

PORTES LOGIQUES

Le simulateur en ligne est accessible à l'URL suivante : <https://logic.modulo-info.ch>.

Exercice 0 Une porte *NON ET* permet de réaliser l'opération logique de même nom, défini par

$$a \text{ NON ET } b = \text{NON}(a \text{ ET } b)$$

1. Construis avec le simulateur une porte *NON ET* en utilisant une porte *NON* et une porte *ET*, puis recopie le circuit sur ton carnet de bord.
2. Écris une fonction Python `non_et(a, b)` qui implémente ce même opérateur.

Exercice 1 On considère la propriété suivante, aussi connue sous le nom de *loi de Morgan* :

$$\text{NON}(a \text{ ET } b) = (\text{NON } a) \text{ OU } (\text{NON } b)$$

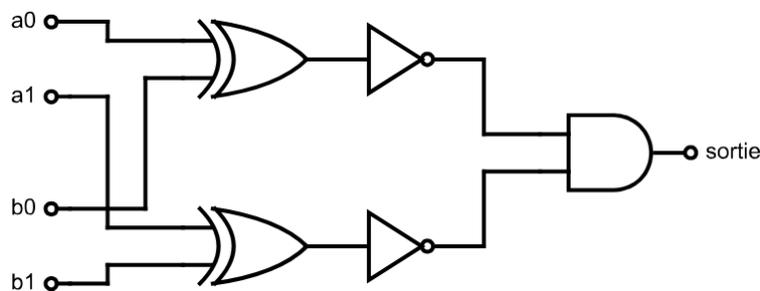
1. Construis avec le simulateur les deux circuits correspondant respectivement au membre de droite et au membre de gauche de cette égalité. Est-elle vérifiée ?
2. Construis la table de vérité de l'expression booléenne de droite à partir de la valeur des expressions $(\text{NON } a)$ et $(\text{NON } b)$.

Exercice 2

1. Construis le circuit correspondant à l'expression $(a \text{ ET } \text{NON } b) \text{ OU } (\text{NON } a \text{ ET } b)$.
2. Vérifie, en utilisant le simulateur, que ce circuit est équivalent à une porte *OUX*.

Exercice 3 On considère un circuit combinatoire appelé *comparateur*, qui permet de comparer des séquences de *bits* (on parle aussi de mot binaire), (a_1, a_0) et (b_1, b_0) de longueur 2. Le *bit* de sortie vaut :

- 1 si $(a_1, a_0) = (b_1, b_0)$;
- 0 sinon.



1. Construis ce circuit avec le simulateur.
2. Complète sa table de vérité.

a_1	a_0	b_1	b_0	sortie
0	0	0	0	...
0	0	0	1	...
...				

3. Écris une fonction `comparateur(a1, a0, b1, b0)` en Python renvoyant le même résultat.