

ARCHITECTURE ET SYSTÈME: TP 1

Licence Info 2 - Stefano Guerrini et Flavien Breuvar, Lê Thành Dũng Nguyễn et Rado Rakotonarivo

A.A. 2018–19 - 11/09/2018

Exercice 1. Vérifier si le programme assembleur `lc3as`, le simulateur `lc3sim` et l'interface graphique `lc3sim-tk` sont installés sur la machine. Sinon, procéder avec une installation locale dans un répertoire personnel. Les programmes sont téléchargeables à l'URL :

http://highered.mheducation.com/sites/0072467509/student_view0/lc-3_simulator.html

Ouvrir l'interface graphique du programme assembleur et observer la fenêtre principale.

La partie haute montre les registres généraux et les registres PC, IR et PSR. Sur la droite, en bas, un rectangle CC montre les valeurs des trois bits N, Z et P ; ce rectangle affiche ZERO si Z est à 1, NEGATIVE si N est à 1 et POSITIVE si P est à 1. Sur la gauche, en bas, on peut voir un rectangle qui affiche l'adresse d'un registre courant et un rectangle avec sa valeur en hexadécimal : le registre courant est marqué par une barre jaune dans la fenêtre de la mémoire.

Au centre de la fenêtre on peut voir (en jaune sur écran bleu) le contenu de la mémoire. Chaque ligne correspond à un registre dont l'interface affiche son adresse, sa valeur en hexadécimal et l'interprétation (le code assembleur) correspondant aux contenu du registre. Certaines registres ont une étiquette, qui correspond à celle associé à l'instruction assembleur correspondant.

Si le registre à été rempli avec une valeur en utilisant la directive `.FILL`, la fenêtre affiche cette directive et la valeur associée.

Exercice 2. Ouvrir un éditeur de texte pour l'écriture de programmes et copier le code suivant dans un fichier, par exemple, nommé `exemple.asm`.

```
.ORIG x3000
    LD      R0, VAL1
    ADD     R0, R0, R0
    LD      R7, VAL2
    ADD     R7, R7, R7
    ADD     R2, R0, R7
    HALT
VAL1      .FILL 1
          .BLKW 3
VAL2      .FILL -1
.END
```

Transformer le code assembleur contenu dans le fichier `exemple.asm` en code machine avec le programme assembleur, avec la commande

```
lc3as exemple.asm
```

Le code machine obtenu sera mémorisé dans le fichier `exemple.obj`. En utilisant le bouton "browse" du simulateur, charger le fichier du code machine `exemple.obj` dans le simulateur et l'exécuter pas à pas.

Pendant l'exécution, remarquer les valeur contenues dans les registres des instructions et leur évolution. Notamment:

1. Quelle est la valeur contenue dans les 5 bits les plus faibles du code hexadécimal de l'instruction LD R0, VAL1? Comparer cette valeur avec la différence entre l'adresse du registre avec étiquette VAL1 et l'adresse de cette instruction. Qu'est-ce qu'on peut en conclure ? Pourquoi cette petite différence entre les valeurs ?
2. Observer les différences entre les codes binaires des trois instructions ADD du programme. Qu'est-ce qu'on peut en déduire ?
3. Les registres VAL1 et VAL2 ne sont pas consécutifs. Pourquoi ? Quelle est la distance entre eux ?
4. Remarquer l'évolution de la valeur de CC durant l'exécution pas à pas du programme.
5. Remarquer les opérations d'addition qui doublent les valeurs des registres R0 et R7. Quel est leur effet au niveau de la représentation binaire des valeurs de ces registres ?

Exercice 3. Écrire, transformer en code machine et exécuter avec le simulateur les programmes assembleur vu en TD.

Exercice 4. Réfléchir à comment réaliser une boucle “while” en LC-3. Par exemple, écrire une simple boucle qui itère N fois (la valeur contenue dans le registre N) l'opération que ajoute 1 au registre R1.