1 - SYSTÈME DE NUMÉRATION :

1.1 - CODE :

Un nombre décimal peut être représenté par son équivalent dans un code différent tel que:

- ✓ le code binaire,
- ✓ le code octal,
- ✓ le code hexadécimal

Un code est un ensemble de règles de représentation de données qui peuvent être

- ✓ numériques,
- ✓ alphabétiques,
- ✓ ou alphanumériques.

Outre le système décimal, les principaux systèmes de numération que l'on utilise dans le domaine du traitement de l'information sont: les systèmes **binaire, octal** et **hexadécimal.**

Lorsqu'un code s'applique à la manière. d'énoncer les nombres, il définît un système de numération.

Exemple :

Le nombre décimal 12 est représenté:

- ✓ *par le nombre 1100 dans le code binaire,*
- ✓ *par le nombre 14 dans le code octal,*
- ✓ *par la lettre C dans le code hexadécimal.*

1.2 - BASE :

Une base B caractérise un système de numération dans lequel tout nombre N peut s'écrire:

$$N = m_n B^n + m_{n-1} B^{n-1} + M_1 B + M_0 B^0$$

avec tous les coefficients *m* < *B*.

Exemples :

- ✓ *Le nombre $341_{(8)}$ en base octale s'écrit:*
  $3 \times 8^2 + 4 \times 8^1 + 1 \times 8^0$
- ✓ *Le nombre $3AF_{(16)}$ en base hexadécimale s'écrit:*
  $3 \times 16^2 + A \times 16^1 + F \times 16^0$

Base	coefficient m
Binaire, B = 2	0, 1
Octale, B = 8	0, 1, 2, 3, 4, 5, 6, 7
Décimale, B = 10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadécimale, B = 16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

1.3 - PONDÉRATION :

Représenter un nombre N_B de n chiffres (ou symboles), dans une base B donnée, consiste en l'écriture en ligne de ces n chiffres de façon telle que:

$$N_B = (\S)_{n-1} \ldots (\S)_i \ldots (\S)_5 (\S)_4 (\S)_3 (\S)_2 (\S)_1 (\S)_0$$

Avec:

- ✓ §: un quelconque des B chiffres ou symboles de la base,
- ✓ n - 1, ... i, 5, 4, 3, 2, 1, 0 indices indiquant le rang ou la position d'ordre du chiffre à partir de la droite.

> La pondération permet l'attribution d'une valeur numérique ou poids à chacun des rangs. Ce poids P dépend de la base dans laquelle est représenté le nombre et a pour valeur:
> $$P = B^{rang}$$

Exemple :

Dans le nombre décimal 425, le chiffre 5 est en position d'ordre 1 ou rang 0, le chiffre 2 en position d'ordre 2 ou rang 1 et le chiffre 4 en position d'ordre 3 ou rang 2.

4	*2*	*5*	*nombre*
2	*1*	*0*	*rang*

Pour la base 10, système décimal:

- ✓ le premier rang ou rang 0 a pour poids 10^0 soit 1, c'est le rang des unités,
- ✓ le rang suivant, rang 1 a pour poids 10^1 soit 10 (rang des dizaines),
- ✓ le rang 2 a pour poids 10^2 soit 100 (rang des centaines),
- ✓ le rang 3 a pour poids 10^3 Soit 1000 (rang des milliers),et ainsi de suite.

Exemple :

Le nombre décimal 2001 est constitué de quatre chiffres:

1 au rang 0 soit	*1 × 1*	*= 1*
0 au rang 1 soit	*0 × 10*	*= 0*
0 au rang 2 soit	*0 × 100*	*= 0*
et 2 au rang 3 soit	*2 ×1000*	*= 2000 ce qui donne un total de : 2001*

Pour la base 2, système binaire:
le premier rang ou rang 0 a pour poids 2^0 soit 1,
le rang 1 a pour poids 2^1 soit 2,
le rang 2 a pour poids 2^2 soit 4,
le rang 3 a pour poids 2^3 soit 8, et ainsi de suite.

Nota :

Dans le système binaire on ne parle plus d'unité, de dizaine ou de centaine mais de bit (contraction de l'anglais binary digit, qui signifie rang binaire). On distingue ainsi le bit 0, le bit 1, le bit 2, le bit 3 ... L'équivalent français de bit est ***élément binaire ou eb,*** *ce terme est relativement peu employé.*

2 - Code binaire pur :

Le code binaire pur ou code binaire naturel est un code pondéré dans lequel les poids sont représentés par les puissances successives de deux.
La valeur décimale du nombre binaire représenté s'obtient directement par addition du poids affecté à chaque bit de valeur 1 .

Nota :
Un groupe de huit bits est appelé octet (en anglais byte of 8 bits). Un groupe de quatre bits est appelé quartet (en anglais byte of 4 bits).

Exemple :
Le nombre binaire 110011 a pour valeur :

$1 \times 2^5 + l \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$ *soit en décimal :*

$32 + 16 + 0 + 0 + 2 + 1 = 51$

2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)	Équivalent décimal $N_{(10)}$
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**
0	0	1	1	2 + 1 ou **3**
0	1	0	0	**4**
0	1	0	1	4 + 1 ou **5**
0	1	1	0	4 + 2 ou **6**
0	1	1	1	4 + 2 + 1 ou **7**

2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)	Équivalent décimal $N_{(10)}$
1	0	0	0	**8**
1	0	0	1	8 + 1 ou **9**
1	0	1	0	8 + 2 ou **10**
1	0	1	1	8 + 2 + 1 ou **11**
1	1	0	0	8 + 4 ou **12**
1	1	0	1	8 + 4 +1 ou **13**
1	1	1	0	8 + 4 + 2 ou **14**
1	1	1	1	8 + 4 +2 +1 ou **15**

3 - Code binaire réfléchi ou code Gray :

Dans le code binaire pur le passage d'une combinaison à l'autre entraîne parfois le changement simultané de plusieurs bits.

c'est par exemple le cas pour la transition de l'équivalent décimal 3 à l'équivalent décimal 4 pour laquelle les bits de poids 1 et 2 passent de 1 à 0 et le bit de poids 4 passe de 0 à 1;

Pour éviter cet inconvénient, cause d'aléas lorsque le code sert à la représentation de grandeurs physiques à variation continue, informations de position par exemple, il est nécessaire d'imaginer des codes pour lesquels le passage d'une combinaison à la suivante n'implique que la modification d'un bit et d'un seul. De tels codes sont appelés "codes réfléchis".

Parmi ceux-ci, le code de Gray est le plus employé.

Nota : Un code réfléchi qui est un code non pondéré ne peut être utilisé pour les opérations arithmétiques.

2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)
0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0
1	0	1	1
1	0	0	1
1	0	0	0

AXES DE SYMÉTRIE OU AXES DE RÉFLEXION

4 - Représentation hexadécimale des nombres binaires :

La représentation hexadécimale ou représentation à base 16 est une notation condensée des nombres binaires. En remarquant que 24 = 16, on peut représenter un octet binaire à l'aide de l'un des 16 symboles du système hexadécimal. Dans ce système les dix premiers symboles sont identiques à ceux utilisés dans le système décimal :

- ✓ **0, 1, 2, 3, 4, 5, 6, 7, 8 et 9,**

et les six derniers correspondent aux premières lettres de l'alphabet latin :

- ✓ **A, B, C, D, E et F,**

lesquelles valent respectivement 10, 11, 12, 13, 14 et 15 en base 10.

Transcodage :

1 - Passage de la Base 2 à la Base 16 :

Pour représenter en hexadécimal un nombre binaire, il suffit de le découper en groupe de quatre bits. Chacun des bits de ces groupes ayant une pondération s'échelonnant de 2^0 à 2^3 , leur somme fournit la valeur hexadécimale de chaque groupe.

Exemple :

Soit le nombre binaire **1011100110101100** *à convertir en hexadécimal.*

Le découpage en quartets de ce nombre donne :

1011 1001 1010 1100

Après pondération, la somme Σ*, bit par bit de chaque groupe de quatre bits est :*

2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)	Σ	
1	0	1	1	11	Soit $B_{(16)}$
1	0	0	1	9	Soit $9_{(16)}$
1	0	1	0	10	Soit $A_{(16)}$
1	1	0	0	12	Soit $C_{(16)}$

Le nombre hexadécimal correspondant au nombre binaire **1011100110101100** *est :* **B9AC**

2 - Passage de la Base 16 à la Base 2 :

Pour convertir en binaire un nombre hexadécimal, il convient de remplacer chacun des symboles hexadécimaux par son quartet binaire équivalent

Exemple :

Les trois quartets binaires équivalents au nombre hexadécimal **8D6** *sont :*

	2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)
8	1	0	0	0
D	1	1	0	1
6	0	1	1	0

Le nombre binaire correspondant au nombre hexadécimal **8D6** *est :* **1000 1101 0110**

5 - REPRÉSENTATION OCTALE DES NOMBRES BINAIRES :

Comme la représentation hexadécimale la représentation octale ou représentation à base 8 est une notation condensée des nombres binaires.

En remarquant que $2^3 = 8$, on peut représenter un triplet binaire à l'aide de l'un des 8 symboles du système octal. Ces huit symboles sont identiques au huit premiers chiffres du système décimal, soit:

0, 1, 2, 3, 4, 5, 6 et 7.

Pour représenter en octal un nombre binaire, il suffit de le découper en groupe de trois bits ou triplet.

Chacun des bits de ces groupes ayant une pondération s'échelonnant de 2^0 à 2^2 leur somme fournit la valeur octale de chaque groupe.

Exemple :

Soit le nombre binaire **110101100** *à convertir en octal. Le découpage en triplets de ce nombre donne:* **110 101 100**

Après pondération, la somme Σ*, bit par bit de chaque groupe est :*

2^2 (4)	2^1 (2)	2^0 (1)	Σ	
1	1	0	6	Soit $6_{(8)}$
1	0	1	5	Soit $5_{(8)}$
1	0	0	4	Soit $4_{(8)}$

Le nombre octal correspondant au nombre binaire **110101100** *est :* **654**

6 - REPRÉSENTATION BINAIRE CODÉE DÉCIMALE :

Dans la représentation binaire codée décimale à chacun des éléments décimaux correspond un quartet représentatif de son équivalent binaire.

Ainsi:

- ✓ au $0_{(10)}$ correspond $0000_{(2)}$,
- ✓ au $1_{(10)}$ correspond $0001_{(2)}$,

- ✓ au $9_{(10)}$ correspond $1001_{(2)}$

Le code Binaire Codé Décimal (BCD) est un code pondéré.

Les poids des bits représentatifs sont:

- ✓ pour le quartet le moins significatif (le premier quartet à droite): **8.4.2.1**.,
- ✓ pour le quartet suivant **80.40.20.10**,
- ✓ **800.400.200.100.** pour le troisième,
- ✓ et ainsi de suite jusqu'au quartet le plus significatif (le dernier quartet à gauche).

Exemple :

Le nombre décimal **$2001_{(10)}$** *devient en BCD :* **0010 0000 0000 0001**

Inversement, le nombre BCD **1001 0101 0001 0111** :

1001	0101	0001	0111
			7
		1	0
	5	0	0
9	0	0	0
9	**5**	**1**	**7**

7 - RÉCAPITULATIF DES MÉTHODES DE CONVERSION :

7.1 - Conversion binaire - décimal :

La méthode consiste à décomposer le nombre en puissances décroissantes de 2 en partant du rang le plus haut, soit :

$1010_{(2)} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
$1010_{(2)} = 8 + 0 + 2 + 0$

$\mathbf{1010_{(2)} = 10_{(10)}}$

7.2 - Conversion décimal - binaire :

On effectue des divisions successives par 2, soit :

$$\begin{array}{r|l} 10 & 2 \\ \hline & 5 \\ & 1 \end{array} \quad \begin{array}{r|l} & 2 \\ \hline & 2 \\ & 0 \end{array} \quad \begin{array}{r|l} & 2 \\ \hline & 1 \\ & 1 \end{array} \quad \begin{array}{r|l} & 2 \\ \hline & 0 \end{array}$$

Sens de Lecture

$\mathbf{10_{(10)} = 1010_{(2)}}$

7.3 - Conversion hexadécimal - décimal :

La méthode consiste à décomposer le nombre en puissances décroissantes de 16 en partant du rang le plus haut, soit:

$BA8_{(16)} = B \times 16^2 + A \times 16^1 + 8 \times 16^0$
$BA8_{(16)} = 11 \times 16^2 + 10 \times 16^1 + 8 \times 16^0$
$BA8_{(16)} = 2816 + 160 + 8$

$\mathbf{BA8_{(16)} = 2984_{(10)}}$

7.4 - Conversion décimal - hexadécimal :

On effectue des divisions successives par 16

$$\begin{array}{r|l} 2984 & 16 \\ \hline 8 & 186 \\ 8 & 10 \\ & A \end{array} \quad \begin{array}{r|l} & 16 \\ \hline & 11 \\ & \\ & B \end{array}$$

Sens de Lecture

$\mathbf{2984_{(10)} = BA8_{(16)}}$

7.5 - Conversion Octal - Décimal :

La méthode consiste à décomposer le nombre en puissances décroissantes de 8 en partant du rang le plus haut, soit:

$777_{(8)} = 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0$
$777_{(8)} = 448 + 56 + 7$

$\mathbf{777_{(8)} = 511_{(10)}}$

7.6 - Conversion Décimal - Octal :

On effectue des divisions successives par 8, soit:

$$\begin{array}{r|l} 511 & 8 \\ \hline 7 & 63 \\ & 7 \end{array} \quad \begin{array}{r|l} & 8 \\ \hline & 7 \\ & 7 \end{array} \quad \begin{array}{r|l} & 8 \\ \hline & 0 \end{array}$$

Sens de Lecture

$\mathbf{511_{(10)} = 777_{(8)}}$

8 - OPÉRATIONS ARITHMÉTIQUES EN BASE 2 :

Les opérations les plus fréquentes en base 2 sont:

- ✓ l'addition,
- ✓ la soustraction.

Ces opérations s'effectuent de la même manière que les opérations décimales en utilisant des tables d'addition et de soustraction beaucoup plus simples.

Tables	
D'addition	De soustraction
0 + 0 = 0 0 + 1 = 1 1 + 0 = 1 1 + 1 =10	0 – 0 = 0 0 – 1 = 10 1 – 0 = 1 1 – 1 = 0
Les éléments entraînant une retenue sont surlignés	

8.1 - ADDITION BINAIRE :

L'addition est l'opération qui consiste à effectuer:

- ✓ dans un premier temps, la somme S_i de deux chiffres binaires de même rang tels que A_i *et* B_i par exemple,
- ✓ puis, dans un second temps, une deuxième somme entre le résultat précédemment obtenu et la valeur du report ou retenue R_{i-1}, issu de l'addition aval de rang i - 1

Exemple :

Effectuer l'addition de deux nombres binaires A et B tels que:

A = 110 *(6 en décimal)*

B = 011 *(3 en décimal)*

Décomposition de la procédure:

- ✓ au premier rang (2^0), la retenue aval est forcément nulle et le total de A. et B. est bien égal à 1,
- ✓ au rang suivant (2^1), la retenue aval est également nulle, le total de A_1 et B_1, est égal à 0 mais génère un report R_1,
- ✓ au troisième rang (2^2) au total de A_2 et B_2 égal à 1 il faut rajouter le report R1 ce qui donne un total définitif de 0 avec un report R2 qui affecte le rang quatre.

	$2^{(3)}$	$2^{(2)}$	$2^{(1)}$	$2^{(0)}$
R	1	1	0	
A		1	1	0
+				
B		0	1	1
S	**1**	**0**	**0**	**1**

Le résultat définitif est donc : **1001** soit 9 en décimal (6 + 3 = 9).