


STI2D	<h1>Numération</h1>	
SIN - Terminale		

1. Le système décimal.....	1
2. Le binaire.....	1
3. L'hexadécimal.....	6

1. Le système décimal

Les nombres que nous utilisons habituellement sont ceux de la base 10 (système décimal).

Nous disposons de dix chiffres différents de 0 à 9 pour écrire tous les nombres.

D'une manière générale, toute base N est composée de N chiffre de 0 à N-1.

Soit un nombre décimal $N = 1871$. Ce nombre est la somme de 1 unités, 7 dizaines, 8 centaines et 1 milliers.

Ainsi :

$$\begin{aligned}
 1871 &= (1 \times 1000) + (8 \times 100) + (7 \times 10) + (1 \times 1) \\
 &= (1 \times 10^3) + (8 \times 10^2) + (7 \times 10^1) + (1 \times 10^0)
 \end{aligned}$$

10 représente la **base**

0, 1, 2 et 3 représente le **rang**

10^3 , 10^2 , 10^1 , 10^0 , représentent les **poids**. 10^0 est appelé le **poids faible**, 10^3 est appelé le **poids fort**

Quelque soit la base, le chiffre de droite est celui des unités.

2. Le binaire

2.1 Présentation

Dans les domaines de l'automatisme, de l'électronique et de l'informatique, nous utilisons la base 2. Tous les nombres s'écrivent avec deux chiffres uniquement (0 et 1). De même que nous utilisons le système décimal parce que nous avons commencé à compter avec nos dix doigts, nous utilisons le binaire car les systèmes technologiques ont souvent deux états stables.

Exemple d'applications du système binaire :

- Un interrupteur est ouvert ou fermé
- Une diode est allumée ou éteinte
- Une tension est présente ou absente
- Une surface est réfléchissante ou pas (CD)
- Un champ magnétique est orienté Nord-Sud ou Sud-Nord (disque dur)

A chaque état du système technologique, on associe un état logique binaire.
La présence d'une tension sera par exemple notée 1 et l'absence 0.

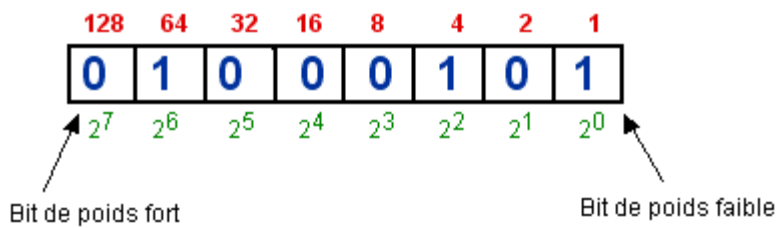
2.2 Définition du binaire

Base : 2

Poids : $2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0$ (64, 32, 16, 8, 4, 2, 1)

Chiffre : 0 ou 1 : le chiffre binaire (0 ou 1) est nommé "Bit" (Binary digit)

Exemple sur 8 bits (appelé un octet)



Avec un bit nous pouvons coder deux états



Avec deux bits nous pouvons coder quatre états



0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Avec trois bits nous pouvons coder huit états

A chaque nouveau bit, le nombre de combinaisons possibles est doublé.

Ce nombre est égal à 2^N (N étant le nombre de bits).

2.3 Passage du binaire au décimal

Objectif : écrire un nombre binaire en décimal

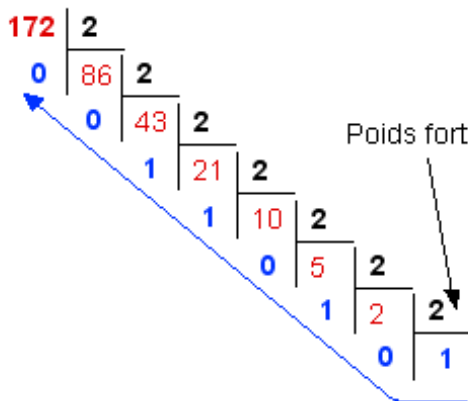
Les règles sont les mêmes que pour le décimal : exemple : $(100101)_2$

$$\begin{aligned}
 (100101)_2 &= (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\
 &= 31 + 0 + 0 + 4 + 0 + 1 \\
 &= 37
 \end{aligned}$$

2.4 Passage du décimal au binaire

Objectif : écrire un nombre décimal en binaire

Méthode 1 : divisions entières successives



On pratique des divisions par 2 successives (divisions entières) :

172 ÷ 2 = 86 (reste 0)
 86 ÷ 2 = 43 (reste 0)
 43 ÷ 2 = 21 (reste 1)
 21 ÷ 2 = 10 (reste 0)

On s'arrête quand la division entière devient impossible.

Le résultat se lit de droite à gauche en prenant les restes de chaque division

Le résultat est donc $(172)_{10} = (10101100)_2$

Méthode 2 : additions successives

poids	256	128	64	32	16	8	4	2	1

On décompose 172 en une suite d'addition (soustraction) des différents poids :

172 - 256 non donc on ne prend pas 256 (on place un 0 dans le tableau) —————→

172 - 128 = 44 donc on prend 128. Il reste 44 (on place un 1 dans le tableau) —————↑

44 - 64 Non donc on ne prend pas 64 (on place un 0 dans le tableau) —————↑

44 - 32 = 12 donc on prend 32. Il reste 12 (on place un 1 dans le tableau) —————↑

12 - 16 Non donc on ne prend pas 16 (on place un 0 dans le tableau) —————↑

12 - 8 = 4 donc on prend 8. Il reste 4 (on place un 1 dans le tableau) —————↑

4 - 4 = 0 donc on prend 4. (on place un 1 dans le tableau). Il reste 0 donc on met un 0 pour ce poids qui restent :

poids	256	128	64	32	16	8	4	2	1
	0	1	0	1	0	1	1	0	0

On peut ainsi voir que $172 = 128 + 32 + 8 + 4$

Et on en déduit : $(172)_{10} = (010101100)_2$ ou $(10101100)_2$

2.5 L'addition binaire

L'addition est réalisée bit à bit.

- 1 + 0 = 1
- 1 + 1 = 10 (0 et retenue = 1)
- 1 + 1 + 1 = 11 (1 et retenue = 1)

Exemple : 1001010 + 1101111 = ?

	1 0 0 1 0 1 0	
+	1 1 0 1 1 1 1	
	1 0 0 1 1 1 0	← retenues
	1 0 1 1 1 0 0 1	

2.6 Code Gray ou binaire réfléchi

En code Gray, pour passer d'une ligne à la suivante, on inverse un seul bit de telle manière qu'il soit le bit le plus à droite possible conduisant à un nouveau nombre.

Exemple sur 2 bits :

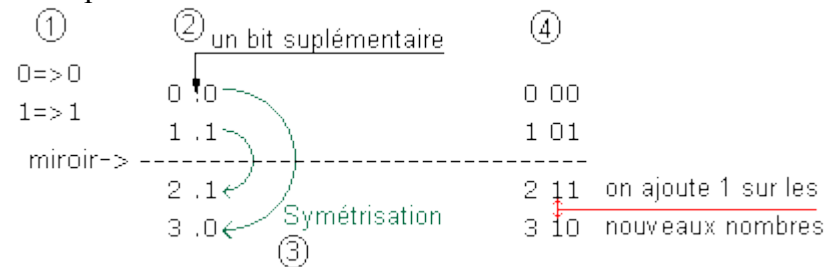
0	0	0	: on commence à zéro,
1	0	1	} on inverse un seul bit le plus à droite possible conduisant
2	1	1	
3	1	0	

Propriété supplémentaire : l'effet miroir

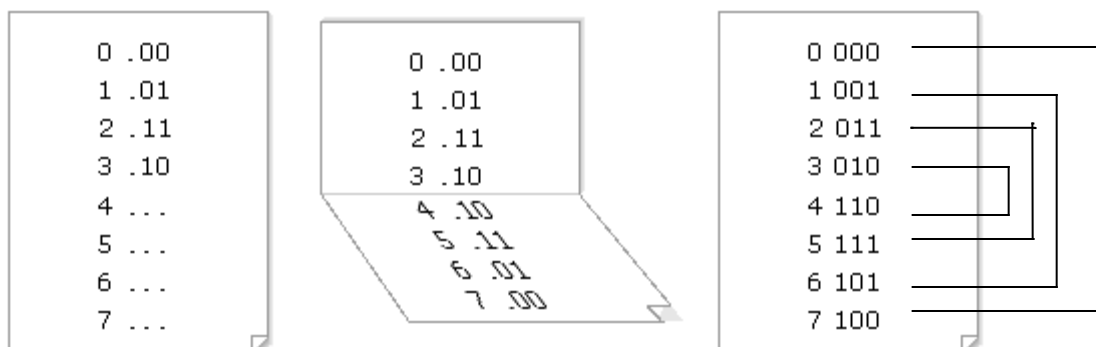
Le nom **code binaire réfléchi** vient d'une autre méthode de construction. Elle est plus pratique ou plus visuelle quand au choix du bit à inverser lors du passage d'un nombre au suivant.

A chaque fois qu'on a besoin d'un bit supplémentaire (2), on symétrise les nombres déjà obtenus (comme une réflexion dans un miroir) (3) et on rajoute un 1 au début des nouveaux nombres et un zéro sur les anciens (4).

Exemple sur 2 bits :



Exemple sur 3 bits :



2.7 D C B (Décimal Codé Binaire) (In english : B C D)

Principe : le nombre décimal est codé décade par décade :

exemple : 124 → 1 2 4
 ↓ ↓ ↓
 0001 0010 0100

Chaque décade est codée sur 4 bits (car une décade code de 0 à 9 c'est à dire 10 valeurs différentes. Or, 4 bits → $2^4 = 16$ valeurs différentes alors que 3 bits → $2^3 = 8$ valeurs différentes) .

Ainsi 124 en binaire = 0 1 1 1 1 0 0

124 en B C D = 0001 0010 0100

Intérêt : - Le décimal est utilisé par l'homme en permanence. Le codage en B C D permet de simplifier l'utilisation du binaire.

3. L'hexadécimal

Pour écrire les nombres en base 16 nous devons disposer de 16 chiffres, pour les dix premiers, nous utilisons les chiffres de la base 10, pour les suivant nous utiliserons des lettres de l'alphabet.

Chiffres :	Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	Hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Poids : $16^5, 16^4, 16^3, 16^2, 16^1, 16^0$ (65536, 4096, 256, 16, 1)

3.1 Passage de l'hexadécimal au décimal

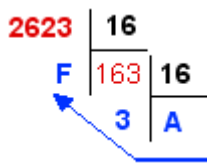
Les règles sont ici aussi les mêmes que pour le décimal.

Exemple : $(C1)_{16} = (C \times 16^1) + (1 \times 16^0) = (12 \times 16^1) + (1 \times 16^0) = 192 + 1 = 193$

3.2 Passage du décimal à l'hexadécimal

La méthode par divisions entières s'applique comme en binaire (sauf que cette fois on divise par 16)

exemple : $N = 2623$.



$2623 \div 16 = 163$ (il reste 15 c'est-à-dire F)
 $163 \div 16 = 10$ (c'est-à-dire A) (il reste 3)
 3 ne peut pas être divisé par 16 donc on s'arrête là
 Le résultat est lu de droite à gauche (dernier résultat + restes des autres divisions)

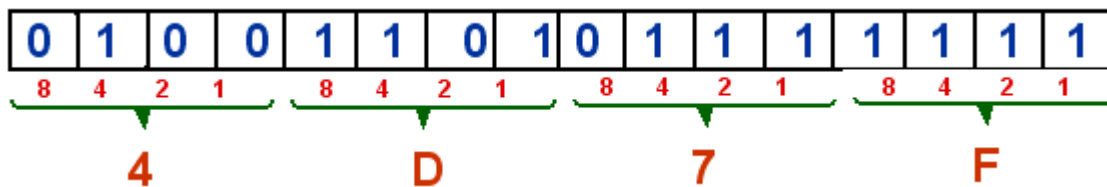
3.3 Correspondance entre binaire et hexadécimal.

La conversion du binaire en hexadécimal est très simple, c'est d'ailleurs la raison pour laquelle nous utilisons cette base.

Il suffit de faire correspondre un mot de quatre bits (quartet) à chaque chiffre hexadécimal.

Exemple 1: conversion d'un mot de 16 bits binaire en hexadécimal

Soit le chiffre binaire : $(0100110101111111)_2$



Donc $(0100110101111111)_2 = (4D7F)_{16}$

Exemple 2: conversion d'un chiffre hexadécimal en binaire

Soit le chiffre : $(3B2)_{16}$

On code nombre par nombre, en binaire, sur 4 bits : 3 B 2

0011 1011 0010

Donc $(3B2)_{16} = (001110110010)_2$