Chapitre II

Système de numération et codage de l'information

☞ **Objectifs:**

- Introduire la numération binaire
- Donner quelques notions d'arithmétique binaire
- Etudier les principaux codes utilisés dans les systèmes numériques.

Chapitre II:

Système de numération et codage de l'information

I- Systèmes de numération:

L'ensemble des outils informatiques sont basés sur les mêmes principes de calcul (loi de tout ou rien). Les calculs habituels sont effectués dans le système de numération décimal, par contre le calculateur électronique ne peut pas utiliser ce système car le circuit électronique ne permet pas de distinguer 10 états. Le système de numération binaire ne comportera que 2 états 0 et 1.

1/ Système décimal: (base 10: 10 éléments de 0 à 9)

- Exemples: $9817 = 9.10^3 + 8.10^2 + 1.10^1 + 7.10^0$

$297,45 = 2.10^2 + 9.10^1 + 7.10^0 + 4.10^{-1} + 5.10^{-2}$

2/ Système binaire: (base 2: 2 éléments 0 et 1 => 2 bits)

☞ Conversion binaire-décimal:

- Exemple n°1: $101101_{(2)}$

Bits	1	0	1	1	0	1
Puissance	2^5	2^4	2^3	2^2	2^1	2^0
Pondération	32	0	8	4	0	1

Somme des pondérations: 32+8+4+1 = 45

donc : $101101_{(2)} = 45_{(10)}$

- Exemple n°2: $1011,011_{(2)}$

Bits	1	0	1	1	0	1	1
Puissance	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
Pondération	8	0	2	1	0	0,25	0,125

Somme des pondérations: 8+2+1+0,25+0,125 = 11,375

donc : $1011,011_{(2)} = 11,375_{(10)}$

☞ Conversion décimal-binaire:

Exemple n°1: $49_{(10)} = ?_{(2)}$	
Quotient	**Reste(/2)**
49	
24	1
12	0
6	0
3	0
1	1
0	1

Donc: $49_{(10)} = 110001_{(2)}$

Exemple n°2 : $0{,}4375_{(10)} = ?_{(2)}$		
2^0	0	‘0’,4375
		* 2
2^{-1}	0	‘0’,8750
		* 2
2^{-2}	1	‘1’,75
		0,75
		* 2
2^{-3}	1	‘1’,5
		0,5
		* 2
2^{-4}	1	‘1’,0

Donc: $0{,}4375_{(10)} = 0{,}0111_{(2)}$

3/ Système octal: (base 8: 8 éléments de 0 à 7)

☞ Conversion octal-décimal:

- Exemple: $476_{(8)} = 4.8^2+7.8^1+6.8^0 = 256+56+6 = 318_{(10)}$

☞ Conversion décimal-octal:

On a 2 méthodes: « passage par la base 2 » ou « divisions successives par 8 »

Octal	**Binaire**
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

- Exemple: $928_{(10)} = ?_{(8)}$

928(10) = 1 110 100 000$_{(2)}$ = 1 6 4 0$_{(8)}$

Ou bien:

Quotient	**Reste (/8)**
928	
116	0
14	4
1	6
0	1

Donc: $928_{(10)} = 1640_{(8)}$

Vérification: $1640_{(8)} = 1.8^3+6.8^2+4.8^1+0.8^0 = 512+384+32 = 928_{(10)}$

4/ Système hexadécimal: (base 16: 16 éléments, 0..9,A,B,C,D,E et F)

☞ Conversion hexadécimal-décimal:

- Exemple: $4CA2_{(16)} = ?_{(10)}$

$4CA2_{(16)} = 4.16^3+12.16^2+10.16^1+2.16^0 = 16384+3072+160+2 = 19618_{(10)}$

☞ Conversion décimal-hexadécimal:

On a 2 méthodes: « passage par la base 2 » ou « divisions successives par 16 »

Base 10	Base 16	Base 2
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

- Exemple: $469_{(10)} = ?_{(16)}$

Quotient	Reste (/16)
469	
29	5
1	13 (D)
0	1

Donc: $469_{(10)} = 1D5_{(16)}$

☞ Conversion hexadécimal-octal: (Passage par la base 2)

- Exemple: $AF9,D1_{(16)} = ?_{(10)}$

$AF9,D1_{(16)}$ $= 1010\ 1111\ 1001,1101\ 0001_{(2)}$

$= 101\ 011\ 111\ 001,110\ 100\ 010_{(2)}$

$= 5\ 3\ 7\ 1,6\ 4\ 2_{(8)}$

II- Notions de codage:

Un dispositif logique ou numérique est destiné à manipuler des informations diverses qui doivent être traduites par un ensemble de 0 et 1, obtenu suivant une loi de correspondance préétablie: c'est l'opération de codage de l'information.

$$\textbf{Information} \xrightarrow{\textbf{Codage (loi de correspondance)}} \textbf{\{Configuration binaire\}}$$

☞ **Exemples de codes:**

* *Code ASCII:* chaque touche du clavier est codée sur 8 bits, donc on peut coder 256 caractères. Exemple: Touche 'A' ==> code ASCII « 01000001 » ??

* *Code DCB* (Décimal Codée en Binaire): utilisé uniquement pour les chiffres décimaux. Ce code est obtenu en remplaçant individuellement chacun des chiffres du nombre à représenter par son équivalent binaire pur.

Exemple: $2458_{(10)} = 0010\ 0100\ 0101\ 1000_{(DCB)}$

- Avantages: Représentation plus simple et très utile pour les systèmes d'affichage à 7 segments.

- Inconvénient: nécessite plus de bits.