CaMUS **7**, 21 – 32

Fonctions booléennes d'addition binaire

Julien Corriveau-Trudel

Résumé L'objectif de cet article est de familiariser le lecteur avec l'algèbre booléenne et les fonctions booléennes utilisées en informatique, aussi appelées « portes logiques ». L'exemple au coeur de l'article est l'additionneur binaire. Seul un niveau collégial en mathématiques est requis pour comprendre cet article.

1 Introduction

En 1937, Claude Shannon montre dans sa thèse de maîtrise [Sha37] qu'il y a des applications électroniques de l'algèbre booléenne, donnée par George Boole en 1854 dans The Laws of Thought [Boo54], notamment dans les circuits logiques et numériques. Aujourd'hui, on n'a qu'à suivre un cours d'introduction à la programmation pour découvrir que les ordinateurs conventionnels fonctionnent à l'aide de 1 et de 0 (de « on » et de « off », de circuit passant et non passant). Il est possible de décrire les opérations à la base du calcul informatique à l'aide de fonctions booléennes, soit des fonctions qui prennent des 1 et des 0 en entrées et qui renvoient 1 ou 0. En particulier, l'addition, lorsqu'appliquée aux chiffres d'un nombre en représentation binaire, s'écrit comme une composition de fonctions booléennes. C'est cette addition qui est décrite et vulgarisée dans le présent article.

Dans les quelques chapitres qui suivent, quelques concepts de base de l'algèbre booléenne, la notation binaire des nombres ainsi que les fonctions booléennes qui régissent l'addition binaire sont expliqués.

2 Algèbre booléenne

2.1 Espace booléen et fonctions booléennes

Soit E l'ensemble contenant les valeurs 0 et 1,

$$E = \{0, 1\}.$$

Cet ensemble est la base de l'algèbre booléenne, et toutes les fonctions booléennes agissent sur cet ensemble.

Définition 2.1. Une *fonction booléenne* est une fonction de E^k vers E, c'est-à-dire qu'elle peut prendre k éléments de E en entrée et renvoyer un 1 ou un 0. Une telle fonction est appelée fonction booléenne de *degré* k.

Définition 2.2. Définissons trois fonctions dites booléennes sur cet ensemble.

$$\begin{array}{rrcl} \text{Et}, \wedge: & E\times E & \rightarrow & E \\ & (x,y) & \mapsto & \begin{cases} 1, \text{ si } x=y=1 \\ 0, \text{ sinon} \end{cases} \\ \\ \text{Ou}, \vee: & E\times E & \rightarrow & E \\ & (x,y) & \mapsto & \begin{cases} 0, \text{ si } x=y=0 \\ 1, \text{ sinon} \end{cases} \\ \\ \text{Non}, \neg: & E & \rightarrow & E \\ & x & \mapsto & \begin{cases} 1, \text{ si } x=0 \\ 0, \text{ si } x=1 \end{cases} \end{array}$$

L'opération Et est l'équivalent de l'opérateur $\wedge$ dans la logique. Si l'on remplace 1 par Vrai et 0 par Faux, $\wedge$ a exactement le même comportement. Les deux termes entrant, soit (x, y), sont les termes à gauche et à droite de l'opérateur. Il en va de même de Ou et l'opérateur $\vee$. L'opération Non est l'équivalent de l'opérateur $\neg$, dont le seul argument est placé à droite. Le résultat est souvent appelé le *complément*[1]. Les fonctions $\wedge$ et $\vee$ sont de degré 2, alors que $\neg$ est de degré 1.

Considérant qu'il n'y a pas beaucoup de valeurs possibles pour un élément de E, générons les tables de valeurs de ces trois opérations :

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

x	¬ x
0	1
1	0

Table 1: Tables de valeurs des opérations $\wedge$, $\vee$ et $\neg$.

On peut définir d'autres fonctions booléennes à partir de ces fonctions de base.

Définition 2.3. Définissons les fonctions Ssi et XOu, qui représentent respectivement le « Si et seulement si » et le « Ou exclusif » de la logique.

[1] À partir de cet endroit, les opérateurs $\wedge, \vee$ et $\neg$ seront utilisés dans le document.

$$\begin{aligned} \text{Ssi}, \ \Leftrightarrow: \quad E \times E &\to E \\ (x,y) &\mapsto \begin{cases} 1, \ \text{si } x = y \\ 0, sinon \end{cases} \end{aligned}$$

$$\begin{aligned} \text{XOu}, \oplus: \quad E \times E &\to E \\ (x,y) &\mapsto \begin{cases} 1, \text{ si } x \neq y \\ 0, sinon \end{cases} \end{aligned}$$

Voyons ensuite les tables de valeurs de ces opérateurs.

$\Leftrightarrow$	0	1
0	1	0
1	0	1

$\oplus$	0	1
0	0	1
1	1	0

Table 3: Tables de valeurs des opérations $\Leftrightarrow$ et $\oplus$.

Remarque 2.4. Il en va de soi que l'opérateur $\oplus$ est égal à la composition de $\neg$ et $\Leftrightarrow$, tel que $x \oplus y = \neg(x \Leftrightarrow y)$, mais il sera plus agréable à l'oeil d'éliminer les $\neg$. De plus, on peut décomposer la fonction Ssi comme une composition de fonction :

$$\text{Ssi}(x,y) = \text{Ou}\Big(\text{Et}(x,y), \text{Et}\Big(\text{Non}(x), \text{Non}(y)\Big)\Big) = (x \wedge y) \vee (\neg x \wedge \neg y) = x \Leftrightarrow y$$

On comprend que la composition des fonctions ci-dessus est l'équivalent de la phrase suivante : $x = 1$ et $y = 1$, ou $x = 0$ et $y = 0$, ce qui revient à la définition de l'opération Ssi. De plus, il est évident que l'écriture avec $\neg$, $\wedge$ et $\vee$ est beaucoup plus lisible.

Maintenant que les opérateurs $\wedge$, $\vee$, $\neg$, $\Leftrightarrow$ et $\oplus$ sont définis, établissons quelques règles pour travailler avec ces opérateurs.

2.2 Simplification des fonctions booléennes

Regardons certains cas particuliers de simplification de fonctions.

Fonction	Idempotence	Tautologie/contradiction
$\vee$	$x \vee x = x$	$x \vee \neg x = 1$
$\wedge$	$x \wedge x = x$	$x \wedge \neg x = 0$
$\Leftrightarrow$	-	$x \Leftrightarrow x = 1$
$\oplus$	-	$x \oplus x = 0$

Table 5: Certaines identités des fonctions booléennes $\vee$, $\wedge$, $\Leftrightarrow$ et $\oplus$.

Ces 6 cas sont évidents et découlent des Tables de valeurs 1 et 3 de la Section 2. Voyons certaines propriétés des fonctions booléennes de base.

Nom	Identités
Loi de De Morgan	$\neg(x \vee y) = \neg x \wedge \neg y$
	$\neg(x \wedge y) = \neg x \vee \neg y$
Commutativité	$x \wedge y = y \wedge x$
	$x \vee y = y \vee x$
Associativité	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$
	$(x \vee y) \vee z = x \vee (y \vee z)$
Distributivité	$(x \wedge y) \vee z = (x \vee z) \wedge (y \vee z)$
	$(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$

Table 6: Certaines propriétés des fonctions booléennes $\wedge$, $\vee$ et $\neg$.

Ces identités et propriétés servent à rendre une équation, une fonction booléenne composée ou toute expression logique, moins complexe.

Exemple 2.5. On les applique sur la fonction suivante pour la simplifier :

$$\text{Ssi}(\text{XOu}(1{,}0), 0) = (1 \oplus 0) \Leftrightarrow 1 = (1) \Leftrightarrow 1 = 1.$$

Note 2.6. Le respect de la priorité des parenthèses est de mise.

Avant de pouvoir appliquer ces fonctions sur l'addition binaire, il faut expliquer ce que sont les nombres binaires.

3 Nombres binaires

Avant de décrire les nombres binaires, il serait intéressant de présenter une remarque sur les nombres utilisés couramment : les *nombres décimaux*. Les chiffres comme on les connaît sont 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. De plus, on sait que lorsqu'on additionne au dernier chiffre, soit 9, un 1, on obtient une *dizaine*. C'est le système commun de nombre, dit *en base 10*. Chaque position d'un nombre représente des puissances de la base, soit 10. L'unité est 10^0, la dizaine est 10^1, etc.

Or, qu'arriverait-il si la base était 2 ? La dizaine s'obtenait lorsqu'on additionne 1 à 1 : $1 + 1 = 10$. De même, à chaque position du nombre, lorsqu'on additionne un 1 avec un 1 , on doit ajouter 1 dans la position supérieure et conserver 0 dans la courante position, de sorte que $10 + 10 = 100$ et $100 + 100 = 1000$. On appelle ce système de nombres : les nombres *binaires*.

Aussi, comme le système en base 10, chaque position d'un nombre en base 2 représente une puissance de 2. Pour calculer la quantité représentée par un nombre en base 2, on prend le nombre à une certaine position et on le multiplie par l'ordre de grandeur de sa position. Par exemple, le nombre 1000101 est égal en nombre décimal à $(1 \times 2^6 + 1 \times 2^2 + 1 \times 2^0)_{10} = (64 + 4 + 1)_{10} = (69)_{10}$.

Notation 3.1. Les nombres décimaux qui peuvent porter à confusion seront notés $(X)_{10}$, à moins qu'il ne soit mentionné autrement.

De plus, une addition binaire peut se faire de la même façon qu'une addition décimale. On additionne les unités, on note la retenue qu'on additionne à la position suivante. Par exemple :

$$\begin{array}{rr} & 1 \\ + & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{rrr} & {\scriptstyle 1} & \\ & & 1 \\ + & & 1 \\ \hline & 1 & 0 \end{array}$$

TABLE 7: Exemples d'addition unitaire en système binaire.

$$\begin{array}{ccccccc} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & & \\ & & 1 & 0 & 1 & 0 & 1 \\ + & & 1 & 1 & 1 & 1 & 0 \\ \hline & 1 & 1 & 0 & 0 & 1 & 1 \end{array}$$

TABLE 8: Exemple d'addition à la main en système binaire.

À la Table 8, on a additionné

$$10101 = (16 + 4 + 1)_{10} = (21)_{10}$$

et

$$11110 = (16 + 8 + 4 + 2)_{10} = (30)_{10}.$$

En additionnant les nombres décimaux, on obtient 51. Le résultat binaire traduit en nombre décimal est

$$110011 = (32 + 16 + 2 + 1)_{10} = (51)_{10}.$$

Ainsi, une addition procède par le même principe en base 2 qu'en base 10, et on obtient le même résultat, à une traduction près. Toutefois, l'utilisation de la base binaire assure que les chiffres utilisés soient les éléments de $E = \{0,1\}$. Ainsi, il est possible d'appliquer des fonctions booléennes sur ces chiffres. Ceci mène au prochain chapitre, dans lequel est développé un additionneur de nombre binaire défini avec des fonctions booléennes.

4 Addition binaire

4.1 Définitions

Avant de décrire l'addition binaire en fonctions booléennes, on va définir l'addition binaire.

Définition 4.1. Une *addition binaire* est une addition en base 2 de deux nombres, soit x et y, chacun formé de $n + 1$ chiffres issus de $E = \{0, 1\}$. La longueur $n+1$ est sans perte de généralité, car on accepterait qu'un nombre commence avec des 0. Les retenues de l'addition sont notées r_k. La $k^{\text{ème}}$ retenue est la retenue de l'addition à la position $k-1$. Il y a des retenues de la position 1 à la position $n+2$. La somme de l'addition est de longueur maximale $n+2$, allant de la position 0 à la position $n+1$.

Remarque 4.2. On peut illustrer les termes des positions de l'addition binaire à 2 nombres, chacun de position au plus n, comme dans la Table 9 suivante. Ces mêmes termes seront réutilisés plus tard.

$$\begin{array}{ccccccc}
 & {\scriptstyle r_{n+1}} & {\scriptstyle r_n} & \cdots & {\scriptstyle r_2} & {\scriptstyle r_1} & \\
 & & x_n & \ldots & x_2 & x_1 & x_0 \\
+ & & y_n & \ldots & y_2 & y_1 & y_0 \\
\hline
 & s_{n+1} & s_n & \ldots & s_2 & s_1 & s_0
\end{array}$$

Table 9: Les termes des positions de l'addition binaire à 2 nombres de longueurs n.

4.2 Position de l'unité

Le premier objectif est de trouver une fonction booléenne qui imite l'addition à la position des unités. Toutefois, comme on peut le voir dans l'exemple de la Table 7, il est possible qu'une addition unitaire renvoie 2 chiffres. Or, une fonction booléenne telle que définie dans cet article ne peut renvoyer qu'une valeur. Il faudra donc deux fonctions : une qui renvoie la valeur unitaire, et l'autre qui renvoie la retenue de l'addition. Évaluons les résultats possibles d'addition unitaire afin de construire ces fonctions booléennes.

s_0	0	1
0	0	1
1	1	0

Table 10: Valeurs de l'*unité* (s_0) suite à l'addition binaire unitaire.

r_1	0	1
0	0	0
1	0	1

Table 11: Valeurs de la *retenue* (r_1) suite à l'addition binaire unitaire.

Les fonctions booléennes liées à ces résultats sont facilement identifiables.

Proposition 4.3. *Supposons une addition unitaire binaire $x+y=s$. Soit $x_0 \in E$ la valeur à la position des unités du premier nombre de l'addition et*

$y_0 \in E$ la valeur à la position des unités du deuxième nombre de l'addition. La valeur à la position des unités de la somme est donnée par s_0, définie :

$$\begin{array}{rccl} s_0 : & E \times E & \to & E \\ & (x_0, y_0) & \mapsto & x_0 \oplus y_0 \end{array}$$

Proposition 4.4. *Supposons une addition unitaire binaire $x + y = s$. Soit $x_0 \in E$ la valeur à la position des unités du premier nombre de l'addition et $y_0 \in E$ la valeur à la position des unités du deuxième nombre de l'addition. La valeur de la retenue à la position des dizaines de l'addition est donnée par r_1, définie :*

$$\begin{array}{rccl} r_1 : & E \times E & \to & E \\ & (x_0, y_0) & \mapsto & x_0 \wedge y_0 \end{array}$$

En résumé, la retenue qui sera utilisée à la position des dizaines dans l'addition est donnée par la fonction r_1 et la valeur de l'unité de la somme est donnée par s_0.

4.3 Position supérieure

On va maintenant chercher à déterminer deux fonctions : une qui renvoie s_1 et l'autre qui renvoie r_2. On va ensuite les généraliser.

La différence entre la somme de la position des dizaines et la somme à la position des unités, c'est que dans l'addition de la dizaine, on doit additionner la retenue. On cherche donc des fonctions booléennes de *degré 3*.

Prenons tous les cas possibles de valeurs des dizaines (x_1 et y_1) et de la retenue (r_1) et regardons les valeurs des dizaines de la somme (s_1) qui en résulte.

r_1	x_1	y_1	s_1
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

TABLE 12: Tables de valeurs de la position des dizaines (r_1, x_1, y_1 et s_1) dans l'addition binaire.

Comme on peut le constater dans la Table 12, la valeur à la position des dizaines de la somme est 1 lorsque :

$$\begin{aligned} &y_1 = 1 \text{ et } r_1 = x_1 = 0, \\ \text{ou } &x_1 = 1 \text{ et } r_1 = y_1 = 0, \\ \text{ou } &r_1 = 1 \text{ et } x_1 = y_1 = 0, \\ \text{ou } &r_1 = x_1 = y_1 = 1 \end{aligned}$$

On se rappelle qu'on peut transformer les « et » et les « ou » en forme logique. Ainsi, comme on vient d'exprimer la fonction logique sous forme de mot, il serait possible de la réécrire sous forme de fonction booléenne, comme ceci :

$$s_1 = (\neg r_1 \wedge \neg x_1 \wedge y_1) \vee (\neg r_1 \wedge x_1 \wedge \neg y_1) \vee (r_1 \wedge \neg x_1 \wedge \neg y_1) \vee (r_1 \wedge x_1 \wedge y_1) \quad (1)$$

Toutefois, on peut réfléchir à comment pré-simplifier cette fonction, en utilisant un fait qu'on connaît déjà : la somme unitaire binaire à 2 termes est déjà définie :

$$s_0 = x_0 \oplus y_0$$

Comme on ne cherche que les unités (c'est-à-dire pas la retenue), si on prenait l'opération $\oplus$ (Ou exclusif), qu'on applique sur y_1 et x_1, puis qu'on réapplique sur ce résultat avec la retenue r_1, cela fonctionnerait-il ? Autrement dit, on connaît l'addition à 2 termes. On cherche à prendre le résultat de cette addition et on refait une addition à 2 termes. Voici la formule présentée :

$$s_1 = (x_1 \oplus y_1) \oplus r_1 \quad (2)$$

Vérifions si cette formule renvoie les mêmes valeurs que l'addition à laquelle on s'attend, c'est-à-dire que les Équations 1 et 2 sont équivalentes.

r_1	x_1	y_1	$x_1 \oplus y_1$	$(x_1 \oplus y_1) \oplus r_1$	s_1
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	1	1

Table 13: Tables de valeurs de la position des dizaines (r_1, x_1, y_1, s_1 et $(x_1 \oplus y_1) \oplus r_1$) dans l'addition binaire.

On voit qu'on obtient exactement les mêmes valeurs avec les deux fonctions booléennes. Ces fonctions sont donc équivalentes.

Afin de généraliser le résultat, il est possible de pointer que l'addition à 3 termes se fasse sous les mêmes règles à la position des dizaines ou à la position des centaines. Les seules positions qui ne respectent pas ces règles sont la position des unités, car elle n'a pas de retenue à prendre en compte, et la position $n+1$, car il ne reste que la retenue. Ceci permet la proposition suivante, qui généralise la somme sans retenue à chaque position de l'addition binaire.

Proposition 4.5. *Supposons une addition binaire $x+y=s$. Soit $i \in 1, 2, \ldots, n$ et soient :*

$x_i \in E$ *la valeur à la $i^{\text{ème}}$ position du 1^{er} nombre de l'addition (x),*
$y_i \in E$ *la valeur à la $i^{\text{ème}}$ position du $2^{\text{ème}}$ nombre de l'addition (y) et*
$r_i \in E$ *la valeur à la $i^{\text{ème}}$ position des retenues de l'addition binaire.*

La valeur à la $i^{\text{ème}}$ position de la somme de l'addition est donnée par s_i, définie :

$$s_{i,\ i\in\{1,2,\ldots,n\}} : \begin{array}{rcl} E \times E \times E & \rightarrow & E \\ (x_i, y_i, r_i) & \mapsto & (x_i \oplus y_i) \oplus r_i \end{array}$$

Approchons d'un autre angle la retenue. On a précédemment comparé les tables de valeurs de deux fonctions afin d'en déduire l'équivalence. Plus loin, on utilisera plutôt le principe de simplification pour déduire l'équivalence.

Ainsi, considérons la retenue à la position 2 (qui sera ensuite généralisée). De même qu'à la recherche de la fonction de la somme, on va commencer par regarder les valeurs possibles de r_2. Prenons tous les cas possibles de valeurs des dizaines (x_1 et y_1) et de la retenue à la position des dizaines (r_1) et regardons le résultat attendu de la retenue (r_2).

r_1	x_1	y_1	r_2
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Table 14: Tables de valeurs de la position des dizaines et de la retenue des centaines (r_1, x_1, y_1 et r_2) dans l'addition binaire.

Il y a une retenue quand au moins 2 des trois valeurs sont 1. C'est normal, puisque pour changer de position, la somme doit être d'au moins 2. Voyons tous les cas possibles explicitement.

La retenue de la position supérieure (r_2) est 1 lorsque

$$\begin{aligned} & y_1 = x_1 = 1 \text{ et } r_1 = 0, \\ \text{ou } & x_1 = r_1 = 1 \text{ et } y_1 = 0, \\ \text{ou } & r_1 = y_1 = 1 \text{ et } x_1 = 0, \\ \text{ou } & r_1 = x_1 = y_1 = 1. \end{aligned}$$

Changeons ces cas en fonction booléenne.

$$r_2 = (r_1 \wedge \neg x_1 \wedge y_1) \vee (\neg r_1 \wedge x_1 \wedge y_1) \vee (r_1 \wedge x_1 \wedge \neg y_1) \vee (r_1 \wedge x_1 \wedge y_1)$$

Maintenant, simplifions cette équation en détail, afin de voir un processus de simplification, en utilisant les propriétés énumérées à la Section 2.2.

$$\begin{aligned} r_2 &= (r_1 \wedge \neg x_1 \wedge y_1) \vee (\neg r_1 \wedge x_1 \wedge y_1) \vee (r_1 \wedge x_1 \wedge \neg y_1) \vee (r_1 \wedge x_1 \wedge y_1) \\ &= \Big[\big((r_1 \wedge \neg x_1) \vee (\neg r_1 \wedge x_1)\big) \wedge y_1\Big] \vee (r_1 \wedge x_1 \wedge y_1) \vee (r_1 \wedge x_1 \wedge \neg y_1) \\ &= \Big[\big((r_1 \wedge \neg x_1) \vee (\neg r_1 \wedge x_1) \vee (r_1 \wedge x_1)\big) \wedge y_1\Big] \vee (r_1 \wedge x_1 \wedge \neg y_1) \\ &= \Big[\big((r_1 \wedge \neg x_1) \vee ((\neg r_1 \vee r_1) \wedge x_1)\big) \wedge y_1\Big] \vee (r_1 \wedge x_1 \wedge \neg y_1) \\ &= \Big[((r_1 \wedge \neg x_1) \vee x_1) \wedge y_1\Big] \vee (r_1 \wedge x_1 \wedge \neg y_1) \\ &= \Big[((r_1 \vee x_1) \wedge (\neg x_1 \vee x_1)) \wedge y_1\Big] \vee (r_1 \wedge x_1 \wedge \neg y_1) \\ &= \Big[(r_1 \vee x_1) \wedge y_1\Big] \vee (r_1 \wedge x_1 \wedge \neg y_1) \\ &= \Big[(r_1 \wedge y_1) \vee (x_1 \wedge y_1)\Big] \vee (r_1 \wedge x_1 \wedge \neg y_1) \\ &= (r_1 \wedge y_1) \vee \Big[x_1 \wedge \big((y_1 \vee (r_1 \wedge \neg y_1)\big)\Big] \\ &= (r_1 \wedge y_1) \vee \Big[x_1 \wedge \big((y_1 \vee r_1) \wedge (y_1 \vee \neg y_1)\big)\Big] \\ &= (r_1 \wedge y_1) \vee \Big[x_1 \wedge (y_1 \vee r_1)\Big] \\ &= (r_1 \wedge y_1) \vee (x_1 \wedge y_1) \vee (x_1 \wedge r_1) \end{aligned}$$

La comparaison de la table des valeurs de la formule initiale et de celle qui est simplifiée est inutile, car la simplification ne change pas les résultats. L'équation a passé d'une formule plutôt rébarbative à une forme dont on peut tirer une certaine logique. Intuitivement, le nouveau résultat propose que dès qu'il y a une paire dont les termes valent tous les deux 1, alors il y a une retenue. Le résultat a du sens. D'ailleurs, on peut comprendre qu'il n'est pas nécessaire non plus de connaître la valeur du 3e terme, car il n'affectera pas la retenue si deux termes valent déjà 1.

Il est important de noter que la règle de retenue est respectée pour toutes positions sauf la position 1. Ceci inclut la position r_{n+1}. On peut donc généraliser la fonction de retenue trouvée précédemment aux positions i, i allant de 1 à $n+1$.

Proposition 4.6. *Supposons une addition binaire $x+y=s$. Soit $i \in 1, 2, \ldots, n$ et soit :*

$x_i \in E$ *la valeur à la $i^{\text{ème}}$ positions du 1^{er} nombre de l'addition (x),*
$y_i \in E$ *la valeur à la $i^{\text{ème}}$ position du $2^{\text{ème}}$ nombre de l'addition (y) et*
$r_i \in E$ *la valeur à la $i^{\text{ème}}$ position des retenues de l'addition binaire.*

La valeur à la $(i+1)^{\text{ème}}$ retenue de l'addition est donnée par r_{i+1}, défini :

$$r_{i+1},\ {}_{i\in\{1,2,\ldots,n-1\}} : \begin{array}{rcl} E \times E \times E & \to & E \\ (x_i, y_i, r_i) & \mapsto & (r_i \wedge y_i) \vee (x_i \wedge y_i) \vee (x_i \wedge r_i) \end{array}$$

Comme à la position $n+1$ il n'y a pas de valeur de x et y, s_{n+1} devient r_{n+1} dans l'addition.

5 Conclusion

Il a été vu quelques théories de l'algèbre booléenne, notamment les fonctions booléennes de base ET, OU, NON, SSI et XOU, et leur simplification. Aussi, on a vu le principe de nombres binaires, les bases de nombre et l'addition booléenne sur les nombres binaires. Les formules pour chaque position de la retenue de l'addition binaire ainsi que la somme ont été construites avec les fonctions booléennes de base.

Pour résumé toutes les fonctions construites, on peut les rassembler les fonctions de retenues et de sommes à chaque position dans le schéma de l'addition binaire à n position suivant :

$$\begin{array}{ccccccc} & {\scriptstyle r_{n+1}} & {\scriptstyle r_n} & \cdots & {\scriptstyle r_2} & {\scriptstyle r_1} & \\ & & x_n & \ldots & x_2 & x_1 & x_0 \\ + & & y_n & \ldots & y_2 & y_1 & y_0 \\ \hline & s_{n+1} & s_n & \ldots & s_2 & s_1 & s_0 \end{array}$$

où x_i est la valeur à la $i^{\text{ème}}$ position du nombre x écrit sous forme binaire ($i \in \{0, 1, 2, \ldots, n\}$), y_i est la valeur à la $i^{\text{ème}}$ position du nombre y écrit sous forme binaire ($i \in \{0, 1, 2, \ldots, n\}$),

$$s_i = \begin{cases} x_i \oplus y_i, & \text{si } i = 0 \\ (x_i \oplus y_i) \oplus r_i, & \text{si } i \in \{1, 2, \ldots, n\} \\ r_i, & \text{si } i = n+1 \end{cases} \quad \text{et}$$

$$r_i = \begin{cases} x_i \wedge y_i, & \text{si } i = 1 \\ (r_i \wedge y_i) \vee (x_i \wedge y_i) \vee (x_i \wedge r_i), & \text{si } i \in \{2, \ \ldots, n+1\}. \end{cases}$$

Références

[Boo54] George Boole : *An Investigation of the Laws of Thought : On which are Founded the Mathematical Theories of Logic and Probabilities.* Walton and Maberly, 1854.

[Sha37] Claude Elwood Shannon : A symbolic analysis of relay and switching circuits. Master's thesis, Massachusetts Institute of Technology, 1937.

Julien Corriveau-Trudel
Département de Mathématiques, Université de Sherbrooke
Courriel: Julien.Corriveau-Trudel@USherbrooke.ca