

## TP 2 : Circuits combinatoires

Dans ce TD vous allez construire des circuits à partir des portes de base : NOT, AND, NAND, OR, NOR et XOR. L'utilisation d'autres outils de Logisim est interdite (sauf LEDs, interrupteurs et le courant nul bien sûr). Une fois vos circuits créés, vous pourrez les réutiliser en construisant des modules.

### Additionneurs

*aussi appelés "plusseurs"!*

1. Réalisez un circuit qui effectue l'additionneur 1-bit qui a deux bits en entrée et deux bits en sortie : un pour le résultat et un pour la retenue.
2. Adaptez le circuit de la question précédente pour construire un additionneur 8-bits.
3. Le BCD (binary-coded decimal) est un codage qui représente chaque chiffre décimal par sa représentation binaire (sur 4 bits donc par chiffre). On suppose que les entiers sont tous positifs. Construisez un additionneur BCD 4-bits.

### Vers un ALU (UAL)

*No, it's not a country!*

1. On souhaite manipuler les entiers négatifs. Réalisez un circuit qui effectue l'addition et la soustraction des nombres sur 8 bits représentés en complément à 2.
2. Réalisez un circuit avec 6 entrées et une sortie telles que les 2 dernières entrées permettent de sélectionner une entrée parmi les 4 premières.
3. Utilisez les questions précédentes pour construire une unité d'arithmétique et logique (UAL) sur 2 bits qui effectue les opérations suivantes : OR, XOR, AND et l'addition.

### Bonus : Optimisation

*gcc -O3 -fno-ffast-math -march=native exo1.c*

1. On souhaite améliorer la performance de l'additionneur 8-bits en anticipant la retenue. Expliquez comment on peut "prédire" la retenue dans un tel additionneur.