NSI en première (2019-2020)
Fonctionnement 1

Une machine ne comprend que le langage binaire. Un langage de programmation, compréhensible par un humain, est traduit par un compilateur ou un assembleur, et permet de donner des instructions à la machine. Le fonctionnement d'une machine repose sur des circuits électroniques. Les liens qui existent entre ces circuits, le calcul logique et le calcul binaire doivent être compris.

1 Circuits et fonctions booléennes

On travaille avec des circuits qui laissent passer un courant ou pas. Les deux états de tels circuits correspondent aux deux chiffres 1 et 0 ou aux deux valeurs booléennes vrai et faux.

Un nombre écrit en binaire peut alors être représenter à l'aide de plusieurs circuits montés en parallèle.

On construit des circuits permettant de traduire des opérations simples :

- l'inversion d'un bit (on passe de 0 à 1 ou de 1 à 0) avec une entrée et une sortie ;
- le ET logique, avec deux entrées et une sortie ;
- le OU logique, avec deux entrées et une sortie ;
- la somme de deux bits avec deux entrées et deux sorties (une sortie contient la retenue).

Ces seuls petits circuits permettent de calculer l'opposé d'un nombre et d'additionner deux nombres.

On complexifie ensuite le système pour traduire d'autres opérations mathématiques, toutes définies à l'aide de l'addition.

2 Portes logiques

On appelle *portes logiques*, des fonctions logiques de base qui permettent de réaliser n'importe quelle autre fonction logique plus complexe. Et comme un calcul quelconque peut se réduire à un ensemble de calculs arithmétiques et logiques simples, ce calcul peut être réalisé de manière physique par un circuit logique.

Les opérateurs logiques utilisé dans le calcul sont : NOT, AND, OR, XOR, (NON, ET, OU, OU exclusif). Des expressions complexes sont construites avec ces opérateurs. Pour réaliser concrètement une fonction logique, nous avons des fils d'entrée, un par variable de la fonction, qui apportent la valeur 0 ou 1 de la variable correspondante et un fil de sortie qui transmet la valeur 0 ou 1 calculée par le circuit. Dans une machine, le calcul de la fonction par le circuit dure quelques nanosecondes. Ces fonctions, ou portes logiques, transforment des signaux d'entrée en un signal de sortie.

Les éléments physiques utilisés peuvent être des transistors. Ce sont des composants munis de trois broches auxquelles sont connectés des fils électriques. Une tension électrique appliquée sur ces broches peut représenter soit 0 soit 1. Parmi ces trois broches, deux servent à la circulation du courant, la troisième commande cette circulation. C'est le même principe qu'un interrupteur commandé par la troisième broche : le courant passe ou ne passe pas entre les deux premières broches selon l'état de la troisième.

Une porte logique est définie par la composition de ces circuits. Par exemple :

- la porte logique NOT par un circuit inverseur ;
- la porte logique OR par deux transistors en parallèle ;
- la porte logique AND par deux transistors en séries (traduit le produit).

La porte XOR est plus complexe à réaliser. On trouve aussi d'autres portes logiques de base comme NOR, NAND et XNOR. NOR signifie NOT OR, soit : a NOR b = NOT (a OR b). NAND signifie NOT AND, soit : a NAND b = NOT (a AND b). La troisième, XNOR, signifie NOT XOR, soit : a XNOR b = NOT (a XOR b).

La porte logique NOR par exemple peut être représentée par deux portes, une porte OR puis une porte NOT. On peut aussi choisir une porte OR et inverser dans le circuit l'état haut et l'état bas (le 5 volt et la masse, soit le 1 et le 0).

L'important est de comprendre qu'on peut réaliser toutes les portes logiques avec des transistors utilisés dans la construction des micro-processeurs. Ils fonctionnent comme des interrupteurs avec une entrée, une sortie, un bouton. On appuie sur le bouton, le circuit est fermé, la lumière s'allume. On appuie à nouveau, le circuit est ouvert, la lumière s'éteint.

Remarquons que dans nos ordinateurs, on ne trouve plus un transistor tout seul avec ses trois broches. Les transistors sont gravés dans du silicium pour constituer un circuit intégré. Le silicium est un matériau semi-conducteur et nous pouvons observer que sa production peut poser des problèmes du point de vue écologique.

Il reste à passer des portes logiques aux calculs mathématiques. La première opération à définir physiquement est l'addition. Les autres opérations s'en déduiront.

3 Circuit additionneur

Un *circuit additionneur* est un circuit qui admet en entrée deux mots de n bits, et fournit en sortie le résultat de l'addition binaire des deux mots d'entrée, sur $(n+1)$ bits. C'est un circuit complexe. Un circuit plus élémentaire est par exemple un demi-additionneur 1 bit, c'est-à-dire un circuit à 2 entrées et 2 sorties qui réalise l'addition binaire de deux nombres à 1 bit.

On associe 0 à faux et 1 à vrai. Chacune des deux entrées a pour valeur vrai ou faux. Les deux sorties notées s et r sont appelées somme et retenue. Par exemple : $1+0=01$ et dans ce cas la sortie s vaut 1 et la retenue r vaut 0, ou $1+1=10$ et dans ce cas s vaut 0 et r vaut 1.

La somme s est réalisée avec une porte XOR, la retenue r avec une porte AND : s = a XOR b et r = a ET b.

On vérifie avec une table de vérité que l'on obtient bien la somme et la retenue sur 1 bit :

a	b	a and b	a xor b
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	0

$1+1=10$, $1+0=01$, $0+1=01$, $0+0=00$.

Pour passer à un additionneur complet 1 bit, nous devons prendre en compte la retenue qui se propage et cela nécessite trois entrées a, b, re et deux sorties s et rs, où re est la retenue en entrée et rs la retenue en sortie. Nous avons les résultats suivants pour s et rs :

s = (a XOR b) XOR re, et rs = (a AND b) OR ((a XOR b) AND re)

On vérifie avec la table de vérité pour la sortie s :

a	b	re	a xor b	(a xor b) xor re
1	1	0	0	0
1	1	1	0	1
1	0	0	1	1
1	0	1	1	0
0	1	0	1	1
0	1	1	1	0
0	0	0	0	0
0	0	1	0	1

On vérifie avec la table de vérité pour la retenue rs :

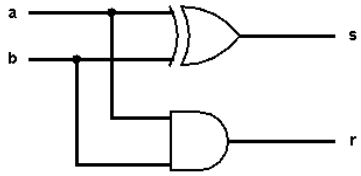
a	b	re	a xor b	a and b	(a xor b) and re	(a and b) or ((a xor b) and re)
1	1	0	0	1	0	1
1	1	1	0	1	0	1
1	0	0	1	0	0	0
1	0	1	1	0	1	1
0	1	0	1	0	0	0
0	1	1	1	0	1	1
0	0	0	0	0	0	0
0	0	1	0	0	0	0

L'utilisation d'un simulateur offre la possibilité de concevoir une représentation effective des circuits mais aussi de les voir fonctionner. Le logiciel Logisim par exemple permet de construire des circuits à l'aide de portes logiques et de les faire fonctionner en gérant les entrées et les sorties.

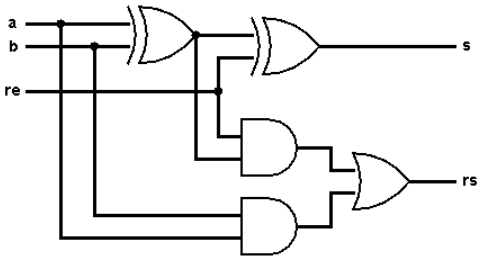
On trouve ce logiciel à l'adresse `http://www.cburch.com/logisim/`.

Les représentations graphiques des portes logiques sont présentées ci-dessous avec, de gauche à droite, les portes NOT, AND, OR, XOR, NAND, NOR, XNOR :

Un circuit demi-additionneur un bit peut se représenter à l'aide du schéma suivant :



Un circuit additionneur un bit avec retenue en entrée et retenue en sortie, donc trois entrées et deux sorties, peut se représenter à l'aide du schéma suivant :



Logisim permet de réaliser toutes sortes de circuits, des plus simples aux plus complexes comme une unité arithmétique et logique qui simule une calculatrice (avec opérations logiques, addition, soustraction), avec des mémoires RAM, des registres, etc.