

Université de Franche Comté

Architecture des Ordinateurs - Fiche TP 1

Les TP sont effectués en utilisant Linux.

Vous utiliserez le simulateur `logisim-evolution-3.8.0-all.jar` disponible ici

<https://github.com/logisim-evolution/logisim-evolution#download>

Pour lancer le simulateur, utilisez la commande, depuis un terminal (java 16 ou supérieur)

```
/opt/java/jdk.../bin/java -jar logisim-evolution-3.8.0-all.jar
```

Ex. 1 — Débuter avec Logisim

1. Tutoriel

Effectuez en détail les tutoriels intégrés suivants :

- Guide du débutant
- Design hiérarchique
- Caractéristiques supplémentaires.

2. Additionneur

Créez un nouveau projet contenant un demi-additionneur, un additionneur complet et un additionneur à propagation ayant deux entrées 2x4 bits.

3. Comparateur non signé

Utilisez l'outil d'analyse pour implémenter le circuit de calcul de $a_i < b_i$ (2 bits).

Ex. 2 — Composants du processeur

Vous complétez les circuits présents dans le fichier `MIPS-ENONCE.circ`

1. **Décodeur CO.** Complétez le circuit qui décode les signaux suivant en fonction de l'entrée nommée Co.

Co	Sortie Activée
0	zero
3	ijal
4	ibeq
5	ibne
9	iaddiu
10	islti

Co	Sortie Activée
11	isltiu
12	iandi
13	iori
14	ixori
15	ilui
37	ilbu
40	isb

2. **Décodeur NF.** Complétez le circuit qui décode les signaux suivant en fonction de l'entrée nommée Nf. Une sortie est validée seulement si le signal `en=1`. Pour cela utiliser l'entrée d'activation du décodeur.

Nf	Sortie Activée
0	isll
2	isrl
3	isra
4	isllv
6	isrlv
7	israv
8	ijr
33	iaddu

Nf	Sortie Activée
35	isubu
36	iand
37	ior
38	ixor
39	inor
42	islt
43	isltu

3. **ENCODEUR.** Complétez le circuit qui place dans un nombre 4 bits les valeurs des indices des entrées e0 à a11.
4. **UAL.** Complétez le circuit qui réalise les opérations suivantes sur a et b en fonction de du signal Op :

Op	Opération
0	A+B
1	A.B
2	A OR B
3	A NOR B
4	A XOR B
5	B SLL A

Op	Opération
6	B SRL A
7	B SRA A
8	A-B
9	A SLL 16
11	A - B < 0 (signé)
12	A < B (non signé)

5. **Comparateur à zéro**
Complétez le circuit ZERO qui renvoie 1 si l'entrée 32 bits est à zéro. Vous utiliserez une porte logique et un séparateur.