

TP N° 01

Objectifs :



Nous allons travailler sur une architecture MIPS 32 bits (Microprocessor without Interlocked Pipeline Stages). L'architecture MIPS est une architecture de processeur de type RISC (Reduced Instruction Set Computer). Son architecture est une référence. Les processeurs fabriqués selon cette architecture sont surtout utilisés dans les systèmes embarqués.



L'objectif n'est pas d'étudier cette architecture en particulier, ni d'étudier le jeu d'instruction de ce processeur. Néanmoins nous allons examiner le déroulement d'un logiciel à l'aide d'un simulateur d'architecture MIPS32. Ceci nous permettra de mieux comprendre l'organisation et l'exécution du code pour le microprocesseur.

Présentation du simulateur MARS :

MARS est un IDE (Integrated development environment) léger pour apprendre le langage assembleur MIPS Multiplateforme: MARS est écrit en langage JAVA, donc compatible avec tous les OS avec la machine virtuelle de java grâce au format .jar

Prise en main de l'environnement :

- Au démarrage il y a seulement trois options: New (créer un document en blanc)
Open (ouvrir un fichier assembleur existant)
Help (affiche l'aide)
- En ouvrant ou créant un fichier s'activent des fonctions classique d'édition et de manipulation de fichier plus la fonction "assemble" dans le menu "run"
- Quand on active "assemble" si il y a des erreurs dans le code elles sont indiquées dans la fenêtre "Mars Messages" en donnant les lignes et les explications relatives aux erreurs. Si le code est écrit sans erreurs. L'onglet "Edit" disparaît et est remplacé par l'onglet "execute" qui montre le suivi des données en mémoire et le segment en action du code. On a alors à sa disposition d'autres fonctions: exécuter en entier, exécuter une seule instruction et redémarrer.
- L'icône  permet de lancer l'exécution du programme jusqu'à la fin. En utilisant cette fonction vous observez le surlignement jaune qui montre l'adresse du programme en cours d'exécution. Les valeurs du programme qui sont calculées apparaissent dans le segment de donnée.
- L'icône  permet de faire un reset du programme et des valeurs de l'ensemble des registres du simulateur.

- L'icône  permet d'exécuter une instruction après l'autre. Autrement dit, de faire de l'exécution pas à pas. Cette fonction est complétée par l'icône  qui permet de revenir d'un pas : "single-step backwards".
- Le processeur utilise un adressage par octet. Vous pouvez changer le format de visualisation du contenu des adresses et des adresses en hexadécimal ou en décimal :

Hexadecimal Addresses Hexadecimal Values



- Dans le logiciel MARS, ajuster le « run speed » : afin d'avoir le temps de voir l'évolution de l'algorithme en fonction du temps.
- Depuis l'aide, (**Help>Help>Basic_Instructions**) vous trouverez l'ensemble des instructions disponibles dans ce microprocesseur.

Premier programme :

1. Dans le logiciel MARS créer un nouveau fichier et recopier le programme assembleur ci-dessous

```

.data
vars: .word 5
.word 10
.text
la $t0, vars
lw $t1, 0($t0)
lw $t2, 4($t0)
saut: bge $t1, $t2, exit
move $a0, $t1
li $v0, 1
syscall
addi $t1, $t1, 1
j saut
exit: li $v0, 10
syscall

```

2. Déterminer ce que fait le programme en utilisant l'aide (**Help>Help**) pour expliquer le rôle de chaque ligne du code.