

Evaluation formative

Département de génie électrique et de génie informatique
Université de Sherbrooke
Architecture et organisation des ordinateurs — S4i unité 4

Été 2022

1. *GIF310-3* En vous inspirant du laboratoire 2, rédigez un plan de vérification pour les accès en lecture d'une cache pleinement associative avec remplacement par *LRU* (*Least Recently Used*), dotée de 8 blocs contenant chacun 2 mots de 4 octets.
2. (*GIF310-3*) Soit un processeur dont l'adressage mémoire est défini sur 32 bits auquel on associe une mémoire cache à correspondance directe contenant 1024 blocs de données, contenant chacun 16 mots de 32 bits. L'étiquette lue à l'index `0x014` correspond à `0xB592`.
 - (a) Quelle est la plage d'adresses mémoire (adresses inférieures et supérieures, en hexadécimal) associée aux octets dans ce bloc de la mémoire cache?
 - (b) Si la cache était plutôt pleinement associative, quelle serait la valeur en binaire de l'étiquette associée pour les mêmes octets?

3. *GIF310-1* En ce qui a trait à la performance en lecture d'une mémoire cache de données, la dimension des blocs de celle-ci affecte le taux d'échec (*taux_echec*) et la pénalité d'échec (*penalite_echec*).

Étant donné que :

- La mémoire fournit une donnée par accès, avec un coût d'accès de 20 cycles. Puisque lors d'un échec il faut remplir le bloc au complet, la pénalité d'échec est proportionnelle à la grandeur des blocs selon

$$penalite_echec = 20 \times taille_bloc;$$

- le taux d'échec varie en fonction de la taille de bloc comme suit:

taux d'échec	taille de bloc
8,0%	8
3,0%	16
1,8%	32
1,5%	64
2,0%	128

- la machine tourne à 1 CPI (cycle par instruction);
- le nombre moyen d'accès à la mémoires par le programme (*taux_ref*) est de 35% des instructions.

Question: Déterminez la dimension optimale des blocs dans ce contexte. Développez l'équation algébrique, puis organisez vos calculs dans une feuille excel.

4. *GIF310-1* Question 1.7 de COD5 (section 1.13). Traduction libre (aidé de [deepl.com](https://www.deepl.com/)):

Les compilateurs peuvent avoir un impact significatif sur les performances d'une application sur une organisation pipeline. Supposons que pour un programme, le compilateur A donne lieu à un code assembleur de 1,0E9 instructions et à un temps d'exécution de 1,1 secondes, tandis que le compilateur B donne lieu à un code assembleur de 1,2E9 instructions et a un temps d'exécution de 1,5 secondes.

- a) Trouvez le CPI moyen pour chaque programme étant donné que le processeur a un temps de cycle d'horloge de 1 ns.
- b) Supposons que les programmes compilés s'exécutent sur deux processeurs différents. Si les temps d'exécution sur les deux processeurs sont les mêmes, quelle est la vitesse de l'horloge du processeur exécutant le code du compilateur A par rapport à l'horloge du processeur exécutant le code du compilateur B ?
- c) Un nouveau compilateur est développé qui utilise seulement 6,0E8 instructions et a un CPI moyen de 1,1. Quelle est la vitesse d'exécution de ce nouveau compilateur par rapport à celle du compilateur A ou B sur le processeur original ?

5. *GIF310-2* Questions sur le langage assembleur.

- (a) Qu'est-ce qu'une instruction avec un mode d'adressage non-soutenu? En quoi est-ce différent d'une pseudo-instruction?
- (b) Comment fait-on en langage assembleur MIPS pour transférer un mot de 32 bits de l'adresse `0x10010000` à l'adresse `0x100100F8`? Écrivez un court segment de code.
- (c) Quelle est la différence entre les instructions `sra` et `srl`?

6. Performance et pipeline

Important: à faire absolument dans Moodle. Reproduit ici en référence additionnelle.

Étant donné le segment de code suivant pour architecture MIPS:

```
1      or $v0, $0, $0
2 L1:  lw $t1, 0($a0)
3      subu $v0, $v0, $t1
4      add $a0, $a0, $s4
5      add $t2, $t2, $s2
6      bne $t2, $0, L1
7      and $v0, $v0, $s3
8      sw $v0, 0($a0)
```

où on considère que

- `$t2` contient initialement `0xff`;
- `$s2` contient initialement `0xffffffff`;
- `$s3` contient initialement `0xffff`;
- `$s4` contient initialement `0x4`;
- `$a0` contient initialement l'adresse quelconque d'un tableau de mots de 32 bits.

On souhaite déterminer la performance de ce code sur 2 réalisations distinctes de l'architecture MIPS, soit

- **Processeur 1:** machine à cycle unique tournant à 100 MHz (COD5, fig. 4.17, p. 265, en annexe);
 - **Processeur 2:** machine à pipeline avec gestion matérielle des aléas et unité d'envoi, cadencée à 350 MHz (COD5, fig. 4.65, p. 325, en annexe). Les branchements sont toujours prédits comme étant "non pris", avec vidange matériel au besoin.
- (GIF310-1) Déterminer la durée d'exécution totale de ce segment de code en nombre de cycles d'horloge pour le processeur unicycle. Laissez des traces de votre démarche de calculs.
 - (GIF310-3) Pour l'organisation pipeline décrite ci-haut, identifiez les aléas dans ce segments. Pour ces cas, identifiez le ou les registres causant l'aléa, s'il y a lieu.
 - (GIF310-3) L'unité de détection des aléas compense pour les problèmes que l'unité d'avancement ne peut pas régler. Combien de type de corrections existe-t-il et en quoi sont-ils différents?
 - (GIF310-1) Déterminer le nombre de cycles d'horloge supplémentaires de ce segment de code dans ce processeur pipeline.
 - (GIF310-3) Décrire l'état du pipeline (quelles instructions sont à quel étage) au cycle numéro 10 sachant que la ligne 1 est à l'étage IF au 1er cycle.

7. (GIF310-3) On considère le code suivant:

```
1      lw $t0, 40($6)
2      add $t1, $t3, $t0
3      add $t0, $t5, $t4
4      sw $t1, 20($t4)
5      and $t0, $t0, $t4
```

- Que se passe-t-il si on fait tourner ce code tel quel sur un processeur MIPS à pipeline à 5 étages sans détection d'aléas ni unité d'avance (figure 4.51 du COD5)?

- (b) Modifier ce programme en introduisant des **nop** pour qu'il fonctionne correctement sur ce processeur sans détection d'aléas ni unité d'avance.
- (c) On souhaite maintenant ajouter une unité de détection des hazard qui insérerait automatiquement des suspensions/bulles (modification à la figure 4.51 du COD5). Comment ce module pourrait-il déterminer quand insérer des bulles pour régler le 1er aléa? C'est-à-dire, quelle équation(s) logique(s) pourrai(en)t générer un signal de requête de suspension?

8. (GIF310-3) On considère l'architecture MIPS dans une organisation à cycle unique (référence COD5, figure 4.17 page 265). On désire ajouter l'instruction **jr** (saut par registre).

- (a) Quelles modifications sont nécessaires au chemin de données et de contrôles? On vous impose d'effectuer vos modifications à l'extérieur des bloc « banc de registre » et « ALU ». Décrivez quelles parties existantes seront mises à profit, justifiez par un court texte les ajouts à faire et annotez ces ajouts supplémentaires au schéma ci-bas.
- (b) Dressez la liste de tous les signaux de contrôle ainsi que leur valeur lorsque cette nouvelle instruction est active. Pour l'ALU, indiquez plutôt le type d'opération effectuée (ADD, AND, NULL, etc).

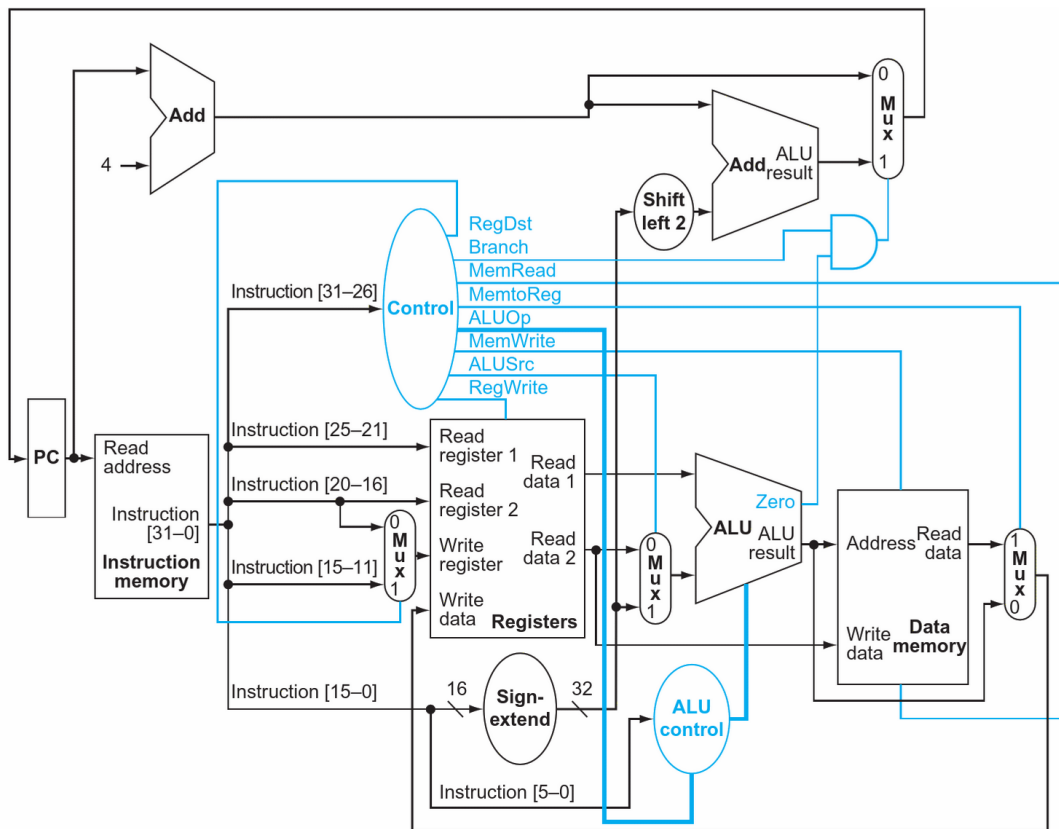
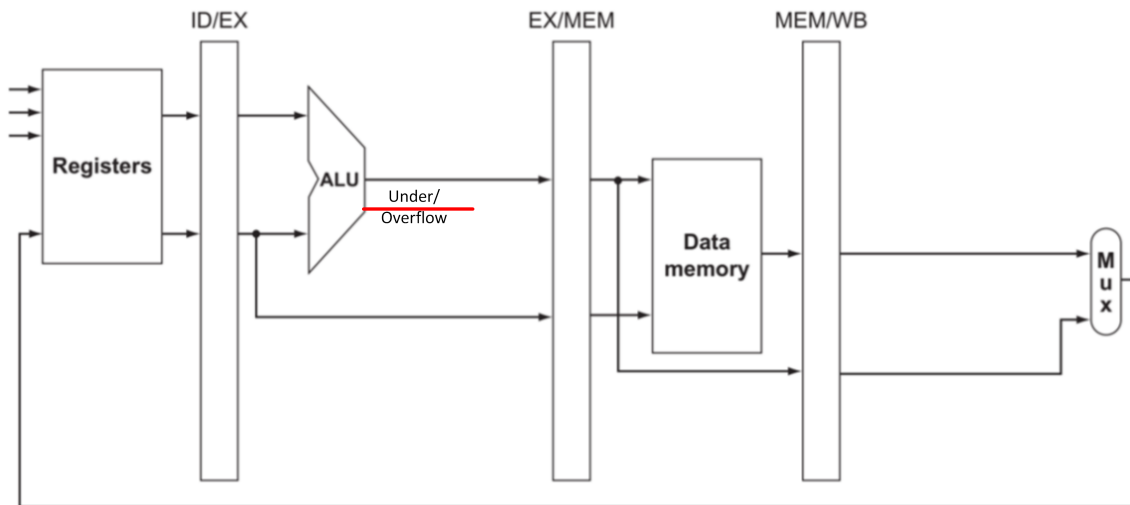


FIGURE 4.17 The simple datapath with the control unit.

9. (GIF310-3) Soit l'architecture MIPS selon une organisation pipeline sans unité d'avancement (référence COD5 fig. 4.51), avec sa représentation simplifiée ci-bas (sans module de contrôle explicite). On souhaite lui ajouter une nouvelle instruction d'addition signée avec saturation (*addsat*), c'est à dire qu'au lieu d'avoir un retournement de bit de signe, le résultat demeure à la valeur maximale (ou minimale pour les nombres négatifs) permise par la représentation en complément 2. Fait intéressant, vous remarquerez que l'ALU a un bit indiquant la présence ou non d'un retournement. Modifiez l'organisation pour ajouter cette nouvelle instruction *addsat*, mais sans changer l'ALU.



10. Divers

- Dans le processeur à pipeline, quel avantage y a-t-il à avoir une mémoire de données et une mémoire d'instructions distinctes?
- Quel est le rôle des bascules qui séparent les étages d'un pipeline?
- Quels sont les principes liés à l'exécution d'un programme typique qui justifient l'utilisation d'une mémoire cache pour améliorer les performances (qui font que le taux de succès est relativement élevé)?
- Comment a été déterminée la valeur initiale du registre `$sp` dans le processeur MIPS unicycle de référence?

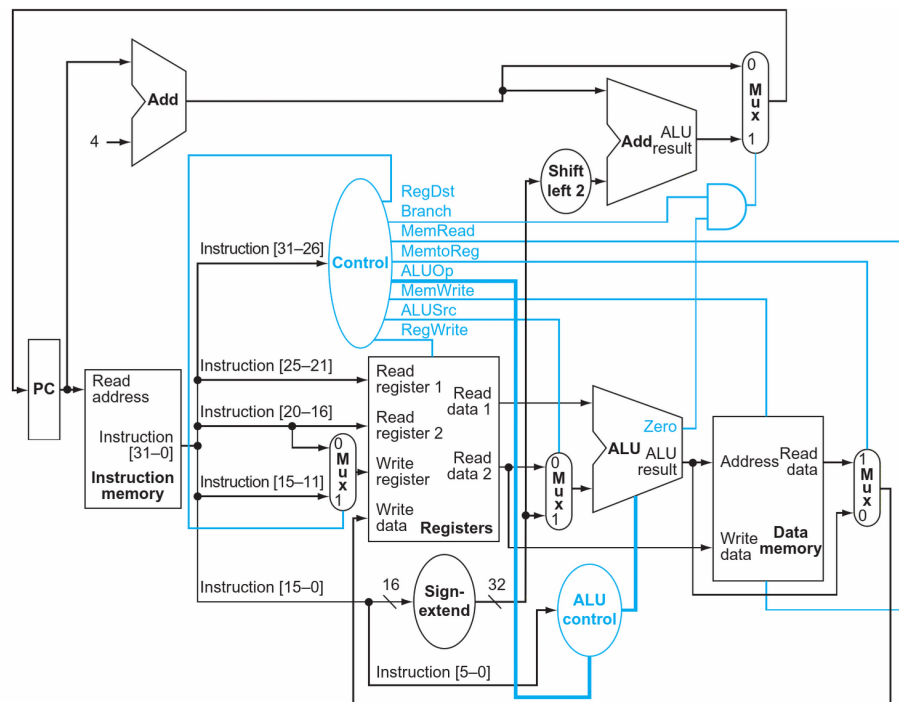
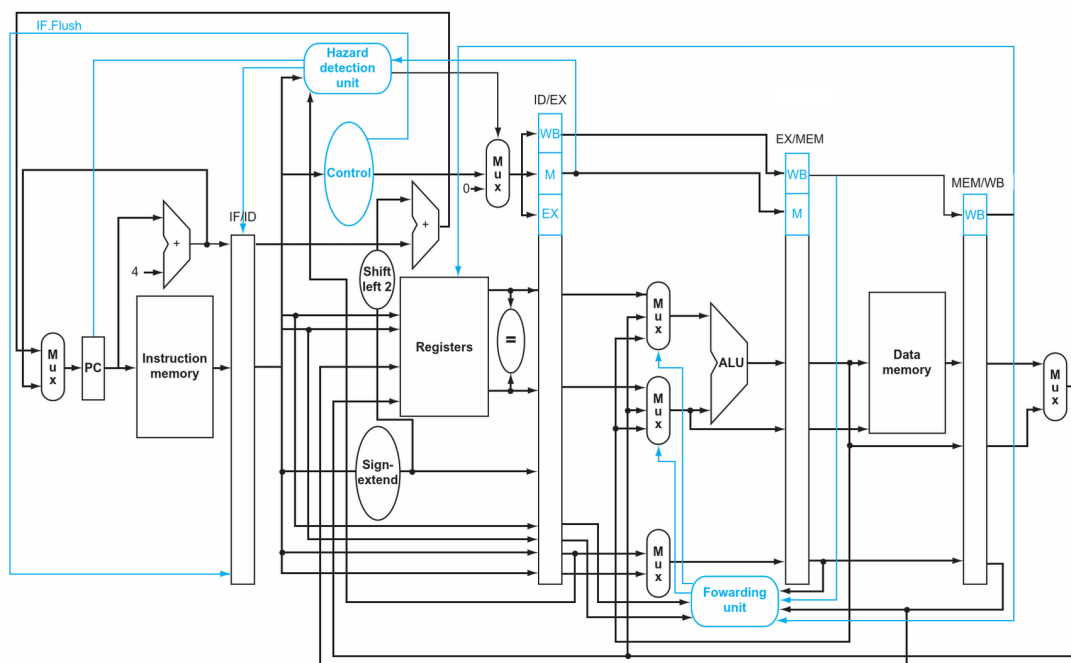
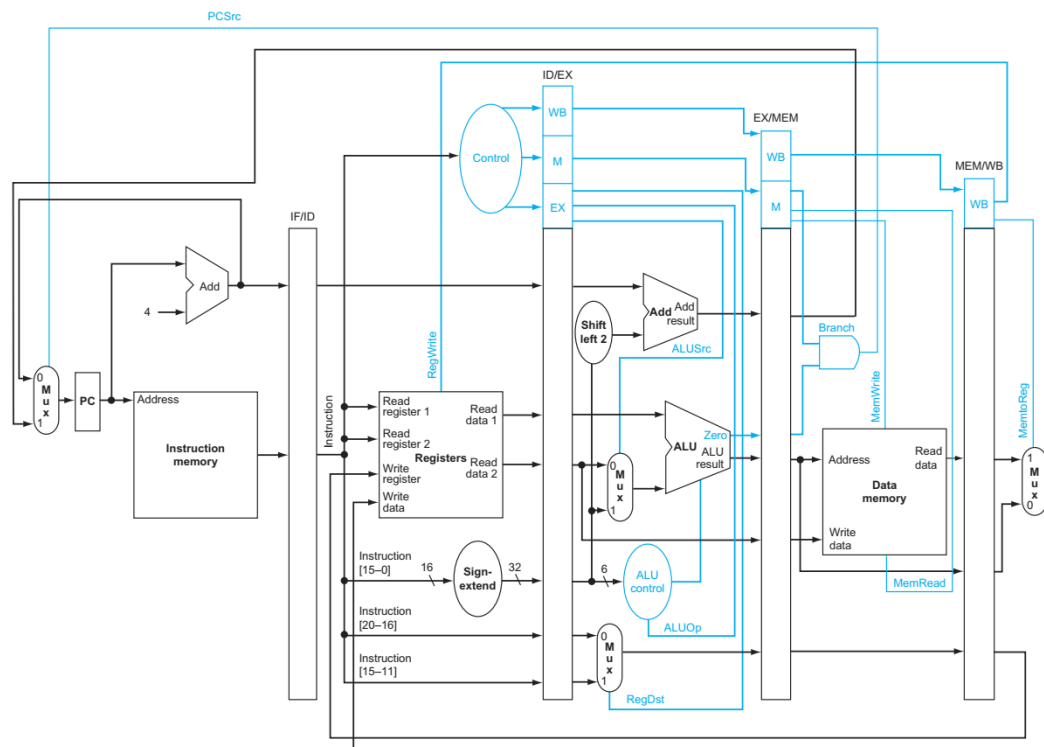


FIGURE 4.17 The simple datapath with the control unit.



Copyright © 2022
Département de génie électrique et de génie informatique
Université de Sherbrooke