

Projet “Circuits et architecture”

À rendre : le 10 décembre 2018 et le 10 Janvier 2019 à 18h

1 Présentation

Ce projet a un double but :

Câbler dans logisim un micro-processeur qui implante un sous-ensemble des instructions du micro-processeur LC-3 et

Programmer avec ces instructions quelques fonctions d’une bibliothèque pour la manipulation des chaînes de caractères.

Le point de départ est le circuit logisim LC-3-v0.circ, que vous trouverez sur l’espace Moodle du cours [1]. Il est **essentiel de modifier précisément ce fichier** pour réaliser votre projet.

Le projet comporte deux niveaux : le niveau 1 est le minimum requis et le niveau 2 est le projet avancé.

1.1 Instructions LC-3 à implanter

Le circuit principal main utilise plusieurs sous-circuits, dont les plus importants sont : ALU, BiClock, DecodeIR, GetAddr, NZP, RamCtrl, RegPC, WriteVal. Les sous-circuits coloriés en noir dans main sont complètement câblés et **ne doivent pas être modifiés**. Le circuit main exécute correctement les instructions NOT, ADD et AND du LC-3.

Avant de commencer le câblage, vous devez comprendre le fonctionnement du circuit distribué.

Pour le niveau 1 du projet, vous devez modifier le câblage (et pas l’interface) des circuits coloriés en rouge afin d’implémenter les instructions LEA, LDR, LD, ST, STR, BR et JMP, ainsi que TRN décrite ci-dessous.

Pour le niveau 2, il s’agit de compléter les modules RegPC et WriteVal afin d’implémenter les instructions JSR et JSRR.

La sémantique de ces instructions est celle vue en cours [2], en particulier “Cours n° 6 : description du LC-3”. Vous pouvez consulter également le site internet du livre de Patt & Patel [3], en particulier les annexes A et C.

1.2 Instruction de rotation circulaire

Le but est d’étendre l’ensemble d’instructions du LC-3 avec une nouvelle instruction arithmétique TRN, qui effectue une permutation circulaire à gauche des 16 bits d’un registre par un nombre de positions donné par un autre registre ou une valeur immédiate. Pour cela, il faut modifier le module ALU uniquement.

Plus précisément, la rotation gauche consiste à permuter circulairement les bits d’une position vers la gauche. Le bit 15 prend la valeur du bit 14, le bit 14 celle du bit 13, ..., le bit 1 la valeur du bit 0 et le bit 0 la valeur du bit 15 avant la rotation. Une rotation gauche de k positions consiste à effectuer k rotations gauches.

Voici quelques questions qui vous aideront à câbler cette instruction :

- Écrire un circuit qui effectue **une** rotation gauche. Le circuit prend 16 bits en entrée et produit 16 bits de sortie.
- Écrire un circuit commandé par un bit qui effectue une rotation gauche si ce bit de commande vaut 1 et qui n'effectue aucune rotation sinon.
- Écrire un circuit commandé par un bit qui effectue une rotation gauche de 2 positions si ce bit de commande vaut 1 et qui n'effectue aucune rotation sinon.
- Combiner les deux circuits précédents pour obtenir un circuit ayant 2 bits de commande permettant d'effectuer une rotation de 0, 1, 2 ou 3 positions.
- Écrire un circuit permettant d'effectuer une rotation de 0 à 15 positions. Il faudra pour cela ajouter des circuits effectuant des rotations sur k bits, pour quelques k donnés.

Le “op-code” de cette instruction est celui disponible en LC-3, 1101. Comme toute instruction arithmétique, TRN doit modifier les indicateurs nzp. L'instruction supporte les deux modes d'adressage (registre ou valeur immédiate) pour indiquer le nombre de positions.

15	12	11	9	8	6	5	4	3	2	0	
1	1	0	1	DR	SR1	0	0	0	SR2		TRN DR, SR1, SR2
1	1	0	1	DR	SR1	1	Imm5				TRN DR, SR1, Imm5

La première ligne correspond à l'adressage registre : DR reçoit la valeur obtenue en effectuant une rotation circulaire gauche de la valeur du registre SR1 d'un nombre de positions donné par SR2. La deuxième ligne correspond à l'adressage immédiat, le nombre de positions est donné par Imm5.

1.3 Programmation en LC-3 étendu

Le circuit construit vous servira pour tester vos programmes (ou sous-routines) LC-3 ci-dessous. Il s'agit de manipulations sur les chaînes de caractères codées comme des séquences de caractères terminées par le caractère nul. On suppose (convention spécifique à LC-3) que les caractères sont codés sur 16 bits. Les programmes à coder ont un comportement similaire aux fonctions homonymes de la bibliothèque du langage C.

- index calcule en R0 l'adresse de la première apparition d'un caractère dont le code est donné en R2 pour une chaîne de caractères dont l'adresse de début est donnée en R1 ; la valeur de R0 est 0 si le caractère est absent.
- strcpy copie la chaîne source (y compris le caractère nul) dont l'adresse de début est donnée en R1 dans la chaîne destination dont l'adresse est en R2.
- strncpy copie la chaîne source dont l'adresse de début est donnée en R1 dans la chaîne destination dont l'adresse est en R2 sur un nombre de caractères donné par le registre R0.
- strtrn copie la chaîne source dont l'adresse de début est donnée en R1 dans la chaîne destination dont l'adresse est en R2 en effectuant une rotation vers la gauche du contenu de la chaîne (sans le caractère nul) d'un nombre de positions donné par le registre R0. Par exemple, la chaîne abcd\0 sur laquelle est appliquée une rotation gauche de 2 positions est cdab\0.

Pour la mise au point de vos programmes, nous vous recommandons d'utiliser les outils de simulation LC-3 `lc3tools`.

Pour le niveau 1 du projet, il faut coder ces fonctions comme autant de programmes indépendants.

Pour le niveau 2 du projet, il faut coder ces fonctions comme des sous-routines qui ne modifient pas d'autres registres que ceux du résultat. Vous pouvez soit utiliser une mémoire de sauvegarde par sous-routine ou implémenter une pile.

2 Paquetage

Le projet à rendre doit contenir :

Circuit : Le fichier logisim LC-3-v1.circ (respectivement LC-3-v2.circ) contenant le circuit complété pour le niveau 1 (respectivement 2) du projet. Vous devez écrire des programmes de test (de type .mem) pour chaque instruction. N'hésitez pas à ajouter des commentaires explicatifs dans le circuit.

Programmes : Les fichiers assembleur LC-3 (de type .asm) avec le code des programmes manipulant des chaînes de caractères. Dans le code, indiquez les données de test que vous avez utilisées.

Rapport : Vous devez rendre un rapport expliquant la démarche suivie et le niveau atteint (1 ou 2). Il faut énumérer rapidement les changements du circuit original et montrer qu'ils réalisent bien les fonctions souhaitées. Votre rapport doit aussi comporter le code des programmes de test utilisés et justifier qu'ils illustrent le fonctionnement du circuit. Le rapport ne doit pas dépasser 5 pages.

3 Modalités pratiques

Le projet s'effectue en binôme. Une pénalité de 30% sera appliquée à la note finale pour les groupes de 3 ou 1 étudiant(s). Les groupes de plus de 3 étudiants ne sont pas acceptés.

Le projet donnera lieu à une soutenance (par binôme) en janvier. La note tiendra compte aussi de la lisibilité du circuit, de la simplicité de la solution choisie et de la qualité du rapport.

3.1 Remise du projet

Le rendu du projet se fera en deux étapes, via l'espace Moodle du cours [1] ou bien, en cas de problème, par mail aux enseignants du TD :

Rendu intermédiaire, 10 décembre 2018 avant 18h : les diagrammes pour les instructions du niveau 1 du projet (NOT, ADD, AND, LEA, LDR, STR, LD, ST, BR, JMP, TRN), le circuit (fichier LC-3-v1.circ) avec l'instruction TRN câblée et les fichiers .asm contenant les programmes LC-3 pour index et strcmp. **Les binômes seront figés** par ce rendu.

Rendu final, 10 janvier 2019 avant 18h : le paquet complet avec les diagrammes, les circuits LC-3 (fichiers LC-3-v1.circ et LC-3-v2.circ), les programmes assembleur LC-3 (fonctions codées, les tests de ces fonctions et des circuits) et le rapport.

Références

[1] <https://moodlesupd.script.univ-paris-diderot.fr/course/view.php?id=9707>

[2] Oliver Carton, Notes de cours, <http://www.irif.fr/~carton/Enseignement/Architecture>

[3] Yale N. Patt, Sanjay J. Patel, Introduction to Computing Systems : From Bits and Gates to C and Beyond, McGraw-Hill, <http://highered.mcgraw-hill.com/sites/0072467509/>