

---

# Architecture des ordinateurs

# TD n° 3

---

## Exercice 1. Unités de la micro-architecture

Dans cet exercice, nous utiliserons la possibilité d'étendre les données d'entrées des circuits de 1 à 32 bits.

### Q- 1.1 Unité Arithmetique et logique

Réaliser puis tester une UAL (ALU) adaptée au jeu d'instruction RV32I. Dans un premier temps cette UAL pourra retourner des zéros pour les opérations de décalage de bits (les shift).

### Q- 1.2 Banc de registres

Réaliser un banc de 32 registres 32 bits pour le jeu d'instruction RV32I. Le registre R0 devra être câblé à zéro.

## Exercice 2. Datapath version 1.1

### Q- 2.1 Datapath 1.1

Réaliser puis tester une première version du chemin de données de la micro-architecture 1-bus avec les registres PC, IR, MAR, MDR, ainsi que les unités précédemment réaliser. Cette première version doit vous permettre d'exécuter les instructions de type R et leur instruction correspondante de type I.

### Q- 2.2 Simulation 1.1

Vous simulerez la version 1 du chemin de données de votre micro-architecture 1-bus en connectant votre chemin de données à une mémoire vive (RAM) de 64 lignes de 32 bits. Vous écrirez quelques instructions assembleurs et les codes binaires correspondants à ses instructions que vous écrirez dans la mémoire vive.

## Exercice 3. Datapath version 1.2

### Q- 3.1 Datapath 1.2

Réaliser et tester la version 1.2 du chemin de données de la micro-architecture 1-bus. Cette version 1.2 doit permettre d'exécuter les instructions de chargement/rangement (load / store).

### Q- 3.2 Simulation 1.2

Vous simulerez la version 1.2 du chemin de données de votre micro-architecture 1-bus en connectant votre chemin de données à une mémoire vive (RAM) de 64 lignes de 32 bits. Vous ajouterez quelques instructions assembleurs et les codes binaires correspondants à votre programme binaire afin de tester le bon fonctionnement de votre nouveau chemin de données.

## **Exercice 4. Datapath version 1.3**

### **Q- 4.1 Datapath 1.3**

Réaliser et tester la version 1.3 du chemin de données de la micro-architecture 1-bus. Cette version 1.3 doit permettre d'exécuter les instructions de branchement conditionnel (BEQ, BNE, BLT et BGE). Lors de l'exécution d'une instruction de branchement, comme par exemple BEQ r1, r2, offset, l'UAL est utilisé pour faire la soustraction  $r1 - r2$ , si le résultat est égale à zéro (comme indiqué par BEQ) alors le branchement doit être réalisé et la valeur immédiate se trouvant dans offset doit être ajouter au PC.

Vous réaliserez l'unité de logique combinatoire **BRANCH**. Cette unité prend en entrées les 3 bits de la fonction 3 (funct3) de l'instruction de branchement et le résultat de l'UAL. En sortie, l'unité **BRANCH** donne un bit Taken a 1 lorsque le branchement doit être pris et reste à 0 sinon.

### **Q- 4.2 Simulation 1.3**

Vous simulerez la version 1.3 du chemin de données de votre micro-architecture 1-bus en connectant votre chemin de données à une mémoire vive (RAM) de 64 lignes de 32 bits. Vous ajouterez quelques instructions assembleurs et les codes binaires correspondants à votre programme binaire afin de tester le bon fonctionnement de votre nouveau chemin de données.

## **Exercice 5. Datapath version 1.4**

### **Q- 5.1 Datapath 1.4**

Réaliser et tester la version 1.4 du chemin de données de la micro-architecture 1-bus. Cette version 1.4 doit permettre d'exécuter les instructions de branchement conditionnel (JAL et JALR).

### **Q- 5.2 Simulation 1.4**

Vous simulerez la version 1.4 du chemin de données de votre micro-architecture 1-bus en connectant votre chemin de données à une mémoire vive (RAM) de 64 lignes de 32 bits. Vous ajouterez quelques instructions assembleurs et les codes binaires correspondants à votre programme binaire afin de tester le bon fonctionnement de votre nouveau chemin de données.

## **Exercice 6. Circuit de contrôle**

### **Q- 6.1 Circuit de contrôle**

Avant de réaliser le circuit de contrôle, vous devez vous assurer du bon fonctionnement de votre chemin de données. En cas d'erreur, tout le travail devra être repris depuis le début.

Pour chaque instruction ou type d'instruction, vous devez écrire ses étapes de lecture, décodage et d'exécution avec la liste des signaux à activer. Vous pouvez vérifier ces étapes par les simulations du chemin de données.

A partir de ces étapes, vous pourrez dessiner l'automate du circuit de contrôle. Puis, en fonction du nombre d'états en déduire le nombre de bascules du circuit de contrôle. Enfin, vous pourrez écrire la table de vérité des bascules et les équations des bascules. Finalement, à partir des équations vous pourrez réaliser le circuit de contrôle.

### **Q- 6.2 Simulation**

Une fois le circuit de contrôle terminé, vous le connecterez à votre chemin de données pour vérifier le fonctionnement de l'ensemble.