Département informatique
2ème Année Licence
Module : Architecture des ordinateurs

Correction Examen Final

Questions de compréhension (3 pts) :

Justifier le vrai et corriger le faux dans ce qui suit :

1. Les DRAM sont des mémoires utilisées dans la réalisation des mémoires caches.
2. Dans un système à microprocesseur le disque dur ne peut pas jouer le rôle d'une mémoire principale.
3. Dans un microprocesseur les accumulateurs sont des registres d'adresses (pointeurs) connectés sur le bus adresses.
4. Le nombre de fils de données d'un boîtier mémoire définit le nombre de cases mémoire que comprend le boîtier.
5. Le mode d'adressage définit la manière dont le microprocesseur va exécuter l'opération.
6. Lors d'un aléa structurel, des parties du chemin de données doivent être utilisés en parallèle par plusieurs étages du pipeline.

N°	Vrai / Faux	Réponses
1	**Faux**	Les DRAM sont utilisés pour la réalisation des mémoires principales.
2	**Vrai**	Moins rapide que la mémoire principale (temps d'accès très élevé par rapport à celui de la DRAM) et comme conséquence directe diminution de la vitesse d'exécution du processeur.
3	**Faux**	L'accumulateur est un registre de travail qui sert à stocker un opérande au début d'une opération et le résultat à la fin de l'opération.
4	**Faux**	Le nombre de fils de données d'un boitier mémoire définit la largeur des mots mémoires.
5	**Faux**	Le mode d'adressage définit la manière dont le microprocesseur va accéder à l'opérande.
6	**Faux**	Lors d'un aléa structurel, des parties du chemin de données doivent être utilisés simultanément par plusieurs étages du pipeline

Exercice n°1 (2 pts) :

1. Commenter le programme suivante et indiquer ce que fait la fonction F(x,y,z). On suppose que la variable x se trouve dans le registre R3, Y se trouve dans le registre R2, z se trouve dans le registre R5 et le résultat dans le registre R7.

```
Add R7, R3, R3
Add R7, R7, R7
Add R7, R7, R5
Add R7, R7, R5
Add R7, R7, R2
```

Réponse

Ce programme évalue une expression algébrique en utilisant des additions successives pour les multiplications :

$$F(x,y,z) = 4x + 2z + y$$

2. Indiquer la durée d'exécution sur un processeur à un cycle ayant une fréquence d'horloge de 700 Ghz pour le programme précédent ?

Réponse

Processeur à un cycle $\Rightarrow$ durée instruction = durée d'un cycle

Pour 5 instructions $\Rightarrow$ Durée exécution = 5 * durée cycle.

$7*10^{11}$ cycle $\rightarrow$ 1s

1 cycle $\rightarrow$ x $\Rightarrow$ $x = 1.4 * 10^{-10}$ s = 14 ns

$\Rightarrow$ durée exécution programme = $5 * 14 * 10^{-9}$ = 70 ns

3. Pourquoi le processeur à un cycle ne représente pas une architecture de processeur qui soit performante ?

Réponse

Parce que le temps de cycle est égal au temps d'exécution de l'instruction la plus longue (Chargement), ce cycle fixé est trop long pour les autres instructions (arithmétiques). Le processeur cumul un temps d'inactivité important en attendant les fins de cycles. Ces temps d'attentes se répercutent négativement sur sa performance.

Exercice n°2 (5 pts) :

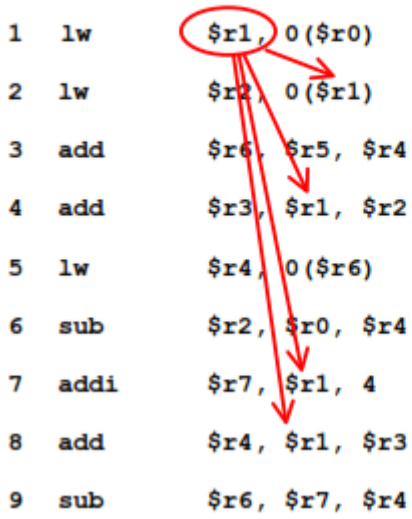
Soit la séquence d'instruction suivante :

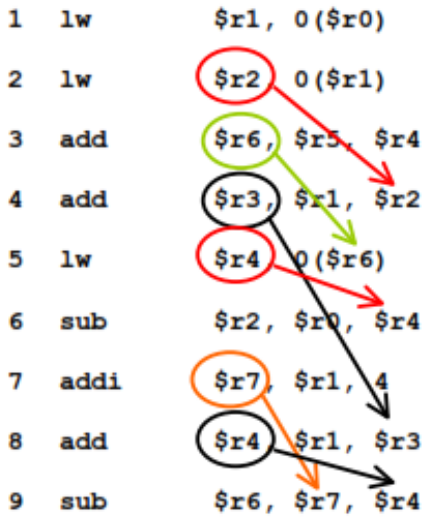
```
1 lw      $r1, 0($r0)
2 lw      $r2, 0($r1)
3 add     $r6, $r5, $r4
4 add     $r3, $r1, $r2
5 lw      $r4, 0($r6)
6 sub     $r2, $r0, $r4
7 addi    $r7, $r1, 4
8 add     $r4, $r1, $r3
9 sub     $r6, $r7, $r4
```

Question : Identifiez tous les aléas possibles dans la séquence d'instruction ci-dessus ?

Réponse

- *Dépendances RAW (Read After Write):*
  - *Après chaque écriture d'un registre, vérifier si le même registre est lu*





- *Dépendances WAR (Write After Read):*
  - *Après chaque lecture d'un registre, vérifier si le même registre est destination*

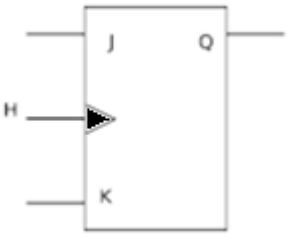
```
1   lw      $r1, 0($r0)
2   lw      $r2, 0($r1)
3   add     $r6, $r5, $r4
4   add     $r3, $r1, $r2
5   lw      $r4, 0($r6)
6   sub     $r2, $r0, $r4
7   addi    $r7, $r1, 4
8   add     $r4, $r1, $r3
9   sub     $r6, $r7, $r4
```

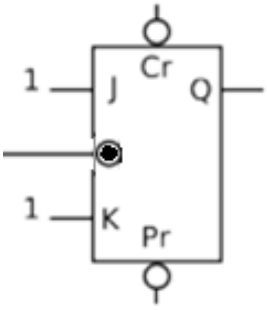
- *Dépendances WAW (Write After Write):*
  - *Après chaque écriture d'un registre, vérifier si le même registre est destination*

```
1   lw      $r1, 0($r0)
2   lw      $r2, 0($r1)
3   add     $r6, $r5, $r4
4   add     $r3, $r1, $r2
5   lw      $r4, 0($r6)
6   sub     $r2, $r0, $r4
7   addi    $r7, $r1, 4
8   add     $r4, $r1, $r3
9   sub     $r6, $r7, $r4
```

Exercice n°1 (4 pts) :

Soit les circuits de la figure suivante, complétez les chronogrammes associés :





Réponse

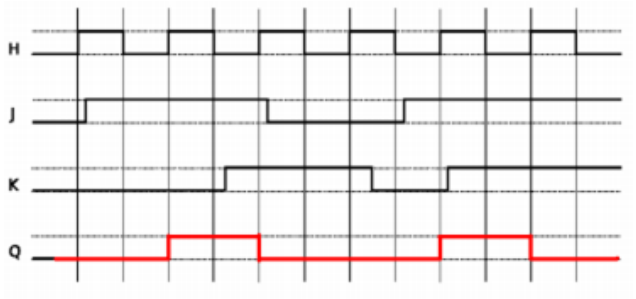
1- Analyse de la figure

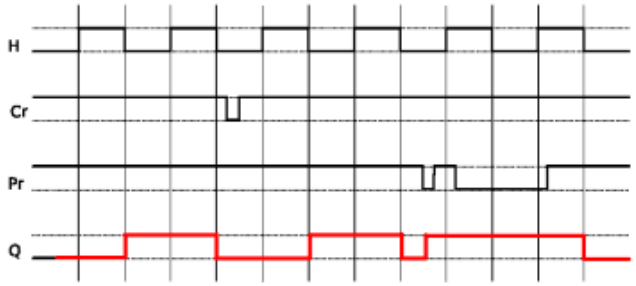
Bascule JK (synchrone) sur front Montant

Bascule JK (asynchrone) avec front descendant

Table de vérité de la bascule JK

	Pr	Cl	h	J	K		Q^+	
Mode Asynchrone	0	0	X	X	X		X	État interdit
	0	1	X	X	X		1	Remise à 1
	1	0	X	X	X		0	Remise à 0
Mode Synchrone	1	1	0/1	x	x		Q^-	Etat mémoire
	1	1	↓	0	0		Q^-	Etat mémoire
	1	1	↓	0	1		0	Remise à 0
	1	1	↓	1	0		1	Remise à 1
	1	1	↓	1	1		$\overline{Q}$	Basculement





Exercice n°4 (6 pts):

Soit un tableau de 10 entiers. On se propose de **ranger** les éléments de T dans un tableau R de façon à mettre les éléments positifs ou nuls de T au début de R suivis des éléments négatifs.

Ecrire un programme MIPS intitulé ''**Tab-Rang**'' qui permet de :

1. **Saisir le tableau T**
2. **Utiliser une fonction ''Rangement'' qui permet de ranger le tableau R**

Exemple :

T	5	7	-1	6	0	-15	9	4	-2	-18
R	5	7	6	0	9	4	-1	-15	-2	-18

N.B.

- Prévoir l'affichage des éléments du tableau rangé.
- N'oubliez pas de commenter le programme.

Réponse

1- Version Turbo Pascal

```
Program   Ranger_tab ;
Uses      Wincrt ;
Type      Tab = Array [1..10] of Integer ;
Var       T, R : Tab ;
           i, j : Integer ;
Begin
    Writeln ('Saisir les 10 éléments  de T') ;
    FOR   i:=1  To   10  Do
          Readln (T[i]) ;
   j:=0 ;
   FOR   i := 1   To  10   Do
      IF T[i] >= 0   Then   Begin
                              j := j+1 ;
                             R[j] := T[i] ;
                           End;
   FOR   i := 1   To  10  Do
      IF T[i] < 0   Then   Begin
                              j := j+1 ;
                             R[j] := T[i] ;
                           End;
   FOR   i := 1 To   10  Do  Write (T[i]:4) ;
   Writeln ;
   FOR   i := 1 To   10  Do  Write (R[i]:4) ;
End.
```

1- Version Mips

```
.data
Message: .asciiz "Entrez une valeur : "
Separateur: .asciiz ","
Tab_T:.word 0,0,0,0,0,0,0,0,0,0
Tab_R:.word 0,0,0,0,0,0,0,0,0,0
Card:.word 10
.text
main:
la $t1,Tab_T                    # initialisation
la $t2,Tab_R
lw $t0, Card
li $t3, 1

boucle1:                # saisir le tableau T
        bgt $t3, $t0, fonction
        li $v0, 4
        la $a0, Message
        syscall
        li $v0, 5
        syscall
        move $t4, $v0
        sw $t4, 0($t1)
        addi $t3, $t3, 1
        addi $t1, $t1, 4
        j boucle1

fonction:               # appel de fonction

jal rangement
jal affichage

li $v0, 10              # fin du programme
syscall

#########################################################

rangement:
la $t1,Tab_T                    # initialisation
li $t3, 1
bouclepos:
        bgt $t3, $t0, finbouclepos
        lw $t4, 0($t1)
        bgez $t4, positif
        addi $t3, $t3, 1
        addi $t1, $t1, 4
        j bouclepos

positif:
        sw $t4, 0($t2)
        addi $t3, $t3, 1
        addi $t1, $t1, 4
        addi $t2, $t2, 4
        j bouclepos
finbouclepos:

        la $t1,Tab_T                    # initialisation
        li $t3, 1
boucleneg:
        bgt $t3, $t0, finboucleneg
        lw $t4, 0($t1)
        bltz $t4, negatif
        addi $t3, $t3, 1
        addi $t1, $t1, 4
        j boucleneg

negatif:
        sw $t4, 0($t2)
        addi $t3, $t3, 1
```

```
        addi $t1, $t1, 4
        addi $t2, $t2, 4
        j boucleneg
finboucleneg: jr $ra

#############################################################

affichage:
la $t2,Tab_R                    # initialisation
li $t3, 1

boucle2:
        bgt $t3, $t0, finaffichage
        lw $a0, 0($t2)
        li,$v0, 1
        syscall
        li $v0, 4
        la $a0, Separateur
        syscall
        addi $t3, $t3, 1
        addi $t2, $t2, 4
        j boucle2

finaffichage: jr $ra
```

2- Exemple d'execution

Tableau T :

Entrez une valeur : 1
Entrez une valeur : -2
Entrez une valeur : 3
Entrez une valeur : -4
Entrez une valeur : 5
Entrez une valeur : -6
Entrez une valeur : 7
Entrez une valeur : -8
Entrez une valeur : 9
Entrez une valeur : 10

Tableau R :

1,3,5,7,9,10,-2,-4,-6,-8,

-- program is finished running --