

Exercices d'application : Les portes logiques

Lorsqu'on regarde un circuit électrique, on distingue généralement deux états : il y a du courant ou il n'y en a pas. Plus concrètement, on distingue l'état "bas", avec une faible tension, l'état "haut" avec une tension "normale".

Ces deux états sont facilement associés aux deux valeurs de l'algèbre de Boole.

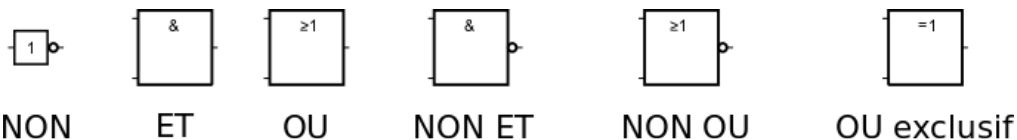


Repères historiques

L'invention du transistor a permis de construire des circuits logiques correspondant aux opérations booléennes. Il fut inventé en 1947 par les américains John Bardeen, William Shockley et Walter Brattain. Le transistor n'est pas le premier composant permettant de faire de tels circuits.

L'ordinateur Colossus utilisait des tubes électroniques dès 1943. Mais le transistor était bien plus fiable et plus petit, permettant de faire ainsi des circuits plus complexes. De nos jours, les transistors sont directement gravés sur les plaques de silicium et un processeur en compte des millions, voire des milliards.

À l'aide des transistors, il est possible de fabriquer des composants, appelés portes logiques, réalisant des opérations booléennes. Les principales portes logiques sont les suivantes :



Il est possible de fabriquer des circuits à l'aide de ces portes afin de faire des opérations plus complexes.

Objectifs :

- Illustrer les principes de calculs mis en œuvre dans un ordinateur.
- Créer une fonction logique ou arithmétique à partir de fonctions logiques élémentaires

1. Association de portes logiques

La porte NAND (NON ET) est dite universelle car elle permet de fabriquer toutes les autres portes logiques. Les associations de portes Non ET suivantes permettent de réaliser n'importe quelle fonction élémentaire. Ces montages sont disponibles sur Moodle dans le fichier de simulation **ex1.circ**.

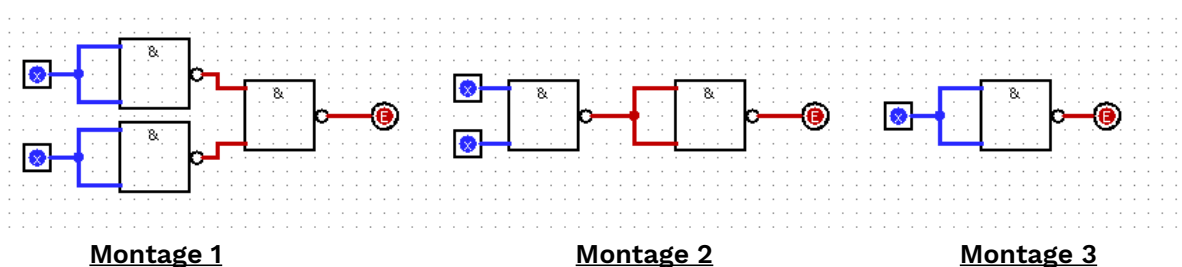


Figure 1: Différents montages à partir de portes NAND



Q1. Ouvrir le fichier **ex1.circ** dans le logiciel portable Logisim (disponible dans la boîte à outil sur Moodle). **Simuler** le fonctionnement de ces montages en agissant sur les états d'entrée et en relevant les codes de sortie. **Identifier** les portes logiques correspondant à ces circuits.

2. L'addition, l'opération de base dans un ordinateur

Une addition sur 1 bit, sans retenue, peut simplement être réalisée avec une porte OU Exclusif dont le fonctionnement a été étudié lors du TD précédent.

Porte OU exclusif

On rappelle qu'une porte OU exclusif (\oplus) est définie par l'équation $s = \bar{a} \cdot b + a \cdot \bar{b}$. Elle respecte donc la table de vérité suivante.

a	b	s = a \oplus b
0	0	0
0	1	1
1	0	1
1	1	0

Q2. Dessiner sur logisim le circuit de cette fonction en associant de portes ET, OU, NON. **Simuler** son fonctionnement et **valider** son fonctionnement.

L'additionneur 1 bit

Un additionneur 1 bit de base comporte deux entrées a, b (bits à additionner). La sortie, bit résultat de l'addition est noté S et la retenue éventuelle du résultat est noté R. Le montage de l'additionneur de base est disponible sur Moodle dans les fichiers associés à l'activité sous le nom **add1b.circ**.

Q3. Simuler le montage de l'additionneur 1 bit avec logicsim et **relever** sa table de vérité. **Justifier** que ce montage réalise une addition.

Additionneur 1 bit complet

Un additionneur complet nécessite une entrée supplémentaire : une retenue. L'intérêt de celle-ci est de permettre le chaînage des circuits afin de faire des additions sur plusieurs bits.

Un additionneur complet comporte donc trois entrées a, b (bits à additionner) et c (retenue). Comme pour l'additionneur de base, la sortie, bit résultat de l'addition est noté S et la retenue éventuelle du résultat est noté R.

Q4. Etablir la table de vérité de ce système.

Le montage de cette opération élémentaire est disponible sur Moodle dans les fichiers associés à l'activité sous le nom **add1complet.circ**.

Q5. Simuler le montage de l'additionneur 1 bit complet avec logicsim et **justifier** que ce montage réalise une addition.



Additionneur 4 bits

Un véritable additionneur doit pouvoir additionner des nombres binaires de 4 bits minimum. Ce type d'opérateur peut être conçu en s'inspirant de la démarche utilisée pour effectuer une addition sur le papier. En effet, dans une addition, les bits du même rang des deux nombres sont additionnés entre eux, le bit de droite de cette addition correspond alors au résultat et celui de gauche est utilisé comme retenue pour l'opération du rang suivant (voir schéma ci-contre).

Pour effectuer l'addition de deux nombres binaires de 4 bits, il est donc possible de mettre en cascade 4 additionneurs complets, chacun réalisant l'opération avec les deux bits d'un même rang et en additionnant une éventuelle retenue provenant du rang précédent.

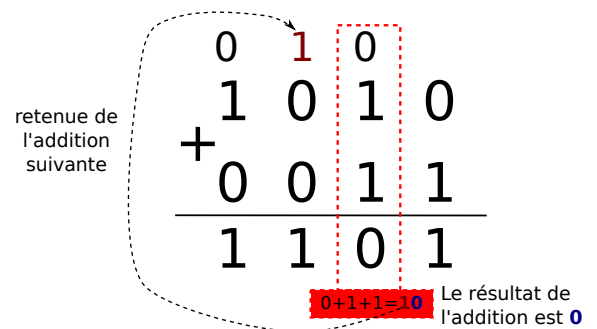


Figure 2: Principe du calcul de l'addition

Le fichier **add4b.circ** contient la structure d'un additionneur 4 bits. Il est composé de quatre additionneurs complets (étudié précédemment).

Q6. A partir de la description faite au dessus, **compléter** le schéma de câblage de l'additionneur 4 bits présent dans le fichier **add4b.circ**.

Q7. Simuler le fonctionnement de la structure et **relever** les valeurs des sorties S0 à S3 et R pour les valeurs suivantes :

Nbre A				Nbre B				Résultat S et retenue R					Conversion
a3	a2	a1	a0	b3	b2	b1	b0	R	S3	S2	S1	S0	
1	0	0	1	0	0	0	1						
0	0	1	1	0	0	1	0						
0	0	0	1	1	1	0	0						
1	1	0	0	1	0	1	0						

Q8. Convertir en décimal chaque grandeur binaire du tableau précédent. **Conclure** en validant le montage.

