

Laboratoire d'architecture des ordinateurs semestre printemps 2022

Microarchitecture - Combinatoire - Instructions

Informations générales

Cette partie est à faire de manière **individuelle**. A l'issue de ce laboratoire, nous considérons que chaque étudiant possède tous les outils nécessaires à la réalisation de tous les laboratoires de ce cours.



N'oubliez pas de sauvegarder et d'archiver votre projet à chaque séance de laboratoire

NOTE : Nous vous rappelons que si vous utilisez les machines de laboratoire situées au niveau A, il ne faut pas considérer les données qui sont dessus comme sauvegardées. Si les machines ont un problème nous les remettons dans leur état d'origine et toutes les données présentes sont effacées.


Objectifs du laboratoire

L'objectif de cette partie du laboratoire sera, dans un premier temps, d'installer tous les outils nécessaires pour la réalisation des différentes étapes à réaliser dans le cadre du laboratoire. Nous allons créer un processeur complet simplifié. Le but étant que vous réalisiez chaque étape vous-mêmes afin de pouvoir mettre en pratique la théorie vue en cours.

Ce laboratoire n'est pas noté mais sert d'introduction à la méthodologie et à la prise en main des outils dont nous aurons besoin tout au long du semestre.

Outils


Pour ce laboratoire, vous devez utiliser les outils disponibles sur les machines de laboratoire (A07 / A09) ou votre ordinateur personnel avec la machine virtuelle fournie par le REDS.

 **L'installation de la machine virtuelle doit se faire en dehors des séances de laboratoire afin que vous puissiez profiter de poser des questions pendant le laboratoire. L'installation n'est pas comptée dans les périodes nécessaires à la réalisation de ce laboratoire.**

Fichiers

Vous devez télécharger à partir du site Cyberlearn un .zip contenant un répertoire «workspace» où vous trouverez :

- **labo_processeur.circ** : Le fichier de travail Logisim
- **main.S** : fichier source du code assembleur
- **Makefile** : fichier contenant les directives d'assemblage

 **Le fichier Makefile ne doit pas être modifié!**

Workspace fourni

Ce workspace sera utilisé uniquement lors de ce laboratoire d'introduction.

Vous allez recevoir un circuit qui contient :

- Mémoire d'instructions
- Processeur_ARO2

1 Laboratoire instructions

Travail à effectuer

Entité du bloc pour la mémoire d'instructions



Nom I/O	Description
addr_i	Adresse de l'instruction courante (en mot de 16 bits)
instr_o	Intruction courante à envoyer au processeur

Etape 1 : Implémenter un composant ROM comme mémoire d'instructions

Le processeur doit avoir accès à une mémoire d'instructions. C'est le processeur qui sera chargé de nous donner l'adresse courante afin que nous lui retournions l'instruction à traiter. Nous allons instancier un composant ROM qui a les caractéristiques suivantes :

- Taille du bus d'adresse : **15 bits**
- Taille du bus de donnée : **16 bits**

Ces valeurs vous sont fournies car elles sont définies par rapport à ce contexte et au processeur que nous allons réaliser. Il s'agit ici de contraintes qu'il a fallu définir.

Pour information : La taille du bus d'adresse permet directement de définir la taille du programme que nous pouvons exécuter dans le système.

Pour information : La taille du bus de donnée permet directement de définir, de manière indirecte, le nombre de registres ainsi que le nombre d'instructions supportés par le système.

Etape 2 : Ecrire un programme simple qui utilise quelques instructions supportées

Utiliser les fonctions *lsh* (Logical Shift Right), *mov* et *add* afin de pouvoir réaliser un algorithme qui effectue l'opération suivante :

$$R = (A + B + C + D)/4$$

Il faut qu'il y a une partie déclarative dans le programme qui permette de pouvoir changer simplement les valeurs à tester (utilisation de *mov* pour les valeurs A, B, C et D).

NOTE : Les différentes valeurs seront considérées comme des valeurs non-signées sur 16 bits. Considérer cette information lorsque vous choisissez les valeurs de tests. L'objectif de ce laboratoire n'est pas de devoir observer ce qui se passe en cas de dépassement mais que vous compreniez bien comment s'exécute le programme.

Exemple d'un programme assembleur (fichier main.S) :


```
mov r1, #5
add r0, r2, r3
```

NOTE : Un manuel des instructions ARM que nous allons supporter se trouve sur Cyberlearn. Il sert de référence quant à l'utilisation des différentes instructions.

Etape 3 : Compiler le programme écrit à l'étape 2

Vous allez compiler le programme que vous venez d'écrire. Vous avez deux options à partir de ce moment-là.

- **Geany** : Avec cet éditeur de texte, vous pouvez compiler avec le menu : Build -> Make.
- **terminal** : Placez-vous dans le dossier qui contient votre programme et tapez : make

 **Il ne faut absolument pas modifier le fichier Makefile**

Etape 4 : Charger le programme compilé dans la mémoire d'instructions

Aller dans le circuit instr_mem et sur la mémoire d'instructions :

1. clic-droit->clear contents : efface le contenu
2. clic-droit->load image : charge une image en mémoire
3. sélectionner le fichier main.raw

Vous devriez voir des valeurs dans la mémoire ROM d'instructions.

Etape 5 : Comparer le contenu de la mémoire avec les valeurs d'instructions compilées

Grâce au fichier main.lss, comparez les valeurs écrites dans la mémoire ROM d'instructions avec le programme que vous avez les instructions que vous avez écrites.

Etape 6 : Tester pas à pas afin d'observer le bon fonctionnement

Changer simplement les valeurs de l'entrée addr_i afin de pouvoir observer que vous fournissez bien les instructions en sortie.

Etape 7 : Tester avec l'émulateur python du processeur

Une version simplifiée du processeur a été implémentée en python afin que vous puissiez observer l'effet sur les registres. Le but étant de vérifier que vous arriviez correctement réaliser l'opération demandée précédemment. Valider le fonctionnement de votre programme avec l'émulateur. Voici les quelques étapes à suivre pour lancer cette partie :

- Télécharger l'émulateur python sur la page Cyberlearn.
- Installer les packages python nécessaires en rentrant les commandes suivantes dans un terminal :

```
$ pip3 install pysimplegui
$ sudo apt-get update
$ sudo apt-get install python3-tk
```

- Lancer l'émulateur depuis un terminal avec la commande :

```
$ python3 proc_aro_emu.py
```

Il faut maintenant fournir à l'émulateur le fichier "main.raw" que vous aviez généré à l'étape 4 et vous pourrez exécuter séquentiellement les instructions afin de voir l'effet sur les registres.

Conclusion

Lorsque vous arriverez à cette étape, nous considérons que tous les étudiants ont correctement installés les outils nécessaires.