



Correction Examen Final

Questions de cours – répondre brièvement (10 pts) :

1. Quels sont les liens physiques entre le processeur et la mémoire ?

le bus mémoire est constitué :

- du bus d'adresse
- du bus de données
- du bus de contrôle

2. Que comporte l'unité de contrôle ?

L'unité de contrôle comporte :

- Un **registre instruction** (RI) : contient l'instruction en cours d'exécution. Chaque instruction est décodée selon son code opération grâce à un décodeur.
- Un registre qui s'appelle **compteur ordinal** (CO) ou le **compteur de programme** (CP) : contient l'adresse de la prochaine instruction à exécuter (pointe vers la prochaine instruction à exécuter). Initialement il contient l'adresse de la première instruction du programme à exécuter.
- Un **séquenceur** : il organise (synchronise) l'exécution des instructions selon le rythme de l'horloge, il génère les signaux nécessaires pour exécuter une instruction.

3. Quels sont les différents aléas du pipeline ?

- **aléa structurel** : des parties du chemin de données doivent être utilisées simultanément par plusieurs étages du pipeline
- **aléa de données** : le calcul d'une valeur à un étage du pipeline nécessite une valeur non encore calculée
- **aléa de contrôle** : l'instruction suivante dépend d'une valeur calculée

4. Donnez une solution pour chaque type d'aléa ?

- Aléa structurel :
 - attendre que l'unité soit disponible en retardant l'exécution : peu efficace
 - dupliquer les différentes unités du chemin de données :
ici, accès mémoire : découpage en 2 parties du cache L1
 - ▶ LE accède à la partie "instruction"
 - ▶ EX accède à la partie "données"
- Aléa de donnée :
 - changer l'ordre d'exécution des instructions : **réordonnement** (réalisé soit à la compilation, soit par le processeur à la volée)
 - Aléa de contrôle :
 - attendre que le résultat de l'opération soit connue : peu efficace (ici, le chargement de l'instruction (LE) ne peut se faire qu'après l'écriture dans $\$1$)
 - réaliser une prédiction de branchement et commencer le calcul avec celle-ci

5. Quelles sont les relations possibles entre les niveaux de cache L1 et L2 ?

- Cache inclusif :
 - ▶ le contenu de L1 est également dans L2 (Le contenu de L1 est inclus dans L2) (L1 contient une copie de L2)
 - ▶ L2 est une copie partielle de la mémoire centrale et L1 est une copie partielle de L2
 - ▶ taille de la mémoire cache = taille de L2
- Cache exclusif :
 - ▶ le contenu de L1 n'est pas présent dans L2 (Les contenus de L1 et L2 sont exclus mutuellement)
 - ▶ L2 contient le trop-plein de L1 : quand L1 est plein, on transvase vers L2 et inversement si une données redevient intéressante.
 - ▶ taille de la mémoire cache = taille de L1 + taille de L2

6. Quelles sont les méthodes utilisées pour assurer la cohérence des données entre la mémoire cache et la mémoire principale lors d'une opération d'écriture ?

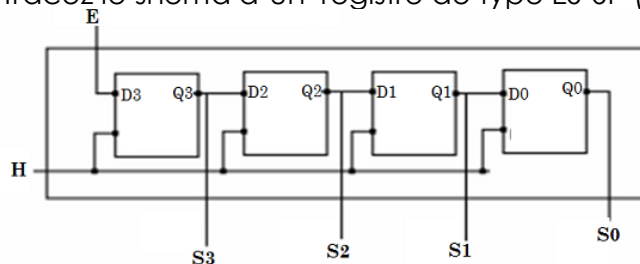
- Write through et write back

7. Complétez la table de vérité suivante associée à la figure :

h	J	K		Q+
0/1	x	x		Q-
↑	0	0		Q-
↑	0	1		0
↑	1	0		1
↑	1	1		Q



8. Tracez le schéma d'un registre de type ES-SP (à 3 bits)



Exercice n°1(6 pts)

```
.data
T: .word 5,7,-1,6,0,-15,9,4,-2,-18
R: .word 0,0,0,0,0,0,0,0,0,0
size: .word 10
separateur: .asciiz "|"
message: .asciiz "le tableau trié est : "
.text
la $t1,T
la $t2,R
lw $a1,size
li $s1,1
li $s2,1

jal trier
jal afficher

li $v0,10
syscall
```

trier:

```
boucle_pos: bgt $s1, $a1, boucle_neg # remplir les valeurs >=0
            lw $t3, 0($t1)
            bgez $t3, positif
            addi $t1, $t1, 4
            addi $s1, $s1, 1
            j boucle_pos
```

```
positif:    sw $t3, 0($t2)
            addi $t1, $t1, 4
            addi $t2, $t2, 4
            addi $s1, $s1, 1
            addi $s2, $s2, 1
            j boucle_pos
```

```
boucle_neg: la $t1, T
            li $s1, 1
```

```
boucle_neg_int: bgt $s1, $a1, fin_trier # remplir les valeurs <0
                lw $t3, 0($t1)
                bltz $t3, negatif
                addi $t1, $t1, 4
                addi $s1, $s1, 1
                j boucle_neg_int
```

```
negatif:     sw $t3, 0($t2)
            addi $t1, $t1, 4
            addi $t2, $t2, 4
            addi $s1, $s1, 1
            addi $s2, $s2, 1
            j boucle_neg_int
```

fin_trier: jr \$ra

afficher:

```
            la $t2, R
            li $s2, 1
```

```
debut_aff:  bgt $s2, $a1, fin_aff
            lw $a0, 0($t2)
            li $v0, 1
            syscall
```

```
            la $a0, separateur
            li $v0, 4
            syscall
```

```
            addi $t2, $t2, 4
            addi $s2, $s2, 1
            j debut_aff
```

fin_aff: jr \$ra

Exercice n°2 (4 pts) :

```
.data
vala:.asciiz "saise val de a:"
valb:.asciiz "saise val de b:"
val:.asciiz "pgcd "

.text
li $v0,4
la $a0,vala
syscall
li $v0,5
syscall
move $t1,$v0
li $v0,4
la $a0,valb
syscall
li $v0,5
syscall
move $t2,$v0

jal pgcd

move $a0,$v0
move $t3,$a0
li $v0,4
la $a0,val
syscall
move $a0,$t3
li $v0,1
syscall
li $v0,10
syscall

pgcd:
while:
beq $t1,$t2,sortir    # if a=b sort
bgt $t1,$t2,lab1      #if a>b go to lab1
lab2:
sub $t2,$t2,$t1    # a<b subb b:=a-b
j while
lab1:
sub $t1,$t1,$t2    # a>b then a:=a-b
#bgt $t2,$t1,lab2    # if b>a go to lab2
j while

sortir:
move $v0,$t1
jr $ra
```