

TD1

Organisation des registres

Michel Agoyan

Décembre 2021

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.

Ce TD, dans un premier temps, porte sur l'étude de l'architecture interne du banc de registres utilisé dans l'architecture RISC-V RV32I, dans un deuxième temps, nous connecterons ce banc de registres à l'unité arithmétique et logique, puis à l'aide d'un test nous decoderons nos premières instructions de type registre-registre.

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'
```

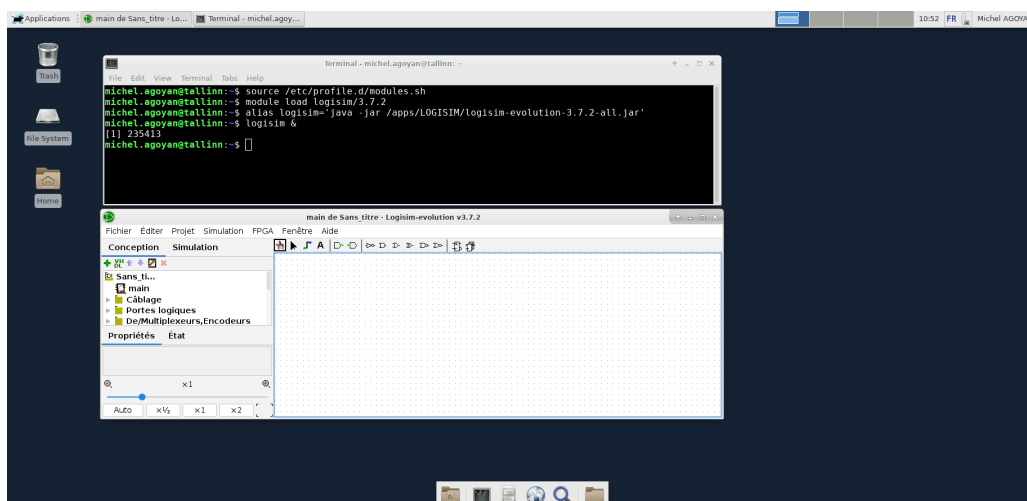


FIGURE 1 – logisim

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

2 Banc de registres : 4 x 8 bits

Dans un premier temps nous allons construire un mini banc de registres de manière à bien comprendre sa structure et son fonctionnement.

Dans ce premier exercice, nous allons donc réaliser un banc de registres formé de quatre registres huit bits. Ce banc de registres doit permettre d'adresser deux registres "sources" à la fois et de lire leur contenu respectif de manière asynchrone. Dans le même temps, il permet aussi d'adresser un registre "destination" et de pouvoir y écrire une donnée 8 bits de manière synchrone à l'horloge qui cadence le système.

Les ports d'entrées et de sorties pour ce banc de registres sont les suivants :

- entrées :
 - **rst_i** : entrée de mise à zéro active au niveau haut
 - **clk_i** : entrée horloge
 - **rs1_add_i** : adresse du registre source numéro 1
 - **rs2_add_i** : adresse du registre source numéro 2
 - **rd_add_i** : adresse du registre de destination
 - **rd_data_i** : donnée d'entrée à écrire dans le registre de destination
 - **we_i** : validation de l'écriture dans le registre de destination active à l'état haut, synchrone à *clk_i*
- sorties :
 - **rs1_data_o** : sortie du registre source 1
 - **rs2_data_o** : sortie du registre source 2

Q1

Quelle est la largeur en nombre de bits des entrées et des sorties : *rd_data_i*, *rs1_data_o*, *rs2_data_o* ?

Q2

Combien de bits sont nécessaires pour pouvoir adresser 4 registres ?

En déduire la largeur en nombre de bits des entrées *rd_add_i*, *rs1_add_i*, *rs2_add_i*.

Q3

Après avoir ouvert le circuit **td1.circ** dans logisim, ajoutez un nouveau circuit que vous nommerez **reg8_bank**.

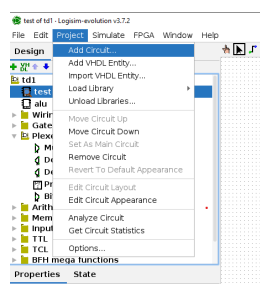


FIGURE 2 – logisim_add_circuit

Sélectionnez ce nouveau circuit et commencez à dessiner le schéma électrique correspondant à la description fonctionnelle faite plus haut.

Pour les ports d'entrées sorties utilisez le composant *Broche* en veillant à bien sélectionner la direction : entrée ou sortie ainsi que la largeur.

Veillez également à utiliser les mêmes noms que ceux donnés dans la section précédente. (Double clic sur le composant broche ou champ *Etiquette* dans la fenêtre *Propriétés*).

Les seuls autres composants actifs que vous aurez à utiliser sont :

- Registre dans la librairie *Mémoire/Séquentiel*
- Multiplexeur dans la librairie *De/Multiplexeurs,encodeurs*
- Démultiplexeur

3 Simu et vérif banc registres 8bits

Q4

Une fois le schéma électrique de votre banc de registres réalisé et sauvé, ajoutez une instance du banc de registres dans le circuit *test*, ajoutez de nouveau des broches d'entrées et sorties sur ce composant en veillant à leur donner la bonne direction, la bonne largeur et le bon nom.

Ajoutez des *sondes*, notamment sur les sortie *rs1_data_o* et *rs2_data_o* pour améliorer la visibilité de l'état du banc de registres pendant la mise au point.

A l'aide de l'outil *Changer les valeurs dans le circuit* et en vous plaçant en mode simulation, vérifiez votre schéma :

- remise à zéro
- écriture synchrone dans plusieurs registres
- lecture asynchrone de deux registres précédemment mis à jour lors d'une écriture

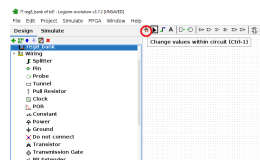


FIGURE 3 – logisim_change_value

Q5

Déssinez le chronogramme correspondant au scénario suivant :

- adresse du registre source 1 ,2 et destination à 0, valeur des données à inscrire dans le registre de destination à 0.
- remise à zero
- écriture de la valeur 0x11,0x22,0x33,0X44 respectivement dans le registre 0,1,2 et 3
- lecture des registres 1 et 2
- remise à zero
- lecture des registres 1 et 2

Q6

En vous aidant de l'outil *Changer les valeurs dans le circuit* essayer de reproduire la séquence de commandes de la question précédente.

Utilisez : *menu Simulation : Chronogramme ...* pour capturer le chronogramme de cette séquence de stimuli.

4 Banc de registres 32bits

Q7

Ouvrez le circuit **registers_bank_incomplete.circ** et complétez ce schéma en vous inspirant de ce que vous venez de faire dans l'exercice précédent.

Q8

Modifiez votre schéma pour que le banc de registres soit conforme à la spécification RISC-V : on ne peut pas écrire dans le registre 0 et la valeur retournée lors d'une lecture du registre 0 est toujours 0.

Indication :

- il existe un composant **Constante** dans la librairie *Câblage*.
- Si une broche d'un composant est non connectée on doit impérativement lui attacher le composant *ne pas connecter* symbolisé par une croix en forme de X rouge avec un point bleu en son centre.

Q9

Reliez le banc de registres 32bits à l'Alu dans le circuit **test**.

Comment doit-on connecter les entrées de l'ALU avec le banc de registres ?

Comment doit-on connecter la sortie de l'ALU au banc de registres ?

Q10

Créez une entrée de 32bits de largeur que l'on nommera **instruction**

Supposez que les seules instructions qui vont être présentées sur cette entrée soient des instructions de type **registre** et soient des **additions**.

Quelle valeur doit être appliquée sur l'entrée **func_i** de l'ALU ?

Utilisez le composant **Répartiteur (splitter)** pour extraire les champs de bits nécessaires pour piloter le reste des entrées de l'ALU..

Q11

Compilez (traduire en langage assembleur RV32I) le programme suivant :

```
Unsigned int a=6 ;  
Unsigned int b=2 ;  
Unsigned int c=3 ;  
Unsigned int d=1;  
Unsigned int e;
```

```
e=(a+b)+(c+d);
```

Listing 1 – programme simple

Assemblez le programme assembleur obtenu.

Q12

Initialisez les registres utilisés avec les valeurs adéquates à l'aide de l'outil *Changer les valeurs dans le circuit*, en vous plaçant en mode simulation.

Appliquez à l'aide de l'outil *Changer les valeurs dans le circuit* les instructions que vous venez d'assembler à la question qui précède sur l'entrée **instruction** du circuit **test** et vérifiez que le résultat final est correct.

Q13

Créez un nouveau circuit : *Fichier ...Nouveau*.

Copiez l'ensemble du banc de registres 32 bits que vous venez de compléter et de vérifier dans ce circuit.

Changez le nom du circuit : Champ *Nom du circuit* dans la fenêtre *Propriétés*, remplacez *main* par *registers_bank*.

Ce circuit servira de librairie pour le prochain TD . Il est donc impératif de terminer ce TD de manière à pouvoir commencer le suivant.