

Architecture des ordinateurs (GIF-3000)

Département de Génie électrique et de Génie informatique

Automne 2017

EXAMEN PARTIEL #1
(solutions)

| Question | Points | Score |
|----------|--------|-------|
| 1 | 16 | |
| 2 | 20 | |
| 3 | 20 | |
| 4 | 44 | |
| Total: | 100 | |

Instructions :

- 1 feuille aide-mémoire manuscrite est permise (recto-verso) ;
- répondre sur le questionnaire ;
- durée de l'examen : 110 minutes.

Pondération : Cet examen compte pour 20% de la note finale.**Question 1** (16 points sur 100)

Un système critique comporte trois ordinateurs identiques, accomplissant les mêmes tâches. Lorsqu'ils produisent tous le même résultat, le système est considéré parfaitement fonctionnel. Lorsque deux sur trois produisent le même résultat, le 3e est considéré comme défectueux. Si les trois produisent des résultats différents, alors le système cesse d'opérer (n'est plus fonctionnel). Sachant que, pour chaque ordinateur, le temps moyen avant une défaillance (MTTF) est de 3 ans et que temps moyen de remplacement (MTTR) est de 4 jours :

- (i) (10 points) Calculez le taux de défaillance du système global, en supposant que les défaillances sont indépendantes et distribuées selon une loi exponentielle. Exprimez votre réponse en FIT.

Solution: Le temps moyen $MTTF_1$ avant d'avoir une première panne sera :

$$MTTF_1 = \frac{3 \text{ ans} \times 365 \text{ jours/an} \times 24 \text{ heures/jour}}{3} = \frac{26\,280 \text{ heures}}{3}$$

La probabilité P d'une seconde panne durant la réparation est donnée par :

$$P = 2 \times \frac{4 \text{ jours} \times 24 \text{ heures/jour}}{26\,280 \text{ heures}} = 7,3 \times 10^{-3}$$

Le temps moyen avant d'avoir une panne du système redondant sera donc :

$$MTTF_{\text{système}} = \frac{MTTF_1}{P} = \frac{26\,280 \text{ heures}}{3 \times 7,3 \times 10^{-3}} = 1\,199\,025 \text{ heures}$$

Le taux de défaillance T sera donc :

$$T = \frac{1}{1\,199\,025} = 8,34 \times 10^{-7} = 834 \text{ FIT}$$

- (ii) (6 points) Supposons maintenant que l'on veuille ajouter des fonctionnalités au système, mais que les processeurs d'origine à 1 seul cœur soient insuffisants pour satisfaire à la nouvelle tâche. En supposant que 30% des traitements pourraient être parallélisés, et en supposant que chacun des cœurs du nouveau processeur a la même performance que le processeur d'origine à 1 seul cœur, calculez le gain maximum de performance (*speedup*) que l'on peut **espérer** en faisant appel à des processeurs multicœurs ? Donnez votre réponse pour des processeurs à 2, 4 et 8 cœurs.

Solution: Le gain se calcule simplement par la loi d'Amdahl :

$$\text{speedup}(n) = \frac{1}{(1 - 0,3) + 0,3/n}$$

On obtient donc :

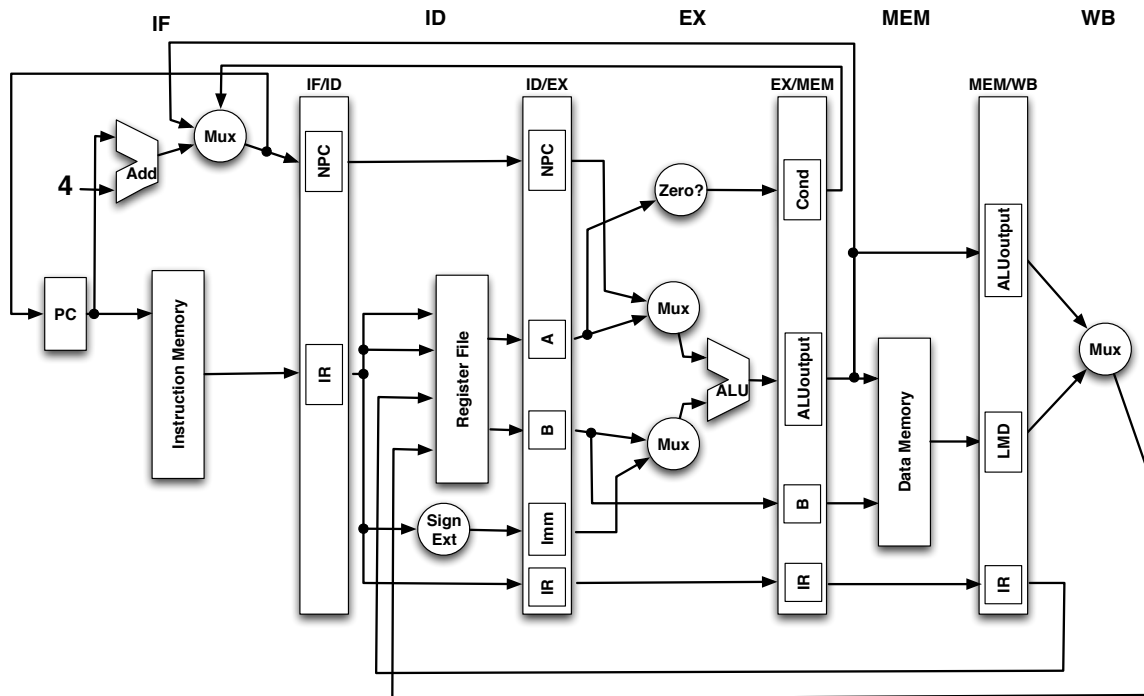
$$\text{speedup}(2) = \frac{1}{(1 - 0,3) + 0,3/2} = 1,18$$

$$\text{speedup}(4) = \frac{1}{(1 - 0,3) + 0,3/4} = 1,29$$

$$\text{speedup}(8) = \frac{1}{(1 - 0,3) + 0,3/8} = 1,36$$

Question 2 (20 points sur 100)

Soit la figure suivante, qui représente la micro architecture du MIPS implantée avec un pipeline à cinq phases, sans **chemin de traverse** (pas de «*forwarding*») ni optimisation pour les branchements, 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 | DADD | R3, R3, R4 | IF | ID | EX | MEM | WB | | | | |
| $i+1$ | DADDI | R2, R2, #-1 | | IF | ID | EX | MEM | WB | | | |
| $i+2$ | DADD | R4, R5, R4 | | | IF | ID | EX | MEM | WB | | |
| $i+3$ | SD | R3, 8 (R1) | | | | IF | ID | ID | EX | MEM | WB |
| $i+4$ | SD | 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.

- (i) (4 points) 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.

- (ii) (4 points) 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`.

- (iii) (4 points) 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`.

- (iv) (4 points) 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).

- (v) (4 points) Sans affecter la rectitude du programme, est-il possible d'améliorer la **performance** de ce code en changeant l'**ordre d'exécution** de certaines instructions ? Si oui, indiquez le ou les changements à faire et quel sera 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 3 (20 points sur 100)

Soit le programme 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

```

Simulez l'exécution de ce programme en bouclant autant de fois que nécessaire pour une durée totale de 17 cycles (du temps t au temps $t + 17$; supposez que la condition de branchement de l'instruction 6 est toujours vraie). Pour ce faire, remplissez les tableaux suivants en inscrivant les noms des **différentes phases** du pipeline, et en écrivant « *bulle* » pour **chaque aléa** rencontrés. Par la suite, expliquez brièvement la **cause** de chaque aléa.

- (i) (10 points) Dans un premier temps, supposez que le pipeline ne supporte aucun **chemin de traverse** (pas de *forwarding*) entre les étages du pipeline ni **aucune** optimisation des instructions de branchement (test de la condition durant la phase EX).

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

Explications des bulles introduites dans le pipeline :

- Puisque le DMUL dépend du résultat de l'instruction précédente (registre R3), le pipeline va insérer des nulles jusqu'à son WB, avant d'exécuter cette instruction.
- De même pour le SD suivant qui, avant de pouvoir s'exécuter, dépend du WB du DMUL.
- De même pour le BNE qui, avant de pouvoir s'exécuter, doit attendre le WB du DADDI.
- Finalement, le LD qui suit le BNE, doit attendre la phase EX du BNE, avant de pouvoir faire son IF.

- (ii) (10 points) Supposez maintenant que le pipeline comporte des **chemins de traverse** (*forwarding*) entre les étages du pipeline, et qu'il **optimise** les instructions de branchement.

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 |
| | | | $t + 6$ | $t + 7$ | $t + 8$ | $t + 9$ | $t + 10$ | $t + 11$ |
| $i + 2$ | 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 |
| | | | $t + 12$ | $t + 13$ | $t + 14$ | $t + 15$ | $t + 16$ | $t + 17$ |
| $i + 6$ | 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 4 (44 points sur 100)

Répondez brièvement et précisément aux questions suivantes.

- (i) (4 points) Quels sont les principaux facteurs qui influencent le **coût** des composants électroniques intégrés ?

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.

- (ii) (4 points) Expliquez pourquoi la **moyenne géométrique** est plus indiquée que la moyenne arithmétique pour comparer la performance des ordinateurs (pour un *benchmark suite*) ?

Solution:

Le ratio, pour les deux ordinateurs, des moyennes géométriques des performances (relatives p. ex. SPECratio ou absolue) pour chaque benchmark de la suite sera égal à la moyenne géométrique des ratios de performance (relative p. ex. SPECratio ou absolue) pour chaque benchmark de la suite. Ainsi, i) on peut calculer les moyennes de la même façon avec la performance relative et la performance absolue, ii) dans les deux cas, les moyennes peuvent être calculées une seule fois pour chaque ordinateur et la référence et ensuite utilisées pour comparer les ordinateurs entre eux ou avec la référence et iii) chaque benchmark de la suite a le même poids dans la comparaison des performances.

- (iii) (4 points) Expliquez la différence entre la **latence** et le **débit** ?

Solution:

Le débit mesure une quantité de travail ou une quantité d'information produite par unité de temps. Par exemple, le nombre d'instructions exécutées par seconde.

La latence est une mesure de délai entre le début et la fin d'une action ou d'un processus. Par exemple, la latence d'exécution d'une instruction est le délai entre le début et la complétion de l'instruction.

- (iv) (4 points) Quel est l'impact d'un pipeline sur la **latence** et le **débit** ?

Solution:

La latence d'exécution dans un microprocesseur correspond au nombre de cycles d'horloge nécessaire pour compléter l'exécution d'une instruction. Le pipeline ne réduit pas la latence d'exécution, car une instruction prend toujours autant de temps à s'exécuter avec ou sans pipeline. Par contre, le pipeline a le potentiel d'augmenter le débit par facteur proportionnel à la profondeur du pipeline.

- (v) (4 points) Pourquoi les jeux d'instruction **mémoire-mémoire** sont-ils aujourd'hui désuets ?

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 termes d'implantation matérielle, car une instruction peut correspondre à plusieurs accès mémoire suivis d'une opération arithmétique. Cette complexification rend très difficile l'utilisation des avancées architecturales développées depuis les années 1980, telles que les pipelines.

- (vi) (4 points) Expliquez en quoi consiste un **aléa de branchement**.

Solution:

Un aléa de branchement 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.

- (vii) (4 points) Expliquez en quoi consiste la **gestion précise** des exceptions.

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 n'ont modifié l'état du programme.

- (viii) (4 points) Les architectures **RISC** ont été proposées au moment où les **langages de programmation** compilés, tels que le langage C, ont gagné en popularité. Expliquez la relation entre ces deux technologies et la synergie 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.

- (ix) (4 points) Quels sont les principaux facteurs qui influencent la **puissance dissipée** dynamiquement par les microprocesseurs et, pour chacun d'entre eux, dites comment celle-ci est affectée ?

Solution:

Les facteurs sont : 1) la charge capacitive des circuits, 2) la tension d'alimentation et 3) la fréquence d'horloge. La puissance dépend linéairement de la charge capacitive et de la fréquence, et de façon quadratique par rapport à la tension d'alimentation.

- (x) (4 points) Qu'est-ce que la règle empirique du **90-10** et comment se traduit-elle dans le **principe de localité** (temporelle et spatiale) ?

Solution:

La règle du 90-10 spécifie que les programmes passent souvent 90% (parfois davantage) de leur temps d'exécution dans seulement 10% du code (parfois moins). De même pour les données : 90% du temps d'exécution pour traiter 10% des données. Il s'agit d'une règle très approximative, mais fréquemment observée. Cela se traduit dans ce qu'on appelle le principe de localité : (temporelle) les items accédés récemment risquent d'être accédés prochainement ; (spatiale) les accès séquentiels (ou rapprochés) en mémoire sont plus fréquents que les accès aléatoires.

- (xi) (4 points) Expliquez ce qu'est un aléa de type **Write-After-Write** (WAW) et dites dans quelle situation il peut notamment se produire ?

Solution:

Un aléa de type WAW survient lorsque deux écritures dans un registre ou deux écritures en mémoire ne surviennent pas dans l'ordre des instructions qui les ont produits. Ainsi, une écriture qui devrait normalement se produire à la suite d'une autre écriture, parce que résultant d'une instruction subséquente, se réalise trop tôt et sera éventuellement écrasée par une écriture qui aurait dû se réaliser plus tôt. Ce type d'aléa se produit lorsque le nombre de phases du pipeline est variable. Ceci se produit notamment avec les opérations à virgule flottante.