

Circuits et Architecture (CA7)

TD n° 7 : Découverte du mini-processeur LC-3

Le but de ce TP est de découvrir le circuit logisim `LC-3-v0.circ` qui implante une partie des instructions du LC3 et que vous aurez à modifier pour le projet. Les différents fichiers mentionnés ci-dessous sont sur la page web du cours.

<https://www.irif.fr/~carton/Enseignement/Architecture/>

L'unité arithmétique et logique

Récupérez le fichier `ALU.circ` et ouvrez-le avec `logisim`. Le module ALU contient une petite partie du circuit du LC-3, l'unité arithmétique et logique. Le module `ALU_test` permet de mettre en situation ce module ALU.

Notez l'usage d'un multiplexeur dans l'ALU, qui sélectionne parmi quatre entrées (dont une actuellement non affectée) en fonction d'un fil de contrôle sur 2 bits.

Exercice 1

1. Expérimentez différentes valeurs pour e_1 et e_2 .
2. Quelles valeurs donner à e_1 et e_2 pour obtenir en sortie le ET bit à bit de *Input1* et *Input2*? Pour obtenir l'addition de *Input1* et *Cst*? Pour obtenir le NON bit à bit de *Input1*?

Exercice 2

Reprenez le codage du jeu d'instructions du LC-3 expliqué en cours. Les opérandes des instructions arithmétiques et logiques ADD et AND sont soit deux registres, soit un registre et un immédiat.

1. Comment sont différenciés les deux cas?
2. Quels bits différencient les opcodes des trois instructions arithmétiques et logiques?
3. Déduisez-en à quoi devront être branché e_1 et les deux bits de e_2 dans le circuit complet.

Exercice 3

1. Construisez la table de Karnaugh de toutes les instructions du LC-3 en fonction des 4 bits de l'opcode. Que pouvez-vous observer? Cette table sera utile ultérieurement, conservez-la.
2. Comment caractériser les instructions qui peuvent modifier des registres généraux?

L’horloge

Exercice 4

Ouvrez maintenant le fichier `BiClock.circ` et lancez sa simulation.

1. Quel est le comportement de ce circuit ?
2. Quel va être l’intérêt de ce circuit pour la réalisation de notre circuit implémentant LC-3 ?

Le circuit LC-3

Le fichier `LC-3-v0.circ` est une réalisation partielle du LC-3, dans laquelle seules les instructions `ADD`, `AND`, `NOT` sont “câblées”, même si beaucoup d’infrastructure est déjà en place pour plus tard. Nous avons utilisé une méthode assez différente de celle présentée en cours, afin de simplifier le circuit. En particulier il n’y a pas de bus unique, ni de registre d’adresse mémoire.

Mode d’emploi. Pour simuler l’exécution de programmes LC-3 avec ce circuit :

- Entrez dans la RAM les octets d’un programme LC-3, d’une des trois manières suivantes.
 - En mode *poke* (curseur main, ou `Ctrl-1`), on peut directement cliquer sur une case de la RAM et la remplir au clavier.
 - Dans le menu obtenu par clic droit sur la RAM, on peut choisir “*Edit Contents*”, ce qui ouvre un petit éditeur hexadécimal intégré à `logisim`.
 - Ce même menu propose aussi “*Load Image*”, permettant de charger dans la RAM le contenu d’un fichier mémoire `.mem` (voir ci-dessous).
- Lancez la simulation, soit via des tics manuels d’horloge (`Ctrl-T`), soit avec une horloge automatique (`Ctrl-K`).
- Si vous souhaitez relancer l’exécution du programme, mettez l’entrée *Reset* à 1, puis de nouveau à 0.

Cycles d’horloge. Toutes les instructions sont exécutées en 2 cycles de l’horloge utilisée dans le module `BiClock`.

- Pendant le premier cycle, qui commence quand `Fetch` passe à 1, l’instruction courante est lue en mémoire et stockée dans le registre d’instruction `IR`.
- Pendant le deuxième cycle, qui commence quand `Exec` passe à 1, cette instruction est exécutée et le compteur de programme est mis à jour (incrémenté ou autre).

Fichier mémoire. Un fichier mémoire chargeable dans une RAM `logisim` est un simple fichier texte dont la première ligne est `v2.0 raw`. Viennent ensuite les différents octets de la mémoire en hexadécimal. Voir la section “**Memory components**” de la documentation `logisim` pour plus de détails. Voici par exemple le contenu du fichier `AddSimple.mem` :

```
v2.0 raw
5020 1025 5260 1266 1440
```

Ce programme correspond au code assembleur contenu dans le fichier `AddSimple.asm`, disponible sur lui aussi la page du cours.

Exercice 5

Ouvrez le circuit `LC-3-v0.circ`, puis récupérez le programme `AddSimple.mem`. Chargez-le dans la RAM et lancez sa simulation. Observez en particulier le fonctionnement du module Registres.

L'exercice suivant vous demande de consulter l'appendice C du livre *Introduction to Computing Systems* de Patt et Patel. Celle-ci est disponible en ligne.

http://highered.mheducation.com/sites/0072467509/student_view0/appendices_a__b__c__d____e.html

Exercice 6 – Diagramme de fonctionnement

Le circuit LC-3 est décrit dans la figure C.3 de l'annexe C. L'exécution de chaque instruction par ce circuit est décrite dans le diagramme C.2 du même livre. En s'inspirant de cette présentation, donnez le diagramme de fonctionnement de l'instruction ADD dans le circuit `LC-3-v0.circ`. Étiquetez chaque état par le cycle d'horloge dans lequel il s'exécute.