Annexe 2

Rappels d'algèbre de Boole

Cette structure algébrique a été étudiée par le mathématicien anglais Georges Boole (1815-1864) pour formaliser les règles de la logique des propositions. Elle a été publiée dans son ouvrage: "The Mathematical Analysis of Logic" en 1847

A2.1. Définition

On appelle *algèbre de Boole*, un quadruplet (B, -, •, +) composé d'un ensemble B = {0, 1}, d'une opération unaire: - $\in$ B $\rightarrow$ B appelée complémentation , et de deux opérations binaires: •, + $\in$ BxB $\rightarrow$ B appelées respectivement "et" et "ou".

Pour tout a, b ,c $\in$ B on a les égalités suivantes:

(a • b) • C = a • (b • c)	(a + b) + c = a + (b + c)
a • c = c • a	a + b = b + a
a • (b + c) = (a • b) + (a • c)	a + (b • c) = (a + b) • (a + c)
a + 0 = a	a • 1 = a
a • -a = 0	a + -a = 1

En partant de ces dix axiomes on constate que 0 est l'élément neutre de l'opération + et 1 celui de l'opération •.

De même, on montre que:

-(a • b) = -a + -b
-(a + b) = -a • -b

appelée la *loi de De Morgan*.

Il existe plusieurs formes possibles d'écriture de l'algèbre de Boole. Les éléments de l'ensemble B peuvent, par exemple, être appelés V, F et les opérations $\cap$, $\cup$. La complémentation est quelquefois notée par un surlignement, ce qui est difficile à reproduire avec les outils typographiques habituels. Il est aussi possible de représenter une algèbre de Boole avec un seul opérateur binaire, par exemple le NI. On montre l'équivalence de cette forme en écrivant:

NI(a,b) = -(a + b)
d'ou
-a = NI(a,a)
a + b = NI(NI(a, b), NI(a, b))
a • b = NI(NI(a, a), NI(b, b))

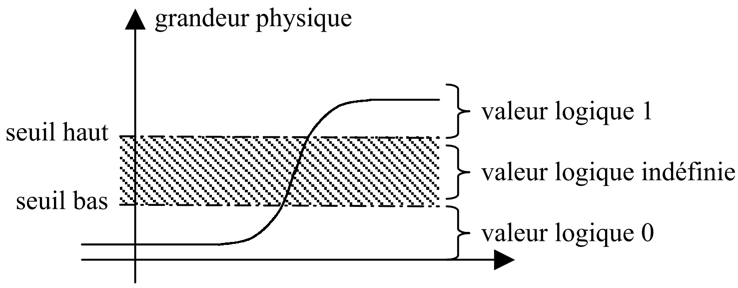
Une autre forme est l'*anneau booléen* noté (B, ⊕, +) dans lequel l'opérateur ⊕ est le *ou-exclusif* qui peut être défini par:

$a \oplus b = (a \bullet -b) + (-a \bullet b)$

A2.2. Interprétation

L'algèbre de Boole, initialement développée pour formaliser les problèmes de la logique des propositions, s'applique à de très nombreux domaines dans lesquels l'ensemble des valeurs se réduit à deux éléments. Dans le cas de l'électronique, ces valeurs pourront être deux niveaux de tension, deux intensités ou bien le fait qu'un élément soit conducteur ou isolant.

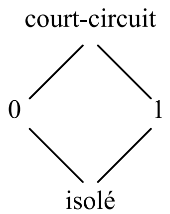
Comme beaucoup de ces grandeurs physiques sont continues, la définition des deux niveaux correspondant à des valeurs booléennes se fera via l'introduction de *seuils* hauts et bas.



Lorsque la valeur du signal est inférieure au seuil bas, on dira que sa *valeur logique* est 0, elle sera dite égale à 1 lorsque la valeur du signal est supérieure au seuil haut (ou réciproquement!). Lorsque la valeur du signal est entre les deux seuils, on dira que sa valeur logique n'est pas définie.

En fait beaucoup de montages électroniques se modélisent à l'aide de fonctions incomplètement définies. En plus des valeurs logiques, il est nécessaire de représenter soit une valeur non définie, soit un état isolé. Ceci peut être formalisé en utilisant des *logiques multivaluées*, ternaires, ou d'arité plus importante, opérant sur des treillis de valeurs (c'est le cas du type `std_ulogic` de la librairie IEEE de VHDL).

ex: treillis des états booléens + isolé



De telles approches nous permettent de parler d'interrupteurs qui peuvent isoler ou connecter des sources logiques.

A2.3. Fonctions booléennes

Nous définirons des *fonctions booléennes* à n arguments comme des applications: $f \in B^n \rightarrow B$. Les n arguments d'une telle fonction constituent un vecteur booléen qui peut prendre 2^n valeurs distinctes. Une fonction booléenne revient à attribuer des valeurs booléennes à ces valeurs des arguments. Elle peut donc être représentée par le tableau des 2^n valeurs de ces arguments auxquelles on associe la valeur de la fonction. Ce tableau est appelé la *table de vérité* de la fonction.

A2.3.1 Terme

On appelle *terme* un produit de variables booléennes directes ou complémentées.
ex: a•-b•c

A2.3.2 Forme canonique d'une fonction booléenne

On appelle *forme canonique* d'une fonction booléenne son écriture sous la forme d'une somme de termes, appelés *monômes*, car contenant l'ensemble des variables directes ou complémentées.
ex: f = a•-b•c + a•-b•-c + a•b•-c

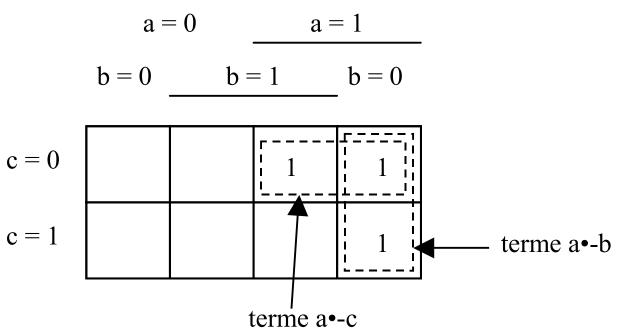
Cette écriture est assez lourde car elle comporte autant de termes que de valeurs à 1 dans la table de vérité de la fonction.

A2.3.3 Simplification d'une fonction booléenne.

Il est possible de réduire la complexité de l'écriture précédente en regroupant des monômes pour obtenir un nombre plus réduit de termes. Par exemple, les monômes a•-b•c et a•-b•-c pourront être condensés dans le terme a•-b.
ex: f peut ainsi s'écrire: a•-b + a •-c

Si la fonction ne comporte que quelques variables, ce processus de simplification pourra être réalisé sur une représentation tabulée de cette fonction appelée *table de Karnaugh*.

ex: table de Karnaugh de la fonction précédente:



Les zones rectangulaires pointillées dans ce tableau représentent des termes.

L'écriture comportant le minimum de terme sera appelée *forme minimale* et ses termes des *min-termes*. Elle sera utilisée pour minimiser la complexité de certains organes comme des PLA.

Il existe des outils informatiques [1, 2] très puissants, capables de minimiser des fonctions booléennes très complexes.

A2.3.4 Duale d'une fonction booléenne.

On appelle *duale* d'une fonction booléenne f (a, b, c) la fonction f* = -f(-a, -b, -c). On montre que la fonction f* s'obtient simplement en remplaçant les occurrences des opérateurs • par + et + par •.

ex: si f = a+b alors f* = a•b
si f = a•(b+-c) alors f* = a+(b•-c)

La notion de duale intervient dans l'étude des montages CMOS.

A2.3.5 Fonction auto-duales.

On appelle *auto-duale* une fonction booléenne égale à sa duale.
ex: f = a ⊕ b ⊕ c = -(-a ⊕ -b ⊕ -c) = f*

Bibliographie:

[1] R.E. Bryant: *Graph-Based Algorithms For Boolean Function Manipulation*, IEEE Transaction on Computers, Vol C-35 n°8, August 1986
[2] J-C. Madre, J-P. Billon: *Proving Circuit Correctness by Formaly Comparing their Expected and Extracted Behaviour*, Proceeding of the 25th Design Automation Conference, Anaheim, june 1988