

Circuits et architecture

TD 10 (TP 4) : Utilisation de tkgate et du simulateur
d'Ashley Wise

Ce TP est une suite du TP3; son but est de continuer à utiliser le circuit tkgate qui implante une partie des instructions du LC3 et aussi de découvrir le simulateur d'Ashley Wise, un outil qui implante le processeur LC3.

Le simulateur d'Ashley Wise est disponible à la page <http://users.crhc.illinois.edu/awise/>. La documentation se trouve sur <http://users.crhc.illinois.edu/awise/AshIDEUsageGuide.htm> (suivre les liens depuis la page web du td). Il est installé sur nivose et on le lance d'une fenêtre shell par `AshIDE -lc3`. Au lancement, sélectionnez `Project/New File` ou `Project/Open File` et répondez non à la question sur le système.

Lorsque le programme est écrit, vous l'assemblez avec `Assemble` et le simulez avec `Simulate`.

Pendant la simulation, on peut voir le contenu des registres avec le menu `View/Registers`.

On peut aussi voir le contenu de la mémoire avec `View/Data Values`. Pour avoir le contenu de la mémoire en hexadécimal, il faut sélectionner `DATA 2` dans le menu `Data Type`. Ainsi, vous pouvez récupérer votre programme en hexadécimal en indiquant "à gauche" de la fenêtre `Data Values` l'adresse mémoire de sa première instruction. (il faudra tout de même faire un peu d'édition : essayez `cut -c13-17` sur le fichier contenant la zone de `Data Values` adaptée.) Attention à écrire `x3000` pour une adresse en hexadécimal. Pour aujourd'hui, on peut aussi se contenter de charger les programmes à l'adresse 0.

Exercice 1

Récupérez le programme de la multiplication naïve non signée (vu en cours), `MultiNaive.lc3`, disponible à l'adresse <http://www.liafa.jussieu.fr/~amicheli/Ens/Archi/>. Testez-le sur `AshIDE`, transformez-le en une suite de valeurs hexadécimales grâce à la fenêtre `Data Values`, puis utilisez `Tkgate` pour le simuler.

À partir de maintenant, on peut utiliser toutes les instructions du LC3. Les programmes écrits ne pourront plus être utilisés sur `tkgate` tant que vous n'avez pas codé ces instructions. Notez que `AshIDE` initialise les registres à

0, alors que dans `tkgate` il faudra impérativement écrire les instructions d'initialisations.

Exercice 2

Écrivez sous `AshIDE` un programme qui calcule les sommes partielles d'une suite de 10 entiers écrits en mémoire (à la fin du programme), en écrasant cette même suite en mémoire. Par exemple, (5,2,10,3,6,120,120,10,0,1) est remplacée par (5,7,17,20,26,146,266,276,276,277).

Exercice 3 [sous-routines]

On veut faire un programme qui retourne une chaîne de caractères en place. Pour cela, on utilisera la sous-routine de calcul de la longueur d'une chaîne donnée dans le cours et une sous-routine qui échange le contenu de 2 adresses mémoire. Pour vérifier le résultat, on sélectionnera : `View/Memory bytes` dans le simulateur qui permet de visualiser les caractères.

Exercice 4 [pile]

Écrivez un programme qui calcule la factorielle d'un nombre de manière récursive. Le nombre sera lu en mémoire et le résultat écrit en mémoire à la fin. On pourra utiliser la multiplication naïve comme sous-routine.