

Rapport architecture microprocesseur

Yannis Kedadry

Oscar Garnier

Antoine Anastassiades

12 decembre 2022

1 Choix de l'architecture

Nous avons choisi de réaliser notre projet en nous basant sur l'architecture de RISC-V 32 bits. Nous utiliserons 32 registres.

2 Ensemble d'instructions

Nous comptons utiliser un ensemble de 28 instructions afin de réaliser des opérations sur des entiers non signés. Nous encoderons nos instructions de plusieurs façon différentes en fonction du type d'instruction (opérations avec immédiats I, avec saut U ou uniquement sur des registres R). Les différentes opérations que nous utiliserons seront :

- *add*, *sub* (types R) et *addi* (type I). Ce sont nos instructions arithmétiques.
- *sll*, *srl*, *sra* de types R et leur pendant de type I afin de réaliser des décalages bit à bit.
- les opérations logiques *and*, *or* et *xor* de type R et leur pendant de type I.
- les opérations de tests *slt* et *slti*.
- *lui* et *auipc* sont des instructions spéciales permettant de se déplacer sur la pile. Ce sont des instructions de type J.
- *beq*, *bne*, *blt*, *blti* et *bge* nous permettent de réaliser des branchements conditionnels. Ils sont de type I.
- *jal* et *jalr* permettent des branchements incondtionnels. Ils sont de type J.
- *lw* charge *sw* stocke des données en mémoire. Ils sont de type I.

Voici les différents encodage pour les différents types d'instructions. Avoir ces types nous permet de manipuler des immédiats de grande taille. Ce n'est peut-être pas impératif mais cela pourrait nous faciliter la tâche dans certains cas précis. *rd* représente les registres de destination et *rsX* les registre sources qui peuvent être unique ou en pair.

	31	20	19	15	14	10	9	5	4	0
R	immédiat		rs2		rs1		rd		opcode	
I	immédiat				rs		rd		opcode	
U	immédiat						rd		opcode	

3 Opcodes

Nos instructions suivront les opcodes suivant:

	000	001	010	011	100	101	110	111
00	Addi		Srli	Srai	Slli	Andi	Ori	Xori
01	Add	Sub	Srl	Sra	Sll	And	Or	Xor
10	Lw	Sw		Beq	Bne	Blt	Blti	Bge
11	Lui	Auipc	Jal	Jalr			Slt	Slti