

# BROUILLON - COMPTER EN PLACE LE NOMBRE DE BITS NON NULS D'UNE ÉCRITURE BINAIRE

CHRISTOPHE BAL

*Document, avec son source L<sup>A</sup>T<sub>E</sub>X, disponible sur la page  
<https://github.com/bc-writing/drafts>.*

---

## Mentions « légales »

Ce document est mis à disposition selon les termes de la licence Creative Commons “Attribution - Pas d'utilisation commerciale - Partage dans les mêmes conditions 4.0 International”.



En informatique, il existe une méthode astucieuse pour compter le nombre de 1 dans une écriture binaire. Expliquons comment faire avec l'écriture binaire 1101.1001.0110.1111 qui contient  $16 = 2^4$  chiffres. Ci-dessous, toutes les opérations se font en base 2. **Notez bien que l'on note les résultats dans une écriture binaire de taille 16 et c'est là la belle astuce du point de vue informatique.**

- On additionne les chiffres voisins puis on conserve le résultat sur deux chiffres.

$$\begin{array}{cccccccc} [1+1 & | & 0+1 & | & 1+0 & | & 0+1 & | & 0+1 & | & 1+0 & | & 1+1 & | & 1+1] \\ [10 & | & 01 & | & 01 & | & 01 & | & 01 & | & 01 & | & 10 & | & 10] \end{array}$$

- On additionne les paires voisines puis on conserve le résultat sur quatre chiffres.

$$\begin{array}{cccccccc} [10 & + & 01 & | & 01 & + & 01 & | & 01 & + & 01 & | & 10 & + & 10] \\ [ & 0011 & & | & 0010 & & | & 0010 & & | & 0100 & & ] \end{array}$$

- On additionne les quadruplets voisins puis on conserve le résultat sur huit chiffres.

$$\begin{array}{cccccccc} [ & 0011 & & + & 0010 & & | & 0010 & & + & 0100 & & ] \\ [ & & 00000101 & & & & | & & 00000110 & & & & ] \end{array}$$

- On additionne les octuplets voisins puis on conserve le résultat sur seize chiffres et on s'arrête pour analyser le résultat.

$$\begin{array}{cccccccc} [ & & 00000101 & & & + & & 00000110 & & & ] \\ [ & & & 0000000000000101 & & & & & & & ] \end{array}$$

- Le dernier nombre a pour valeur décimale 11 qui est exactement le nombre de 1 contenus dans 1101.1001.0110.1111.

Très joli! Non? Très bien! Mais pourquoi, ou comment cela marche-t-il? En fait c'est tout simple. Pour le voir reprenons les étapes en expliquant l'information qu'elles construisent.

- La 1<sup>re</sup> étape redonnée ci-dessous compte le nombre de 1 par paire de voisins. Il est important de noter qu'il n'y a pas de perte d'information car au maximum il y aura deux 1, or deux s'écrit 10 en base 2.

$$\begin{array}{cccccccc} [1+1 & | & 0+1 & | & 1+0 & | & 0+1 & | & 0+1 & | & 1+0 & | & 1+1 & | & 1+1] \\ [10 & | & 01 & | & 01 & | & 01 & | & 01 & | & 01 & | & 10 & | & 10] \end{array}$$

- La 2<sup>e</sup> étape redonnée ci-dessous compte le nombre de 1 par quadruplets de voisins sans perdre d'information car au maximum il y aura quatre 1, or 4 s'écrit 0100 en base 2.

$$\begin{array}{cccccccc} [10 & + & 01 & | & 01 & + & 01 & | & 01 & + & 01 & | & 10 & + & 10] \\ [ & 0011 & & | & 0010 & & | & 0010 & & | & 0100 & & ] \end{array}$$

- La 3<sup>e</sup> étape redonnée ci-dessous compte le nombre de 1 par octuplets de voisins sans perdre d'information.

$$\begin{array}{cccccccc} [ & 0011 & & + & 0010 & & | & 0010 & & + & 0100 & & ] \\ [ & & 00000101 & & & & | & & & 00000110 & & ] \end{array}$$

- La 4<sup>e</sup> étape redonnée ci-dessous compte le nombre de 1 pour 16 voisins, c'est à dire tous les chiffres, sans perdre d'information. Simple mais efficace !

$$\begin{array}{cccccccc} [ & & 00000101 & & & + & & 00000110 & & ] \\ [ & & & & 00000000000001011 & & & ] \end{array}$$

Vous noterez le caractère général des explications précédentes. Ainsi il est immédiat d'étendre la méthode à toute écriture binaire contenant  $2^p$  chiffres où  $p \in \mathbb{N}^*$ .