

## TD2 Architecture des ordinateurs

### Organisation Mémoire en MIPS

1. Dans l'architecture MIPS, la mémoire centrale est vue comme un tableau d'octets. Pour adresser un octet de la mémoire, on utilise un bus d'adresse de 32 bits.

- Donner en hexadécimale l'adresse du **premier** octet et du **dernier** octet.

**Réponse :** Premier octet  $0x00000000$

Dernier octet  $0xffffffff$

- Trouver la taille maximale de la mémoire en octets.

**Réponse :** Taille =  $2^{32}$  Octets = 4 Go

- Quel est le nombre de mots (.word) qu'on peut les stocker.

**Réponse :** nombre de mots =  $2^{30}$  mots (Sachant qu'un .word occupe 4 octets)

2. Rappelons qu'en MIPS, la partie des données (.data) commence par l'octet ayant l'adresse  $0x10010000$  et se termine par l'octet de l'adresse  $0x1003ffff$ .

- Donner en octets la taille de la partie .data

**Réponse :** Taille =  $(0x1003ffff - 0x10010000)_{16} = 196\ 608$  Octets = 192 Ko

- Quel est le nombre maximum d'entiers qu'on peut déclarer dans un program MIPS ?

**Réponse :** Nombre de mots =  $(196\ 608 \text{ Octets} / 4) = 48\ 152$  mots

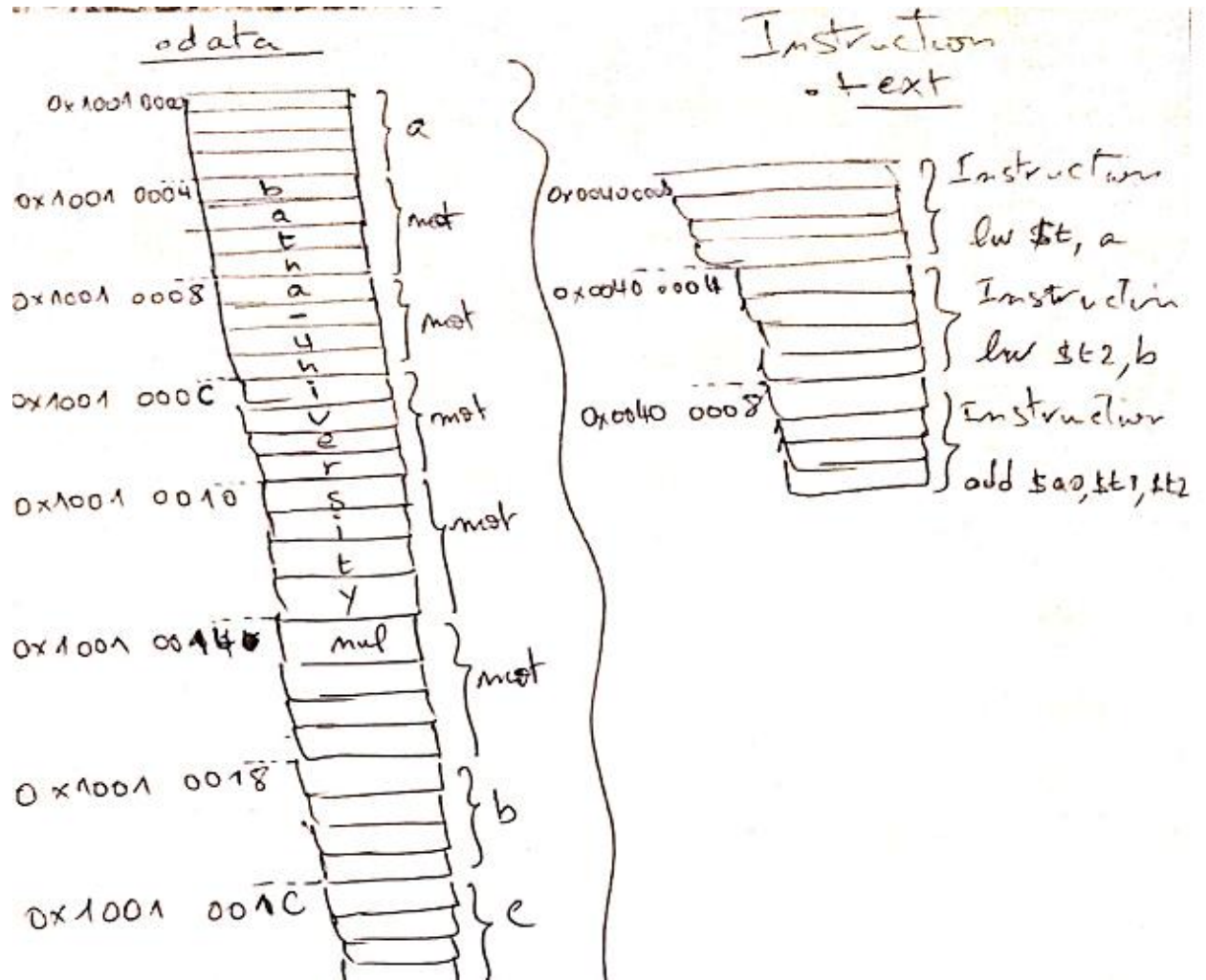
### Code MIPS

Soit le code MIPS suivant :

```
.data
    a: .word 5
    tableau: .asciiz "batna-university"
    b: .word 8
    c: .word
.text
    lw $16,a
    lw $8,b
    add $13,$16,$8
    sw $13,C
```

- Donner l'adresse de chaque unité d'information (data ou instruction) en supposant que chaque instruction est codée sur un seul mot.

Réponse :



- Pourquoi la chaîne `batna-university` occupe 5 mots pour stockage mémoire.

Chaque caractères nécessite un mot  
 $\frac{\text{batna-university}}{1 \quad 2 \quad 3 \quad 4} \quad | \quad \square \square \square$   
 le 5<sup>ème</sup> mot contient `mul` pour un byte et  
 les 3 derniers byte restent vides (ceci  
 est due à la nature du type asciiZ)

- Ecrire le program MIPS qui permet de : (1) déclarer 2 réels de type `.float` en définissant leurs valeurs initiales, (2) faire l'addition, (3) puis garder le résultat sur une 3<sup>ème</sup> variable.

```
.data
  a: .float -15.0
  b: .float 128.1
  c: .float
.text
  l.s $f1,a
  l.s $f2,b
  add.s $f3,$f1,$f2
  s.s $f3,c
```