

# TD3

## RV32I architecture pipeline

Michel Agoyan

Janvier 2023

### 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 une bonne assimilation de ce cours de réaliser un travail personnel.

Dans ce troisième TD, nous allons expérimenter les effets de la dépendance de donnée des instructions sur microarchitecture de type **pipeline**.

Nous essaierons deux méthodes simples pour tenter de résoudre cette dépendance :

- insérer des instructions NOP
- implémenter un interlock dans sa version la plus simple

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'
```

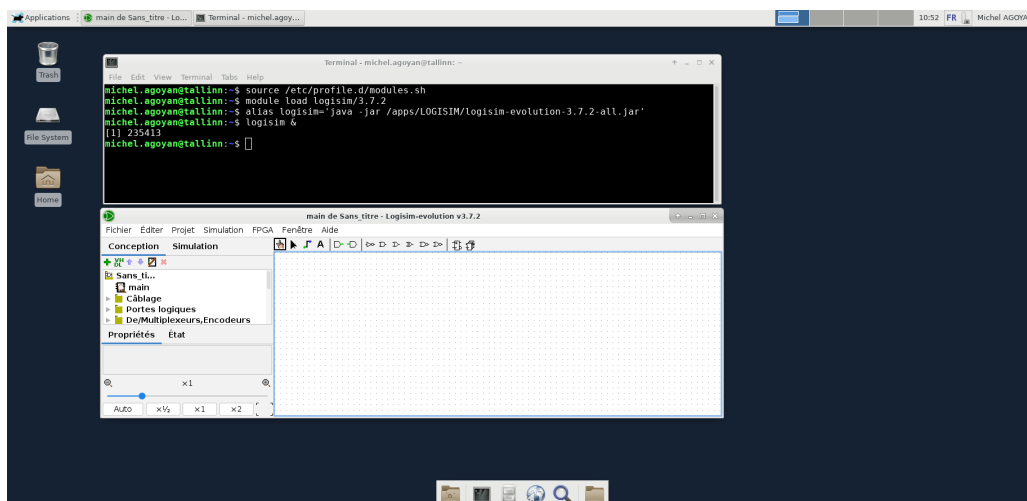


FIGURE 1 – logisim

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

## 2 Ajout de bibliothèques

### Intégration du banc de registres 32 bits

Le banc de registres 32 bits développé dans le cadre du TD1 que vous avez sauvegardé dans le circuit logisim : *registers\_bank.circ* va être inclus dans un nouveau circuit logisim relatif à ce TD : *riscv\_5\_stages\_incomplete.circ*. Pour ce faire, ouvrez le circuit *riscv\_5\_stages\_incomplete.circ* et ajoutez *registers\_bank.circ* en tant que bibliothèque : menu **Projet, Charger une bibliothèque, Bibliothèque logisim-evolution ...**

Le banc de registres apparaît alors comme une bibliothèque (Figure 2) et vous pouvez l'utiliser comme un composant standard.

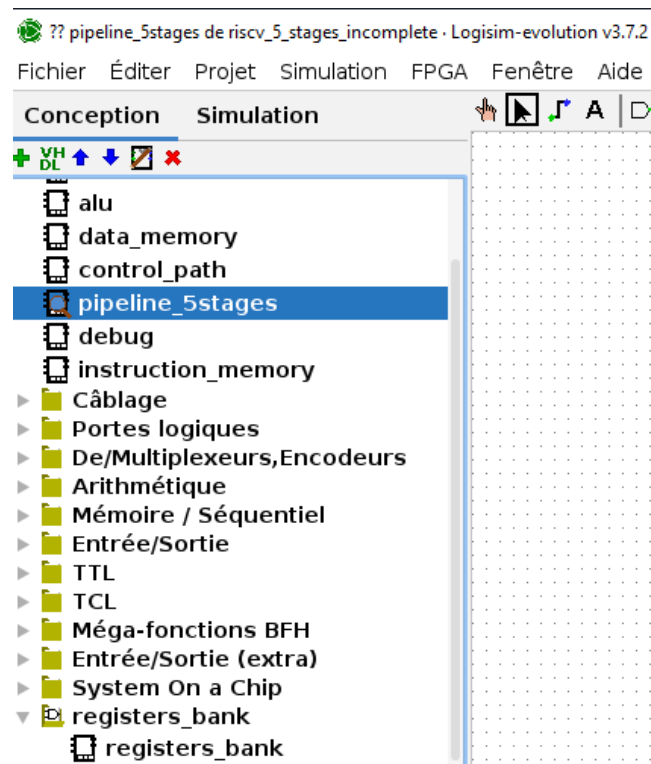


FIGURE 2 – Bibliothèque : Banc de registres

### Intégration du générateur de valeur immédiate

Sélectionnez tous les éléments qui forment le générateur de valeur immédiate mis au point lors du TD2 et copiez les dans le presse papier. Depuis le menu **fichier, Nouveau** créez un nouveau circuit **imm\_gen** dont vous nommerez le fichier : **immediate\_generator.circ**.

Ce circuit a 2 ports d'entrée :

- imm\_gen\_data\_i (32bits)
- imm\_gen\_sel\_i (1bit)

et un port de sortie :

- imm\_gen\_data\_o (32bits)

Importez ce nouveau composant en tant que bibliothèque : menu **Projet, Charger une bibliothèque, Bibliothèque logisim-evolution ...**

Compléter le **data\_path** à l'aide des deux composants que l'on vient d'importer.

### 3 Étude rapide du pipeline

Examinez le sous circuit **data\_path**

#### Q1

Quel est le rôle du signal **stall\_i** ?

Examinez le sous circuit **control\_path**

#### Q2

Que représentent les signaux :

- **rs1\_add\_w**
- **rs2\_add\_w**

#### Q3

Que représente le signal **rd\_add\_w** ?

#### Q4

Comment est actuellement généré le signal **stall\_w** ?

#### Q5

Que se passe-t-il quand le signal **stall\_w** est actif ?

#### Q6

En supposant que vous génériez le signal **stall\_w** à l'aide des signaux **rs1\_add\_w**, **rs2\_add\_w** et **rd\_add\_w**, quel type de dépendance pourrait gérer le pipeline ?

## 4 Exécution d'un programme

### Q7

Assemblez le programme suivant :

```
addi r5,r0,#0
addi r6,r0,#0
addi r7,r0,#0
ori r5,r5,2
ori r6,r6,3
ori r7,r7,4
add r28,r5,r5
add r29,r6,r6
add r30,r7,r7
addi r0,r0,#0
```

Une fois le programme assemblé, placez les valeurs des instructions en hexadécimal dans le fichier **rom.hex**

### Q8

Quel est le résultat normalement attendu dans les registres **r28**, **r29**, **r30** ?

### Q9

De manière à pouvoir ajouter facilement les registres au chronogramme, nommez si ce n'est pas déjà fait : les registres 5,6,7,28,29,30 respectivement R5,R6,R7,R28,R29,R30.

Placez-vous sur l'onglet simulation et sélectionnez l'outil **Changer les valeurs dans le circuit** puis déplacez vous dans la hiérarchie sur le sous circuit **instruction\_memory**, un clic droit de la souris vous permet de charger la mémoire à l'aide de **Charger l'image...**

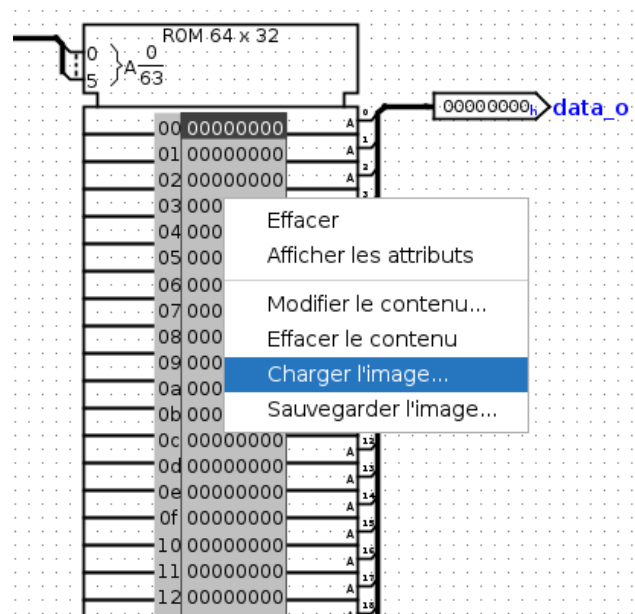


FIGURE 3 – Charger l'image

Avant de lancer la simulation, placez vous en haut de la hiérarchie sur le circuit **test** puis créez un chronogramme : menu **Simulation, Chronogramme...**

Sélectionnez l'onglet **Chronogramme** puis à l'aide du bouton **Ajouter ou Supprimer des signaux**, ajoutez les signaux suivants :

- Dans le sous circuit **data\_path**
  - la sonde :**alu\_o**
  - la sonde :**alu\_op1**
  - la sonde :**alu\_op2**
  - la sonde :**rd\_data\_w**
  - le port :**reg\_write\_i**
  - la sonde :**pc\_w**
- Dans le sous sous circuit **registers\_bank**
  - le registre :**R5**
  - le registre :**R6**
  - le registre :**R7**
  - le registre :**R28**
  - le registre :**R29**
  - le registre :**R30**
  - le registre :**R31**
- Dans le sous circuit **control\_path**
  - la sonde :**IFETCH**
  - la sonde :**IDEC**
  - la sonde :**EXE**
  - la sonde :**MEM**
  - le port :**WB**
  - la sonde :**rs1\_add\_w**
  - la sonde :**rs2\_add\_w**
  - la sonde :**rsd\_add\_w**
  - la sonde :**stall\_w**

Sélectionnez les signaux d'une largeur supérieure à 1 et changez le format d'affichage : passez en hexadécimal. Vous pouvez activer le zoom sur les chronogrammes en utilisant la touche **ctrl** et la molette centrale de la souris.

## Q10

Lancez la simulation en activant les **Tic** horloge.  
Quel est le résultat obtenu ? Pourquoi ?

## 5 correction du problème

### correction logicielle

#### Q11

Comment peut-on coder une instruction **NOP** sur l'architecture RISC-V ?

#### Q12

Utilisez cette instruction pour éliminer les dépendances. De quel type de dépendance s'agit-il ? Combien d'instructions NOP devez-vous utiliser ?

#### Q13

Vérifiez, en utilisant la simulation, que le programme fonctionne comme attendu.

### correction matérielle

#### Q14

Lorsque vous avez répondu à la question **Q13** quels étaient les étages du pipeline concernés par les dépendances que vous avez trouvées ?

#### Q15

Implémentez un interlock dans sa version la plus simple (qui ne prend en compte que les étages du pipeline concernés par les dépendances que vous avez trouvées).

#### Q16

Testez votre solution en utilisant le programme d'origine sans les instructions NOP insérées et comparez la séquence d'instructions sur la sonde **WB** avec celle issue du programme de la question **Q12**