# Université d'Alger 1

Faculté des Sciences
Département Informatique

# Rattrapage L2-ADO
14-Mai-2022

| | | |
|---|---|---|
| Nom :<br>Prénom :<br>Groupe :<br>Matricule : | | / 20 |

## Exercice 1 : (5 points)

Considérez la séquence de code MIPS suivante :

```
lw $5, 100($2)
add $2, $3, $5
sub $5, $5, $2
sw $5, 100($2)
```

(5 pts) Identifiez toutes les dépendances entre les paires d'instructions.

```
lw $5, 100($2) et add $2, $3, $5  # registre $5
lw $5, 100($2) et sub $5, $5, $2  # registre $5
add $2, $3, $5 et sub $5, $5, $2  # registre $2
add $2, $3, $5 et sw $5, 100($2)  # registre $2
sub $5, $5, $2 et sw $5, 100($2)  # registre $5
```

## Exercice 2 : (15 points)

Soit le fragment de code MIPS suivant :

```
I0: addi $3, $0, 100    # $3 = 100
I1: add  $4, $0, $0     # $4 = 0
Loop:
I2: lw   $5, 0($1)      # $5 = MEM[$1]
I3: add  $4, $4, $5     # $4 = $4 + $5
I4: lw   $6, 0($2)      # $6 = MEM[$2]
I5: sub  $4, $4, $6     # $4 = $4 - $6
I6: addi $1, $1, 4      # $1 = $1 + 4
I7: addi $2, $2, 4      # $2 = $2 + 4
I8: addi $3, $3, -1     # $3 = $3 - 1
I9: bne  $3, $0, Loop   # if ($3 != 0) goto Loop
```

**a)** (10 pts) Illustrez l'exécution d'une itération de la boucle sur un processeur MIPS ayant un pipeline de 5 étages. Le processeur est équipé d'une unité de suspension produisant un cycle de retard éventuel, **mais ne possède pas d'unité de transfert**. Indiquez toutes les bulles de suspension et cycles de retard.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I0 : addi | IF | ID | EX | * | WB | | | | | | | | | | | | | | | | | | | | |
| I1 : add | | IF | ID | EX | * | WB | | | | | | | | | | | | | | | | | | | |
| I2 : lw | | | IF | ID | EX | M | WB | | | | | | | | | | | | | | | | | | |
| I3 : add | | | | IF | | | ID | EX | * | WB | | | | | | | | | | | | | | | |
| I4 : lw | | | | | | | IF | ID | EX | M | WB | | | | | | | | | | | | | | |
| I5 : sub | | | | | | | | IF | | | ID | EX | * | WB | | | | | | | | | | | |
| I6 : addi | | | | | | | | | | | IF | ID | EX | * | WB | | | | | | | | | | |
| I7 : addi | | | | | | | | | | | | IF | ID | EX | * | WB | | | | | | | | | |
| I8 : addi | | | | | | | | | | | | | IF | ID | EX | * | WB | | | | | | | | |
| I9 : bne | | | | | | | | | | | | | | IF | | | ID | | | | | | | | |
| I2 : lw | | | | | | | | | | | | | | | | | IF | IF | ID | EX | M | WB | | | |
| I3 : add | | | | | | | | | | | | | 1 cycle de retard | | | | | | IF | | | ID | EX | * | WB |

**b)** (5 pts) Réorganisez les instructions de la boucle ci-dessus pour éliminer les bulles de suspension et cycles de retard **sans que cela affecte le calcul**. Écrivez le code de la boucle modifiée (justifiez vos choix !).

```
I0: addi $3, $0, 100    #
I1: add  $4, $0, $0     #
Loop:
I2: lw   $5, 0($1)      #
I4: lw   $6, 0($2)      #
I6: addi $1, $1, 4      #
I3: add  $4, $4, $5     #  déplacé plus tard pour éviter de suspendre à cause de I2 (et I1 initial)
I8: addi $3, $3, -1     #  déplacé plus tôt pour éviter tout retard de branche
I7: addi $2, $2, 4      #
I5: sub  $4, $4, $6     #  déplacé ici pour éviter de suspendre à cause de I3 et I4
I9: bne  $3, $0, Loop   #
```