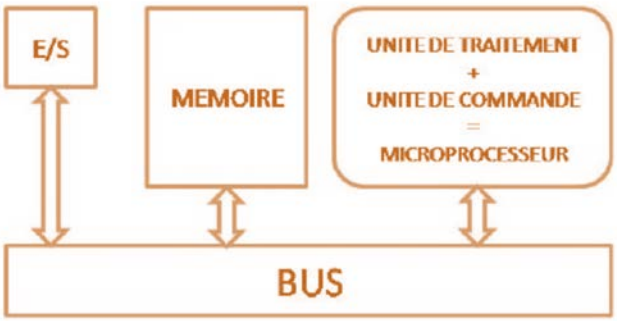
UNIVERSITE ABOU-BAKR BELKAID TLEMCEN
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE ELECTRIQUE ET ELECTRONIQUE
FILIERE DE GENIE INDUSTRIEL

CORRIGE EXAMEN DE MOYENNE DUREE

Architecture des ordinateurs L2 – S4
Année : 2014/2015

Questions de Cours : (05 points) 15' Pts

1. Quelle architecture est utilisée pour les ordinateurs actuels ?
   - **L'architecture de von Neumann.** **(1 pt)**
2. Représenter cette architecture sur un schéma en citant les différentes unités.



(1 pt)

3. Quel est le rôle de l'UAL dans l'unité centrale ?
   - **L'unité arithmétique est l'organe de l'ordinateur chargé d'effectuer les calculs arithmétiques et logiques sur 2 opérandes en entrée produisant un résultat en sortie.** **(1 pt)**
4. Quels sont les rôles du séquenceur et du décodeur ?
   - **Séquenceur : génère les signaux de commande, et met à jour le Compteur ordinal (CO).** **(0.5 pt)**
   - **Décodeur : décode le champ code opération pour chaque instruction et indique au séquenceur quelle opération (parmi celles du jeu d'instruction) doit être effectuée.** **(0.5 pt)**
5. Pour stocker une donnée dans la mémoire, le processeur envoie sur le bus de données la référence du mot-mémoire à sélectionner. (Vrai ou faux ? justifier.)
   - **Faux. C'est le bus d'adresse qui contient l'adresse du mot-mémoire à sélectionner.** **(1 pt)**

UNIVERSITE ABOU-BAKR BELKAID TLEMCEN
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE ELECTRIQUE ET ELECTRONIQUE
FILIERE DE GENIE INDUSTRIEL

Problème 1 : (7 points) 30' Pts

On considère une mémoire centrale de 4 Mo, où chaque octet est adressable séparément,

1. Quelles sont les tailles des bus (adresse/donnée) ?
   - Capacité = 4 Mo = $2^2 x 2^{20}$ octets = 2^{22} octets.
   - Taille du mot = 1 octet => **Taille du bus de données : 8 bits.** **(1 pt)**
   - Nombre mots = Capacité / Taille du mot = 2^{22} mots.
   - **Taille du bus d'adresse =** Log_2 (Nombre mots) = $Log_2(2^{22})$ = **22 bits.** **(1 pt)**
2. Calculer l'adresse, en octal du $5^{ème}$ élément d'un tableau dont l'adresse du premier élément est 65_8 et dont tous les éléments sont composés de 32 bits.
   - Adresse 1^{er} élément : $65_8 = 53_{10}$
   - Taille de chaque élément : 32 bits = 4 octets **(1 pt)**
   - **Adresse 5^{er} élément** : Adresse 1^{er} élément + 4 x 4 octets = $53_{10} + 16 = \mathbf{69_{10}} = \mathbf{105_8}$ **(1+1 pt)**
3. Calculer la taille de cette mémoire en l'exprimant en mots de 16 bits puis en mots de 32 bits.
   - Taille de la mémoire = 222 octets
   - Taille du mot = 16 bits = 2 octets
   - Nombre de mots = Taille mémoire / Taille du mot = $2^{22} / 2 = 2^{21}$ mots de 16 bits. **(1 pt)**
   - Taille du mot = 32 bits = 4 octets
   - Nombre de mots = Taille mémoire / Taille du mot = $2^{22} / 4 = 2^{20}$ mots de 32 bits. **(1 pt)**

UNIVERSITE ABOU-BAKR BELKAID TLEMCEN
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE ELECTRIQUE ET ELECTRONIQUE
FILIERE DE GENIE INDUSTRIEL

Problème 2 : (8 points) 40‘ Pts

Ecrire un programme en assembleur MIPS qui demande à l’utilisateur de saisir 2 entiers strictement positifs puis affiche leur moyenne. Le programme doit répéter la saisie tant que les nombres saisis ne sont pas positifs.

```
        .data
str1:   .asciiz "Entrer la valeur de a :"
str2:   .asciiz "Entrer la valeur de b :"
str3:   .asciiz "La moyenne de a et b :"
str4:   .asciiz "Erreur.Entrez une valeur positive :"
        .text
#################################### Saisie de a ##################################################
main :  addi $v0,$zero,4        # charger $v0 par 4 (affichage chaine)
        la $a0,str1             # charger $a0 par l'adresse de la chaine à afficher (entrer a)
        syscall                 # appel de la fonction pour l'affichage
loop_a: addi $v0,$zero,5        # charger $v0 par 5 (Saisie entier)
        syscall                 # appel de la fonction pour la lecture de a
        sle $t0,$v0,$zero       # test si a <= 0 mettre 1 dans $t0 sinon 0
        beq $t0,$zero,suite1    # si t0 = 0 continuer dans suite1
        addi $v0,$zero,4        # sinon charger $v0 par 4 (affichage chaine)
        la $a0,str4             # charger $a0 par l'adresse de la chaine à afficher (erreur)
        syscall                 # appel de la fonction pour l'affichage
        j loop_a                # recommencer la saisie de a
suite1: add $s0,$zero,$v0       # sauvegarder a dans le registre $s0

#################################### Saisie de b ##################################################
        addi $v0,$zero,4        # charger $v0 par 4 (affichage chaine)
        la $a0,str2             # charger $a0 par l'adresse de la chaine à afficher (entrer b)
        syscall                 # appel de la fonction pour l'affichage
loop_b: addi $v0,$zero,5        # charger $v0 par 5 (Saisie entier)
        syscall                 # appel de la fonction pour la lecture de b
        sle $t0,$v0,$zero       # test si b <= 0 mettre 1 dans $t0 sinon 0
        beq $t0,$zero,suite2    # si t0 = 0 continuer dans suite2
        addi $v0,$zero,4        # sinon charger $v0 par 4 (affichage chaine)
        la $a0,str4             # charger $a0 par l'adresse de la chaine à afficher (erreur)
        syscall                 # appel de la fonction pour l'affichage
        j loop_b                # recommencer la saisie de b
suite2: add $s1,$zero,$v0       # sauvegarder b dans le registre $s1

#################################### Calcul de la moyenne #########################################

        add $t1,$s0,$s1         # mettre dans $t1 la somme de a et b
        srl $s2,$t1,1           # décalge à droite par 1 bit du résultat (équivalent à une division par 2)

#################################### Affichage du résultat ########################################

        addi $v0,$zero,4        # charger $v0 par 4 (affichage chaine)
        la $a0,str3             # charger $a0 par l'adresse de la chaine à afficher (moyenne)
        syscall                 # appel de la fonction pour l'affichage
        addi $v0,$zero,1        # charger $v0 par 1 (affichage entier)
        add $a0,$zero,$s2       # charger $a0 par la valeur de $s2 qui contient la moyenne etière de a et b
        syscall                 # appel de la fonction pour l'affichage
```

(1 pt) **(1 pt)** **(1 pt)** **(1 pt)** **(1 pt)** **(1 pt)** **(1 pt)** **(1 pt)**