

Projet d'architecture et circuits

1 Présentation

Le but de ce projet est de construire avec `tkgate` un processeur qui implante certaines instructions du LC-3. Le point de départ est le circuit `tkgate` nommé `LC-3.v` que nous avons construit et que vous pouvez trouver sur l'espace Didel du cours [1], ou sur le réseau de l'UFR dans `~1etouzey/archi`.

1.1 Instructions à implanter

Ce circuit implémente déjà les instructions `NOT`, `ADD`, `AND`, `BR`, `JMP`, `LDR`, `STR` du LC-3. Il reste à ajouter des composants au circuit pour qu'il implémente d'autres instructions :

- Minimum requis : "câbler" les instructions `SHT`, `LEA`, `LD`, `ST`, où `SHT` correspond à une rotation circulaire (circular shift) bit à bit décrite plus bas.
- Pour continuer : ajouter les instructions `JSR` et `JSRR`.
- Pour aller plus loin : `LDI`, `STI` et les entrées/sorties avec le module "terminal" de `tkgate` (`Tty`, abréviation 'T') : elles doivent être associées à des adresses mémoires.

1.2 La rotation circulaire

Le but est de construire un composant du circuit permettant de permuter circulairement les bits d'un registre 16 bits, où on suppose que les bits sont numérotés de 0 à 15 (le bit n° 15 est le plus significatif).

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 instructions :

- a) Écrire un circuit qui effectue une rotation gauche. Le circuit prend 16 bits en entrée et produit 16 bits de sortie.
- b) 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.
- c) 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.

- d) 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.
- e) 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.

Codage de l'instruction : l'opcode de SHT sera 1101, i.e. l'opcode qui ne correspond à aucune instruction dans le LC3. L'instruction SHT doit modifier nzp. Il supportera deux modes d'adressage calqués sur ceux de AND :

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

La première ligne correspond à l'adressage registre : DR reçoit SR1 auquel on a appliqué une rotation de SR2 positions. La deuxième ligne correspond à l'adressage immédiat : DR reçoit SR1 auquel on a appliqué une rotation de Imm5 positions.

1.3 Les documents de référence

- Les documents de référence pour la description du LC-3 sont :
- le cours [2], en particulier “Cours n° 6 : description du LC-3”).
 - les TD7 et ultérieurs, ainsi que le document `ISA.pdf` disponibles sur l'espace Didel du cours [1] (ou localement dans `~letouzey/archi`). Vous trouverez bientôt au même endroit des programmes d'exemples.
 - le site internet du livre de Patt & Patel [3], voir en particulier l'annexe A.

2 Paquetage

Le projet à rendre doit contenir :

2.1 Fichiers

Le fichier `tkgate` ainsi qu'un ou plusieurs programmes d'exemples (de type `proc.mem`). Des commentaires dans `tkgate` doivent nommer les différents composants ajoutés et des afficheurs doivent être prévus pour pouvoir tracer le déroulement du programme. Vous pouvez supprimer ou ajouter des commentaires du fichier `LC-3.v` initial et réorganiser ses composants comme bon vous semble (surtout pour le rendre plus lisible).

2.2 Mémoire

Vous devez rendre un mémoire expliquant la démarche suivie. Il faut énumérer rapidement les modifications et composants ajoutés au circuit et montrer qu'ils réalisent bien les fonctions souhaitées. Votre mémoire doit aussi comporter le

code du ou des programmes (désassemblés) et justifier qu'ils illustrent le fonctionnement du simulateur et l'intérêt des nouvelles instructions. Le mémoire ne devrait pas dépasser 5 pages.

3 Modalités pratiques

Le projet s'effectue seul ou en binôme. Il donnera lieu à une soutenance *individuelle* en janvier. La note tiendra compte de la lisibilité du circuit et de la simplicité de la solution choisie.

3.1 Remise du projet

Le rendu du projet se fera via l'espace Didel du cours (ou bien en cas de problème par mail : pierre.letouzey@inria.fr). Le projet devra être rendu vers la fin de la session d'examen de janvier, la date limite précise vous sera précisée ultérieurement.

Le programme et les fichiers d'exemples devront se trouver sur les machines de l'ufr le jour de la soutenance.

Références

- [1] <http://didel.script.univ-paris-diderot.fr/claroline/course/index.php?cid=M1ARCHI>
- [2] Oliver Carton : <http://www.liafa.jussieu.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/>