

TD2

RV32I architecture monocycle

Michel Agoyan

Janvier 2022

1 Introduction

Il vous est demandé de rédiger un compte rendu pour ce TD dans lequel vous ferez figurer toutes les remarques pertinentes qui vous semblent nécessaires à la bonne compréhension du problème posé et de la solution apportée.

Les fichiers de simulations **.circ** utilisés pour chaque TD seront également joints au compte rendu de TD.

Nous vous rappelons qu'il est important pour une bonne compréhension et bonne assimilation de ce cours de réaliser un travail personnel.

Nous allons, tout au long de ce TD, construire une microarchitecture monocycle. Comme il a été vu en cours, la microarchitecture est formée de deux blocs :

1. Le chemin de données ("data-path") où l'on connecte les divers composants constituant la microarchitecture (Compteur programme, mémoire instruction, banc de registre, unité arithmétique et logique) pour le traitement des données.
2. Le séquenceur ("control-path") qui va contrôler le data-path et changer sa configuration afin de traiter les données conformément à l'instruction en cours d'exécution

Pour ce cours et l'ensemble des TDs qui suivront, nous allons utiliser le logiciel de simulation numérique **logisim evolution** : <https://github.com/logisim-evolution/logisim-evolution>.

logisim evolution nécessite qu'un environnement Java soit installé sur votre PC : <https://www.oracle.com/java/technologies/downloads/>.

logisim evolution est également installé sur le serveur **tallinn.emse.fr** ; ci-après vous trouverez les lignes de commandes pour l'activer :

```
source /etc/profile.d/modules.sh
module load logisim/3.7.2
alias logisim='java -jar /apps/LOGISIM/logisim-evolution-3.7.2-all.jar'
```

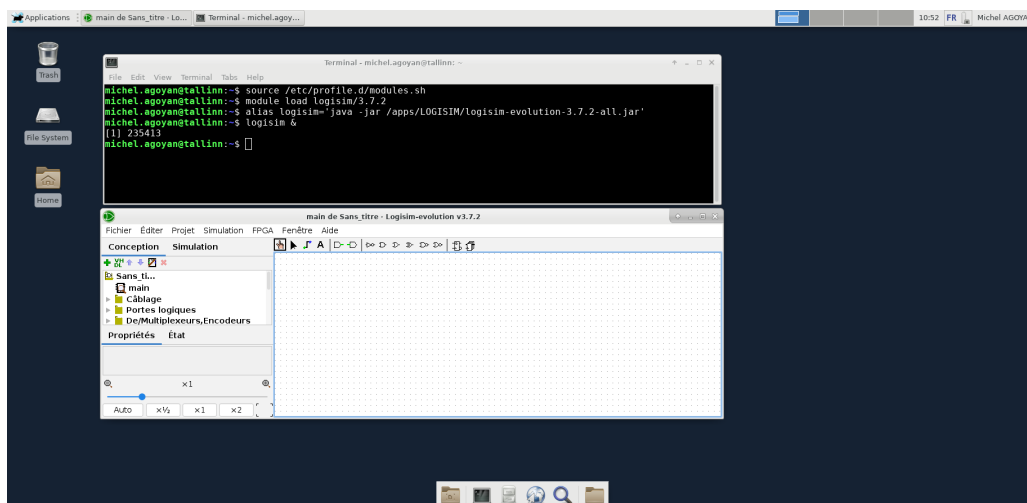


FIGURE 1 – logisim

Lors du premier lancement de l'application, choisissez le français comme langue pour l'interface. Menu *Fichier* .. *Préférences* .. *onglet international* .. *langue*

2 Intégration du banc de registres 32 bits

Le banc de registres 32 bits développé dans le cadre du TD1 que vous avez sauvé dans le circuit logisim : *registers_bank.circ* va être inclus dans un nouveau circuit logisim relatif à ce TD : *riscv_monocycle_incomplete.circ*. Pour ce faire, ouvrez le circuit *riscv_monocycle_incomplete.circ* et ajoutez *registers_bank.circ* en tant que librairie : menu **Projet, Charger une librairie, Librairie logisim-evolution ...**

Le banc de registres apparaît alors comme une librairie (Figure ??) et vous pouvez l'utiliser comme un composant standard.

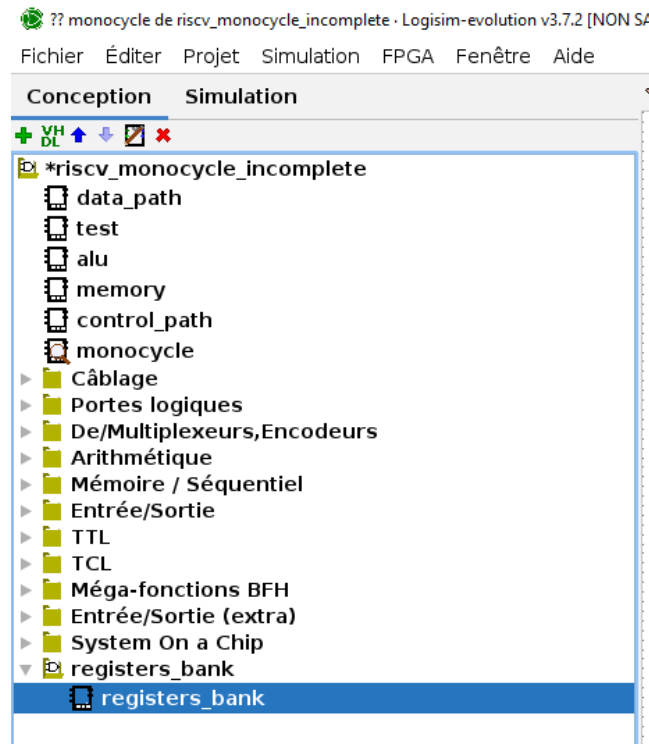


FIGURE 2 – Librairie : Banc de registres

Q1

Dans quel sous-circuit faut-il instancier le banc de registres ?

Placez le banc de registres dans le sous-circuit identifié et connectez-le aux autres composants de ce sous-circuit.

3 Ajout d'une instruction

Q2

Examinez les différents sous circuits et en déduire le type d'instruction actuellement supporté, argumentez votre réponse.

Q3

Quel(s) circuit(s) contient(contiennent) les instructions du programme à exécuter sur cette micro-architecture ?

Q4

en supposant que vous souhaitiez exécuter les deux instructions suivantes :

— **001080B3**

— **001080B3**

Quelles valeurs initiales de quels composants faut-il positionner(indication : décomposez l'instruction en octet) ?

Q5

Désassemblez l'instruction de la question 4, quel est le résultat attendu ?

Q6

Sélectionnez le circuit *monocycle* et placez vous en mode simulation (sélectionnez l'onglet *simulation*).

A l'aide de l'outil *Changez les valeurs dans le circuit* et en vous déplaçant dans la hiérarchie du circuit affichée dans l'onglet simulation :

— Initialisez les composants identifiés à la question 4 avec les deux instructions choisies

— Initialisez également le ou les registres impacté(s) par ces instructions à la valeur **1**.

Remplacez vous au niveau du circuit *monocycle* et Lancez la simulation (menu *simulation*, *simulation activée* ou *ctrl-E*).

Descendez dans la hiérarchie et vérifiez si l'état des sous-circuits est conforme à ce que vous attendez.

De nouveau, remplacez vous au niveau du circuit *monocycle* et à l'aide de l'outil *Changez les valeurs dans le circuit* faites passer l'entrée horloge du circuit *clk_i* à la valeur 1 puis zéro = 1 front d'horloge.

Déplacez vous de nouveau dans la hiérarchie pour vérifier si l'état des sous-circuits est conforme à ce que vous attendez

Réitérez la procédure jusqu'à ce que vous ayez exécuté les deux instructions.

Quel est le résultat final ?

Q7

Ajoutez à l'architecture actuelle le support de l'instruction **OR**

4 Ajout des types I et U

Pour pouvoir charger n'importe quels registres du RV32I avec une valeur d'initialisation sur 32bits, deux instructions sont nécessaires (voir cours diapo 15) :

- **LUI**
- **ADDI**

Q8

Modifiez le *datapath* et le *control path* de manière à pouvoir supporter les types d'instruction. Procédez progressivement, ajoutez le support d'un type d'instruction puis vérifiez le, puis ajoutez le support du deuxième type d'instruction

Q9

Assemblez à la main, le programme suivant : **LI R1,0xDEADBEEF**.

Q10

Chargez les instructions ainsi assemblées comme indiqué à la question 6 et Lancez la simulation pour vérifier votre implémentation.

5 Pour aller plus loin

Pour ceux qui restent sur leur faim, vous pouvez continuer l'exercice en implémentant le LW, SW.