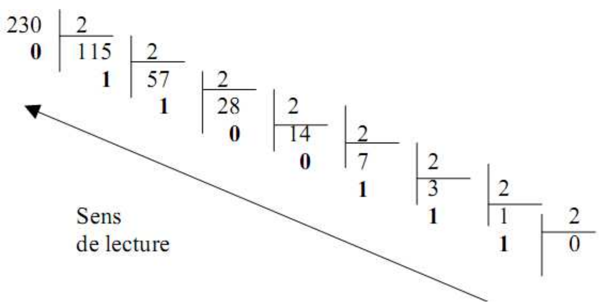
S9 – AUTOMATISMES ET INFORMATIQUE INDUSTRIELLE

Fascicule 7

Numération

S9



Numération

1) Introduction

Le langage et la communication sont les éléments indispensables de notre société avec toujours plus de rapidité et toujours plus de fiabilité. L'informatique a bouleversé ces concepts en imposants des règles strictes mais efficace.

Nous allons étudier quelques systèmes de numération couramment utilisés dont les subtilités nous échappent tant nous sommes habitués à leur utilisation.

2) Rappels :

2-1) La base :

Elle est représentée par le nombre de caractères différents qu'utilise le système pour représenter les nombres.

- Système décimal : 0,1,2,3,4,5,6,7,8,9
- Système Octal : 0,1,2,3,4,5,6,7
- Système binaire : 0,1
- Système Hexadécimale : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

On précisera la valeur de cette base en indice. A défaut, c'est la base décimal qui prime.

Exemples : 706 en base 8 sera écrit $706_{(8)}$
1001 en base 2 sera écrit $1001_{(2)}$

2-2) Systèmes de numération :

- Système décimal :

La conversion d'un nombre dans un système de numération vers le système décimal est toujours la même. Pour retrouver le nombre décimal, il suffit d'additionner les monômes représentés chacun par le chiffre appartenant au système de numération multiplié par la puissance de la base correspondant au rang de ce chiffre.

- Conversion binaire → décimal :
  $100101_{(2)} = 1*2^5+0*2^4+0*2^3+1*2^2+0*2^1+1*2^0 = 32 + 0 + 0 + 4 + 1 = 37$

- Conversion octal → décimal :
  $7036_{(8)} = 7*8^3+0*8^2+3*8^1+6*8^0 = 3584 + 0 + 24 + 6 = 3614$

- Conversion hexadecimal → décimal :

  2C5A$_{(16)}$ = $2*16^3+12*16^2+5*16^1+10*16^0$ = 8192 + 3072 + 80 + 10 = 11354

➢ Système binaire :

C'est la base de numération couramment utilisée en électronique et informatique. On utilise des 1 et 0 pour constituer les nombres. Chacun de ces caractères est appelé bit.

On a donc 0+0=0, 1+0=0+1=1 et 1+1=10

La correspondance est donc la suivante :

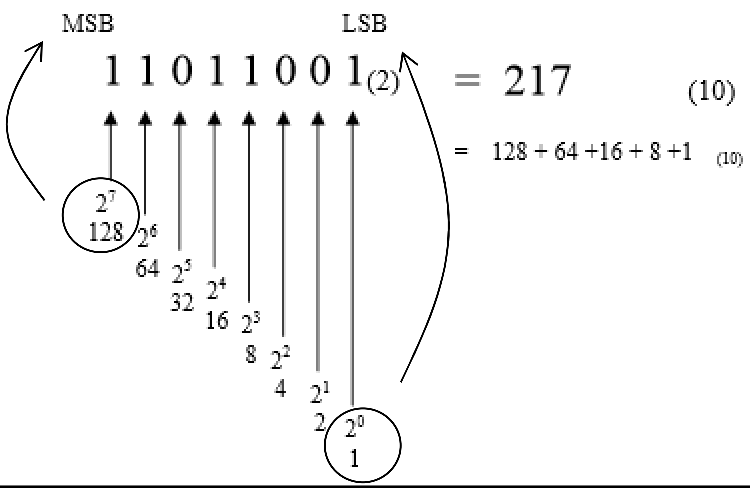
Base 10	Base 2
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Retenue

Un nombre binaire comporte un nombre de bits plus ou moins normalisé :

- 1 bit pour indiquer un état actif ou non d'une variable
- 4 bits représentent un Quartet
- 8 bits représentent un Octet (Byte en anglais)
- 16 bits sont appelés Word (Intel) ou Double Byte (Motorola)
- 32 bits
- 64 bits

Lorsque l'on écrit un nombre binaire on distingue les poids fort (MSB pour Most Significant Bit) et faible (LSB pour Less Significant Bit)

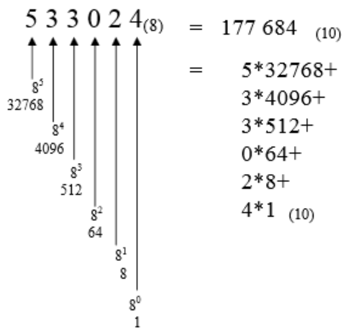


- Système octal

Ce système à base 8 s'est imposé en électronique numérique pendant de nombreuses années, mais la base hexadécimale a pris le pas, et la base octale est donc en voie d'extinction, cependant on peut le retrouver sur de très vieux systèmes informatiques.

Base 10	Base 8
0	00
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	10
9	11
10	12

Retenue

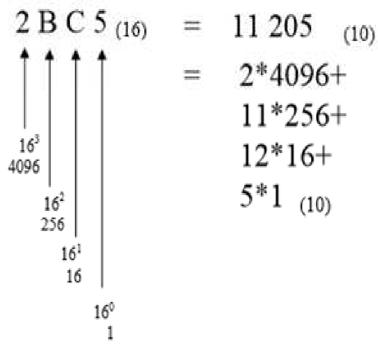


- Système hexadécimal :

Ce système à base 16 est le plus utilisé en électronique numérique car il permet une manipulation de quartets (mot binaire de 4 bits) en représentation compacte. Ce qui, dans les systèmes actuels à grande capacité mémoire par exemple, est un avantage non négligeable. La base 16 est une forme contractée de la base 2.

Base 10	Base 16
0	00
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
10	0A
11	0B
12	0C
13	0D
14	0E
15	0F
16	10
17	11

Retenue



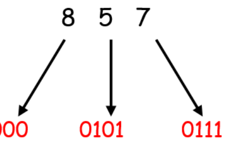
Numération

- Système BCD : Décimal Code Binaire :

Ce code conserve les avantages du système binaire naturel et du système décimal. Chaque chiffre du code décimal est représenté par un quartet binaire, mais on compte en base 10, ce qui veut dire que la valeur la plus élevée dans un quartet est 9 $_{(10)}$ =1001$_{(2)}$

Codage du chiffre 857 :

Codage du chiffre 857 :



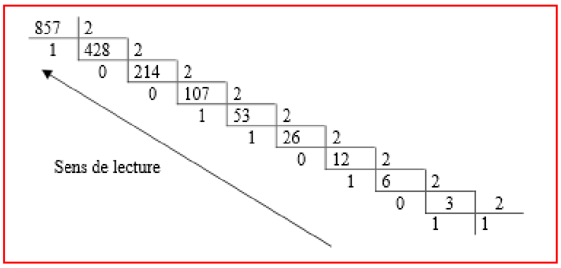
Donc 857$_{(10)}$ = 1000 0101 0111$_{(BCD)}$

2-3) Conversion des systèmes :

Le passage du binaire vers décimal a été vu précédemment et ne pose pas de problème. La transformation inverse est plus délicate.

- Conversion décimal →binaire :

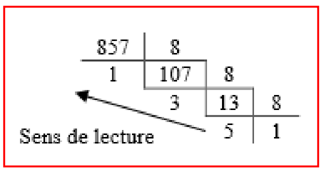
On procède à une série de division par deux jusqu'à obtenir une résultat de 1 ou 0 :



On a donc 857$_{(10)}$ = 11 0101 1001$_{(2)}$

- Conversion décimal →Octal :

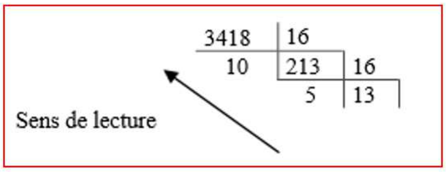
Le principe est le même que celui vu précédemment, c'est à dire que l'on divise le nombre décimal par la base 8 , jusqu'à ce que l'on obtienne un résultat inférieur à la base



On a donc 857$_{(10)}$ = 1531$_{(8)}$

- Conversion décimal →Hexadécimal :

  - 1ère méthode :
    Le principe est le même que celui vu précédemment, c'est à dire que l'on divise le nombre décimal par la base 16, jusqu'à ce que l'on obtienne un résultat inférieur à la base.



On a donc $3418_{(10)}=D5A_{(16)}$

  - 2ème méthode :
    Plus couramment utilisée du fait que les nombres sont déjà écrit en binaire dans les systèmes numériques, consiste à effectuer une conversion en base 2 (binaire) du nombre, puis de convertir chaque quartet obtenu en hexadécimal

$3418_{(10)} = 1101\ 0101\ 1010_{(2)} = D5A_{(16)}$

$D_{(16)}$ $5_{(16)}$ $A_{(16)}$

- Conversion Binaire →Hexadécimal :

  Il suffit de regrouper les bits par quartet et trouver l'équivalent hexadécimal de chaque quartet.

  $1001\ 1110_{(2)} = 9E_{(16)}$

- Conversion Hexadécimal→Binaire :
  Il suffit de remplacer chaque symbole hexadécimal du nombre par son équivalent binaire.

  $BA_{(16)} = 1011\ 1010_{(2)}$

3) Opérations arithmétiques en binaires :

- Addition :

On se rappelle les règles d'addition :

0 + 0 = 0
1 + 1 = 0 report de 1
1 + 0 = 1
1 + 1 + 1 = 1 report de 1

Exemple :

```
 1   1 1   1 1
  1100110011
+ 1001100110
 ------------
 10110011001
```

- Soustraction

Règles de soustraction :

0 – 0 = 0
1 – 1 = 0
²1 – 0 = 1
0 – 1 = 1 retenue 1

Exemple :

```
  1100110001
- 1001100110
  ----------
  0011001011
```

4) Multiplication en base deux

La multiplication s'effectuera de la même manière qu'en base dix

Exemple :

		1	0	1	1	1	0
	x				1	0	1
		1	0	1	1	1	0
1	0	1	1	1	0	0	
1	1	1	0	0	1	1	0

5) Divisions en base deux

La division s'effectuera de la même manière qu'en base dix

Exemple :

1	1	1	0	0	1	1	1		1	0	1			
1	0	1							1	0	1	1	1	0
0	1	0	0	0										
	-	1	0	1										
	0	0	1	1	1									
			1	0	1									
			0	1	0	1								
				1	0	1								
				0	0	0	1							