SYSTEMES DE NUMERATIONS ET CODAGES

1- Introduction

En binaire, on distingue trois principaux systèmes de codage :

- Binaire pur,
- Binaire DCB (Décimal Codé Binaire),
- Binaire réfléchi (code Gray).

En informatique on utilise aussi et surtout le codage :

- Hexadécimal,
- ASCII.

2- Définition

➢ Codage : Opération consistant à représenter des informations à l'aide d'un code.
➢ Codage binaire : Le code **binaire** utilise exclusivement les symboles **0 et 1** (systèmes logiques).
➢ Bit : C'est le chiffre élémentaire de la numérotation binaire.
➢ Mot : Groupe de "n" bits; un mot de **4 bits** s'appelle un **quartet**, **8 bits** s'appelle un **octet**...
➢ Poids : Coefficient attaché au rang d'un chiffre dans un système de numérotation. En numérotation binaire, on parle du **bit de plus faible poids (LSB)** qui est la position binaire de droite dans un mot et du **bit de plus fort poids (MSB)** qui représente le bit situé le plus à gauche dans mot.

3- Rappels sur le système décimal (base 10)

Le système décimal que nous employons, utilise 10 symboles : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.
Un nombre N s'écrit avec une succession de chiffres qui représentent les coefficients des puissances de 10.
Soit le nombre : $N_{(10)} = 2345$
A l'aide des puissances de 10 ce nombre s'écrit : $\mathbf{N_{(10)}}$ **=...**

Pour un nombre quelconque, nous aurons l'expression :

$$N_{(10)} = C_n \cdot 10^n + C_{n-1} \cdot 10^{n-1} + \ldots\ldots + C_1 \cdot 10^1 + C_0 \cdot 10^0$$

C_n : Coefficient compris entre 0 et 9 pour la base 10.

- **Pour une base quelconque, un nombre N peut s'écrire sous la forme :**

$$N_{(B)} = C_n \cdot B^n + C_{n-1} \cdot B^{n-1} + \ldots\ldots + C_1 \cdot B^1 + C_0 \cdot B^0$$

Avec :

- B : la base du système.
- Cn : le coefficient compris entre 0 et B-1.
- Bn : le poids du coefficient Cn.

4- Système binaire pur

La **base** de ce système est **2**; on utilisera seulement 2 chiffres : **0 ou 1**.

Exemple : Une lampe est **allumée** (**1**) ou **éteinte** (**0**).

Soit le nombre $N_{(2)}$ = 10101, en utilisant la méthode générale, on peut l'écrire sous la forme :

$N_{(2)}$ = 10101 =..

4-1 : Passage de la base 2 à la base 10

Il suffit d'appliquer la formule générale et, ensuite, d'effectuer la somme des différents termes :

$N_{(2)}$ = 10101 = ...

donc **$N_{(10)}$ = ...**

- **Chercher** le nombre décimal qui a pour écriture en bas 2 :

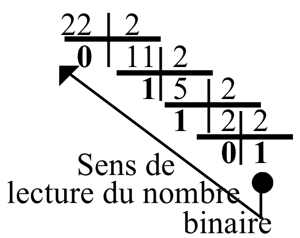
☞ $N_{(2)} = 11 \Rightarrow N_{(10)}$ = ...

☞ $N_{(2)} = 101 \Rightarrow N_{(10)}$ = ...

☞ $N_{(2)} = 11001 \Rightarrow N_{(10)}$ = ...

4-2 : Passage de la base 10 à la base 2.

Exemple, pour écrire le nombre $22_{(10)}$ en base 2, il faut diviser le nombre 22 par 2, inscrire le résultat en nombre entier à gauche du nombre divisé et ainsi de suite. On place, par la suite, un 1 au dessous de chaque nombre impaire et un zéro au dessous de chaque nombre pair.



La numération obtenu est donc :

$N_{(2)} = 10110 = 22_{(10)}$

- Application : **Coder** en binaire les nombres :

☞ 5 ⇒ ....

☞ 56 ⇒ ....

☞ 194 ⇒ ....

Remarque : Comme dans le système décimal, on définit pour le système binaire, les opérations additions, soustractions, multiplication et division (à voir dans un cours prochain).

5- Nombres en **D**écimal **C**odé **B**inaire (**DCB** ou **BCD**)

Il existe un autre code que le binaire pur qui fait appel, également, qu'à deux seuls symboles 0 ou 1; c'est le code décimal codé binaire (DCB ou BCD).
Pour passer d'un nombre décimal en un nombre décimal codé binaire, il suffit de prendre un à un les chiffres du nombre décimal et de le remplacer par son équivalent binaire.

Exemple : Convertir le nombre décimal 279 en BCD :

$2 \Rightarrow 0010_{(2)}$

$7 \Rightarrow 0111_{(2)}$

$9 \Rightarrow 1001_{(2)}$

donc **$279_{(10)} \Rightarrow 0010\ 0111\ 1001_{(BCD)}$**

- Application : **Donner** la correspondance en binaire pur et en Décimal codé binaire (BCD) des nombres suivants :

Nombre décimal	Binaire pur	BCD
9		
42		
301		

- **Indiquer** les nombres décimaux qui ont la même écriture en binaire pur et en BCD :

....

6- Le code binaire réfléchi (code Gray)

Dans ce codage, un seul bit change d'état lorsque l'on passe d'un terme au suivant. A l'apparition d'une variable supplémentaire on fait la symétrie du code déjà obtenu plus le nouveau bit à 1. Le code peut se refermer sur lui-même sans perdre ses propriétés dans la mesure ou le dernier terme se situe juste
avant un axe de symétrie.

c	b	a
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Le code Gray sert souvent dans des situations où d'autres codes, comme le binaire, peuvent produire des résultats ambigus ou erronés au moment de transitions entraînant le changement de plusieurs bits dans le code. Par exemple, en binaire, lorqu'on passe de 0111 à 1000, les 4 bits changent en même temps, ce qui pourra occasionner des états intermédiaires pouvant perturber le fonctionnement d'un système :

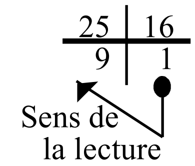
- 0111 soit 7 en décimal
- 1111 code eronné
- 1000 soit 8 en décimal

7- Le système Hexadécimal

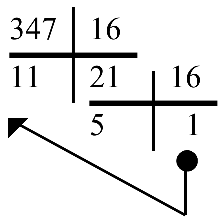
Le système hexadécimal utilise **16 symboles : 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**.
Le **A**$_{(16)}$ correspond au **10**$_{(10)}$, le **B**$_{(16)}$ correspond au **11**$_{(10)}$, etc...(pour garder un seul symbole, on est obligé de passer par les lettres).

Conversion d'un nombre décimal en en nombre hexadécimal

- Exemples **:** convertir $25_{(10)}$ et $347_{(10)}$ en hexadécimal.



Donc **$25_{(10)} = 19_{(16)}$**



En lisant de droit à gauche : (1) (5) (11) ⇒ **$15B_{(16)}$**

- Application : **Chercher** le nombre hexa des nombres décimaux suivants :

N = 40 ⇒	N = 732 ⇒ ....

8- Le codage ASCII

Un ordinateur ne serait pas d'une grande utilité s'il n'était pas capable de traiter l'information non numérique. On veut dire par là qu'un ordinateur doit reconnaître des codes qui correspondent à des nombres, des lettres et des caractères spéciaux. Les codes de ce genre sont dits alphanumériques et le plus connu est appelé **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange (**ASCII**). Un ensemble de caractères complet et acceptable doit renfermer au moins :

- les 26 lettres minuscules,
- les 26 lettres majuscules,
- les dix chiffres,
- environ 25 caractères spéciaux comme +, -, #, %, …

On utilise ainsi environ 87 caractères et pour les représenter il faut au moins 7 bits, car on dispose de 2^7=128 nombres binaires. Ci-dessous une liste partielle du code ASCII.

- Application : un opérateur compose au clavier l'instruction suivante codée en ASCII, **trouver** ce qu'elle signifie.

1010011	1010100	1001111	1010000

Caractère	Code ASCII	Valeur hexadécimale
A	100 0001	41
B	100 0010	42
C	100 0011	43
D	100 0100	44
E	100 0101	45
F	100 0110	46
G	100 0111	47
H	100 1000	48
I	100 1001	49
J	100 1010	4A
K	100 1011	4B
L	100 1100	4C
M	100 1101	4D
N	100 1110	4E
O	100 1111	4F
P	101 0000	50
Q	101 0001	51
R	101 0010	52
S	101 0011	53
T	101 0100	54
U	101 0101	55
V	101 0110	56
W	101 0111	57
X	101 1000	58
Y	101 1001	59
Z	101 1010	5A
0	011 0000	30
1	011 0001	31
2	011 0010	32
3	011 0011	33
4	011 0100	34
5	011 0101	35
6	011 0110	36
7	011 0111	37
8	011 1000	38
9	011 1001	39
Blanc	010 0000	20
.	010 1110	2E
(	010 1000	28
+	010 1011	2B
$	010 0100	24
*	010 1010	2A
)	010 1001	29
-	010 1101	2D
/	010 1111	2F
,	010 1100	2C
=	011 1101	3D