

# Présentation du binaire

Vers la fin des années 30, Claude Shannon démontra qu'à l'aide de « contacteurs » (interrupteurs) fermés pour « vrai » et ouverts pour « faux » il était possible d'effectuer des opérations logiques en associant le nombre 1 pour « vrai » et 0 pour « faux ».

Ce codage de l'information est nommé **base binaire**. C'est avec ce codage que fonctionnent les ordinateurs. Il consiste à utiliser deux états (représentés par les chiffres 0 et 1) pour coder les informations.

L'homme calcule depuis 2000 ans avant Jésus-Christ avec 10 chiffres (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), on parle alors de base décimale (ou base 10). Toutefois dans des civilisations plus anciennes ou pour certaines applications actuelles d'autres bases de calcul ont et sont toujours utilisées :

- base sexagésimale (60), utilisée par les Sumériens. Cette base est également utilisée dans le système horaire actuel, pour les minutes et les secondes ;
- base vicésimale (20), utilisée par les Mayas ;
- base duodécimale (12), utilisée par les anglo-saxons dans leur système monétaire jusqu'en 1960 : un « pound » représentait vingt « shilling » et un « shilling » représentait douze « pences ». Le système d'heure actuel fonctionne également sur douze heures (notamment dans la notation anglo-saxonne) ;
- base quinaire (5), utilisée par les Mayas ;
- base binaire (2), utilisée par l'ensemble des technologies numériques.

## Bit

Le terme **bit** (*b* avec une minuscule dans les notations) signifie « **binary digit** », c'est-à-dire 0 ou 1 en numérotation binaire. Il s'agit de la plus petite unité d'information manipulable par une machine numérique. Il est possible de représenter physiquement cette information binaire :

- par un signal électrique ou magnétique, qui, au-delà d'un certain seuil, correspond à la valeur 1 ;
- par des [aspérités géométriques](#) dans une surface ;
- grâce à des bistables, c'est-à-dire des composants électroniques qui ont deux états d'équilibre (l'un correspond à l'état 1, l'autre à 0).

Avec un bit il est ainsi possible d'obtenir deux états : soit 1, soit 0. Grâce à 2 bits, il est possible d'obtenir quatre états différents ( $2^2$ ) :

0 0  
0 1  
1 0  
1 1

Avec 3 bits, il est possible d'obtenir huit états différents ( $2 \times 2 \times 2$ ) :

Valeur binaire sur 3 bits	Valeur décimale
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Pour un groupe de  $n$  bits, il est possible de représenter  $2^n$  valeurs.

## Poids des bits

Dans un nombre binaire, la valeur d'un bit, appelée **poids**, dépend de la position du bit en partant de la droite. A la manière des dizaines, des centaines et des milliers pour un nombre décimal, le poids d'un bit croît d'une puissance de deux en allant de la droite vers la gauche comme le montre le tableau suivant :

Nombre binaire	1	1	1	1	1	1	1	
Poids	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

## Conversions

Pour convertir un mot binaire en nombre décimal, il suffit de multiplier la valeur de chaque bit par son poids, puis d'additionner chaque résultat. Ainsi, le mot binaire 0101 vaut en décimal :

$$\begin{aligned} & 2^3 \times 0 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 \\ & = 8 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 1 \\ & = 5 \end{aligned}$$

## Octet

L'**octet** (en anglais *byte* ou *B* avec une majuscule dans les notations) est une unité d'information composée de 8 bits. Il permet par exemple de stocker un caractère, tel qu'une lettre ou un chiffre.

Ce regroupement de nombres par série de 8 permet une lisibilité plus grande, au même titre que l'on apprécie, en base décimale, de regrouper les nombres par trois pour pouvoir distinguer les milliers. Le nombre « 1 256 245 » est par exemple plus lisible que « 1256245 ».

Une unité d'information composée de 16 bits est généralement appelée **mot** (en anglais *word*).

Une unité d'information de 32 bits de longueur est appelée **mot double** (en anglais *double word*, d'où l'appellation *dword*).

Pour un octet, le plus petit nombre est 0 (représenté par huit zéros 00000000), et le plus grand est 255 (représenté par huit chiffres « un » 11111111), ce qui représente 256 possibilités de valeurs différentes.

$$\begin{array}{cccccccc} 2^7 = 128 & 2^6 = 64 & 2^5 = 32 & 2^4 = 16 & 2^3 = 8 & 2^2 = 4 & 2^1 = 2 & 2^0 = 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

## KiloOctets, MégaOctets

Longtemps l'informatique s'est singularisée par l'utilisation de différentes valeurs pour les unités du système international. Ainsi beaucoup d'informaticiens ont appris que 1 kilooctet valait 1024 octets. Or, depuis décembre 1998, l'organisme international *IEC* a statué sur la question (<http://physics.nist.gov/cuu/Units/binary.html>). Voici donc les unités standardisées :

- Un kilooctet (ko ou kB) = 1000 octets
- Un Mégaoctet (Mo ou MB) = 1000 ko = 1 000 000 octets
- Un Gigaoctet (Go ou GB) = 1000 Mo = 1 000 000 000 octets
- Un Téraoctet (To) = 1000 Go = 1 000 000 000 000 octets

Attention ! De nombreux logiciels (parfois même certains systèmes d'exploitation) utilisent toujours la notation antérieure à 1998 pour laquelle :



- Un kilooctet (ko) =  $2^{10}$  octets = 1024 octets
- Un Mégaoctet (Mo) =  $2^{20}$  octets = 1024 ko = 1 048 576 octets
- Un Gigaoctet (Go) =  $2^{30}$  octets = 1024 Mo = 1 073 741 824 octets
- Un Téraoctet (To) =  $2^{40}$  octets = 1024 Go = 1 099 511 627 776 octets

## Opérations en binaire

Les opérations arithmétiques simples telles que l'addition, la soustraction et la multiplication sont faciles à effectuer en binaire.

## Addition binaire

L'addition en binaire se fait avec les mêmes règles qu'en décimale :

On commence à additionner les bits de poids faible (les bits de droite) puis on a des retenues lorsque la somme de deux bits de même poids dépasse la valeur de l'unité la plus grande (dans le cas du binaire : 1), cette retenue est reportée sur le bit de poids plus fort suivant...

Par exemple :

```
  0 1 1 0 1
+ 0 1 1 1 0
- - - - -
  1 1 0 1 1
```

## Multiplication binaire

La table de multiplication en binaire est très simple :

- $0 \times 0 = 0$
- $0 \times 1 = 0$
- $1 \times 0 = 0$
- $1 \times 1 = 1$

La multiplication se fait en formant un produit partiel pour chaque digit du multiplicateur (seuls les bits non nuls donneront un résultat non nul). Lorsque le bit du multiplicateur est nul, le produit partiel est nul, lorsqu'il vaut un, le produit partiel est constitué du multiplicande décalé du nombre de positions égal au poids du bit du multiplicateur.

Par exemple :

```
  0 1 0 1 multiplicande
x  0 0 1 0 multiplicateur
- - - - -
  0 0 0 0
  0 1 0 1
0 0 0 0
- - - - -
  0 1 0 1 0
```