

CIRCUITS LOGIQUES – TUTORIEL

1 Les fonctions logiques

1.1 Définitions

Une fonction logique est une application :

$$f : \{0, 1\}^n \mapsto \{0, 1\}$$

$$(x_1, \dots, x_n) \rightarrow f(x_1, \dots, x_n)$$

Les fonctions logiques admettent également d'autres représentations : circuits logiques, tables de vérité, tableaux de Karnaugh.

1.2 Les fonctions logiques fondamentales

- La fonction **NOT** ou $\bar{}$
 - NOT $a = \bar{a}$
 - Vrai si a est faux.
- La fonction **AND** ou \cdot
 - a AND $b = a \cdot b$
 - Vrai uniquement si a et b sont vrais.
- La fonction **OR** ou $+$
 - a OR $b = a + b$
 - Faux uniquement si a et b sont faux.
- La fonction **XOR** (ou exclusif) ou \oplus
 - a XOR $b = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b$
 - Vrai uniquement si a est vrai et b est faux ou si b est vrai et a est faux.
- La fonction **NAND** ou \uparrow
 - $a \uparrow b = \text{NOT}(a \text{ AND } b) = \overline{a \cdot b}$
 - Faux uniquement si a et b sont vrais.
- La fonction **NOR** ou \downarrow
 - $a \downarrow b = \text{NOT}(a \text{ OR } b) = \overline{a + b}$
 - Vrai uniquement si a et b sont faux.

Exercice 1 (Les tables de vérité)

si vous ne nous croyez pas...

– NOT

a	\bar{a}
0	1
1	0

– OR

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

– NAND

a	b	$a \uparrow b$
0	0	1
0	1	1
1	0	1
1	1	0

– AND

a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

– XOR

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

– NOR

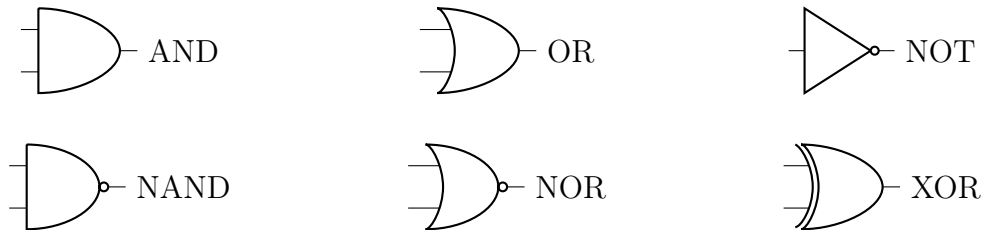
a	b	$a \downarrow b$
0	0	1
0	1	0
1	0	0
1	1	0

2 Les circuits logiques

Un circuit logique (ou combinatoire) est une autre représentation des fonctions logiques, permettant leur implémentation physique. Un circuit logique à n entrées et m sorties calcule une fonction

$$f : \{0, 1\}^n \mapsto \{0, 1\}^m \\ (x_1, \dots, x_n) \rightarrow (f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n))$$

où f_1, \dots, f_m sont des fonctions logiques. Les sorties ne dépendent que de l'état présent des entrées. On représente les portes logiques élémentaires par les logigrammes suivants :



Exercice 2 (Les portes)

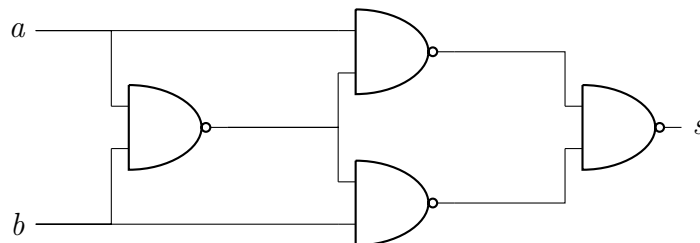
this is the end...

- (i) Construire une porte AND au moyen de deux portes NAND.
- (ii) Construire une porte OR au moyen de deux portes NOR.
- (iii) Utiliser des portes NAND pour construire les circuits ayant les sorties respectivement NOT, OR, AND.
- (iv) Idem en utilisant uniquement des portes NOR.

Pour les exercices suivants, utilisez le simulateur de circuits `tkgate`*. Vous commencerez par faire le tutoriel proposé dans l'accueil du logiciel.

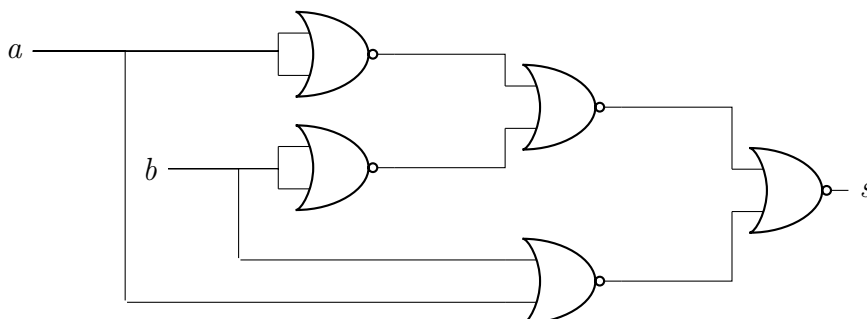
Exercice 3

Qu'est-ce qu'il fait lui ?



Exercice 4

et pi lui ?



Exercice 5 (On y va à deux)

alors, il y a erreur ou pas ?

Réalisez un circuit à 8 entrées (un octet) et dont la sortie est 0 si le nombre d'entrées à 1 est pair et 1 sinon.

*. <http://www.tkgate.org/>