

LICENCE INFORMATIQUE S3 - Architecture

Assembleur MIPS : TP 1

1 Simulateur : prise en main

Téléchargez depuis Arche le simulateur Java de programmes écrits en assembleur MIPS (**Mars4_4.jar**), ainsi que le tutoriel et le programme *Fibonacci_src.asm*.

- en suivant le tutoriel pas à pas, familiarisez-vous avec les principales fonctionnalités de Mars (pour lancer, tapez la commande suivante dans un terminal : **java -jar Mars4_4.jar**)
- comparez le programme assembleur fourni avec celui vu en TD, proposez une nouvelle version plus proche du TD (modifiez les lignes 27 et 28 de façon à ne plus faire des accès mémoire pour récupérer les deux termes précédents de la suite)
- calculez le terme d'ordre 30 de la suite de Fibonacci

2 Statistiques d'exécution

Cet exercice utilise l'outil "Instruction statistics" du simulateur *Mars* avec lequel il faudra se familiariser (il faut lier l'outil au programme assembleur en cours de simulation avec le bouton "Connect to MIPS").

Exécutez *column-major.asm* et *row-major.asm* et dans "Instruction statistics", observez les proportions des différents types d'instructions dans ces programmes.

3 Programmation, appels de fonctions

- Ecrivez un programme assembleur calculant la somme des entiers situés en mémoire entre les adresses *start* (inclus) et *end* (exclus). Vous pourrez compléter le début de programme suivant :

```
.data
    start: .word 17:20
    end:   .word 0
.text
    la $a0,start
    la $a1,end
```

- Transformez ce programme en une fonction **sum** ayant comme arguments a_0 (inclus) et a_1 (exclus) placés dans les registres **\$a0** et **\$a1**, le résultat étant placé dans **\$v0**.

(on fera dans la suite particulièrement attention à la phase de sauvegarde et de restauration des registres lors des appels de fonctions)

- Ecrivez une fonction **moy** qui calcule la moyenne des entiers situés en mémoire entre les adresses a_0 (inclus) et a_1 (exclus), cette fonction faisant appel à **sum**.