{ FIT}
 \end{aligned}$$

- (10) (b) Supposons maintenant que lors d'une requête, le temps de recherche sur **un** serveur d'indexation correspond à 60 % du temps total, alors que le 40 % du temps restant permet de récupérer les données sur les unités de traitement. De plus, le temps de recherche dans les serveurs d'indexation est parallélisable à 75 %. Calculez le gain en réduction du temps de traitement d'une requête offert par l'ajout d'un quatrième serveur d'indexation, relativement à la configuration actuelle avec trois serveurs d'indexation.

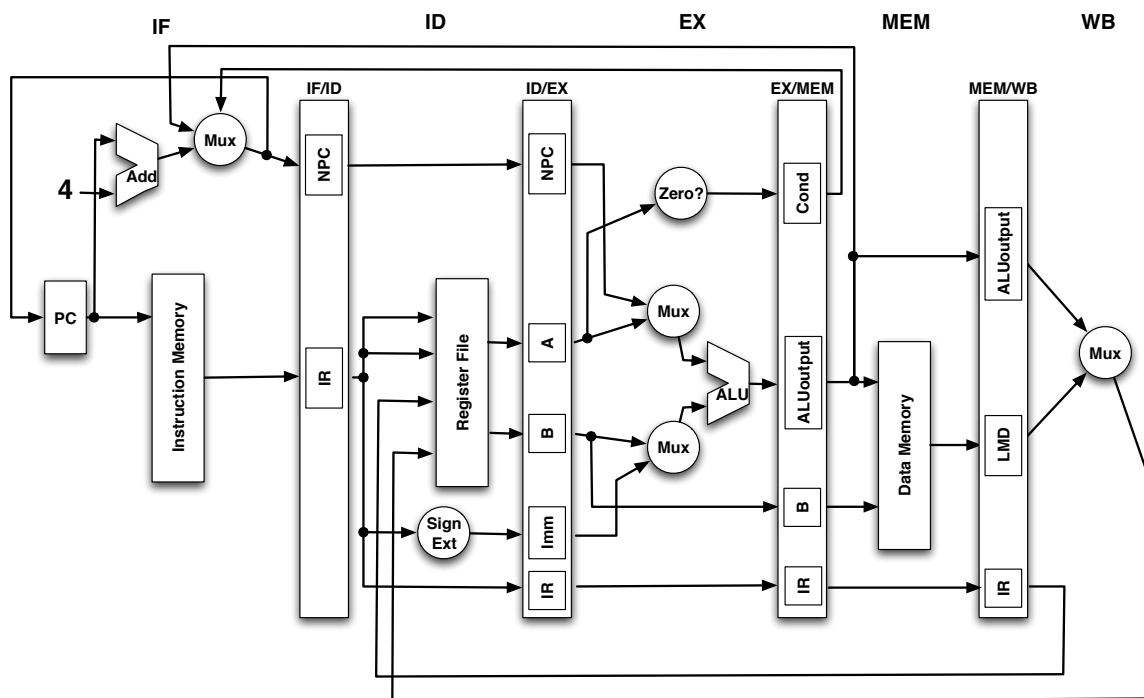
Solution: Il faut calculer les temps relatifs entre les deux configurations avant de pouvoir évaluer le gain.

$$\begin{aligned}
 \text{temps}_{\text{para}} &= 0,6 \times 0,75 = 0,45 \\
 \text{temps}_{\text{seq}} &= 1 - \text{temps}_{\text{para}} = 1 - 0,45 = 0,55 \\
 \text{temps}_{3\text{index}} &= \frac{\text{temps}_{\text{para}}}{3} + \text{temps}_{\text{seq}} = \frac{0,45}{3} + 0,55 = 0,7 \\
 \text{temps}_{4\text{index}} &= \frac{\text{temps}_{\text{para}}}{4} + \text{temps}_{\text{seq}} = \frac{0,45}{4} + 0,55 = 0,6625 \\
 \text{gain} &= \frac{\text{temps}_{3\text{index}}}{\text{temps}_{4\text{index}}} = \frac{0,7}{0,6625} = 1,0566
 \end{aligned}$$

Il y aura donc un gain en performance de 5,66 % avec un quatrième serveur d'indexation.

Question 4 (20 points sur 100)

Soit la figure suivante, qui présente la microarchitecture du MIPS (32 bits) implantée en pipeline, **sans envoi** de données, tel que vue 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 . Imm au début du cycle $t+3$.

Solution: La valeur du registre ID/EX . Imm au cycle $t+3$ correspond à la constante de l'instruction ADDIU R2, R2, #-1, soit -1.

- (4) (b) Donnez la valeur du registre IF/ID . IR au début du cycle $t+5$.

Solution: Le valeur du registre IF/ID . IR au cycle $t+5$ correspond au code machine de l'instruction $i+4$, soit SW R4, 16 (R1) .

- (4) (c) Donnez la valeur du registre EX/MEM . ALUoutput au début du cycle $t+7$.

Solution: Le registre EX/MEM . ALUoutput au cycle $t+7$ contient le calcul de l'adresse de l'écriture de donnée de l'instruction $i+3$, soit R1+8.

- (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 5 (28 points sur 100)

Répondez aussi brièvement et clairement que possible aux questions suivantes.

- (4) (a) Indiquez pourquoi dit-on que des aléas de données de type *Write-After-Write* (WAW) sont possibles avec des instructions ayant des temps d'exécution non uniformes, comme les opérations à virgule flottante avec MIPS, alors qu'avec des instructions aux temps d'exécution uniformes, ce n'est pas possible.

Solution: Avec des instructions aux temps d'exécution uniformes et une écriture aux registres qui se fait toujours au même étage, comme ce qui est fait dans MIPS, l'écriture se fera toujours dans l'ordre des instructions dans le programme, de sorte que les aléas de données de type WAW ne sont pas possibles. Avec des instructions aux temps d'exécution non uniformes, même si l'écriture aux registres se fait au même étage, l'ordre d'écriture peut ne pas respecter l'ordre d'exécution des instructions du programme, et donc générer des aléas de données de type WAW. Par exemple, si deux instructions d'opérations à virgule flottante consécutives et modifiant le même registre sont exécutées, si la deuxième instruction se termine avant la première, alors la valeur inscrite dans le registre destination correspondra à la valeur de la première instruction, ce qui ne respecte pas l'ordre des instructions du programme.

- (4) (b) Des architectures de processeurs tels que le x86 et l'ARM utilisent un code de condition afin de déterminer si un branchement est pris ou non. Indiquez le principal avantage et le principal désavantage de cette approche.

Solution: Avantage : parfois la valeur du code de condition est disponible « gratuitement », sans devoir faire d'opérations particulières.
Désavantage : l'utilisation d'un tel code contraint l'ordre d'exécution des instructions, et donc peut limiter le parallélisme d'instructions.

- (4) (c) Indiquez le désavantage principal associé à l'utilisation d'un jeu d'instructions de taille fixe, où la représentation en code machine de toutes les instructions occupe le même nombre de bits.

Solution: Un jeu d'instructions à taille fixe produit des programmes généralement plus gros qu'un jeu d'instruction de taille variable, ce qui peut être problématique en informatique embarquée, où la taille des programmes a de l'importance.

- (4) (d) Donnez des exemples d'architectures courantes d'ordinateurs de type SIMD (selon la taxonomie de Flynn) et ainsi que des architectures d'ordinateur de type MIMD.

Solution: SIMD : GPU, processeurs vectoriels MIMD : ordinateurs multi-cœurs, SMP, grappes de calcul, centres informatique

- (4) (e) Expliquez clairement ce que sont la bande passante et la latence dans les systèmes informatiques.

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) (f) Soit l'équation suivante, donnant le taux d'acceptation de dés dans la fabrication de circuits imprimés.

$$\text{Taux d'acceptation de dés} = \text{Taux d'acceptation de plaquettes} \times \frac{1}{(1 + \text{Défectuosité par unité d'aire} \times \text{Aire dé})^N}$$

Indiquez ce à quoi correspond la variable N dans cette équation relativement au processus de fabrication.

Solution: La variable N est une constante qui caractérise la complexité manufacturière du processus de fabrication.

- (4) (g) Expliquez de quelle façon on arrive à maintenir les exceptions précises dans un pipeline MIPS.

Solution: Lorsqu'une exception est rencontrée dans le pipeline MIPS, un vecteur de statut sauvegarde l'exception de l'instruction et les instructions suivant l'instruction fautive dans le pipeline sont désactivées. Le pipeline poursuit son exécution jusqu'à ce que l'instruction fautive ait atteint l'étage WB, où l'exception sera alors traitée. De cette façon, l'étage WB agit comme barrière de synchronisation permettant au processeur de traiter les exceptions dans l'ordre.

Résultats

Question:	1	2	3	4	5	Total
Points:	15	17	20	20	28	100
Score:						