Codage de l’information

1.1 Système binaire

Le système binaire est un système qui comprend deux chiffres : 0 et 1. Tous les nombres sont formés par ces chiffres. De ce fait, le système a pour base 2.

Dans les domaines de l'automatisme, de l'électronique, de l'informatique et de télécommunications, nous utilisons la base 2.

Exemple : Un interrupteur est ouvert ou fermé, Une diode est allumée ou éteinte, Une tension est présente ou absente. A chaque état on associe un état logique soit 0 ou 1.

Le chiffre binaire qui peut prendre ces deux états est nommé ***Bit*** (*Binary digit*).

<table>
<tr><td>Avec 2 bits nous pouvons coder 4 états</td><td>
<table>
<tr><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Avec 3 bits nous pouvons coder 8 états</td><td>
<table>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

A chaque nouveau **bit**, le nombre de combinaisons possibles est doublé. Ce nombre est égal à 2^N (N étant le nombre de bits).

Note : définition d’un **Octet**

Un groupe de bits est appelé un mot, un mot de huit bits est nommé un **octet** (*byte*).

0	1	1	0	1	0	1	1

Avec un octet, nous pouvons écrire 2^8 = 256 nombres binaires, soit de 00000000 (**0**) à 11111111 (**255**).

1.2 Système hexadécimal

Le système hexadécimal est le système le plus utilisé par l'homme en informatique car il permet l'interprétation rapide d'une valeur binaire.

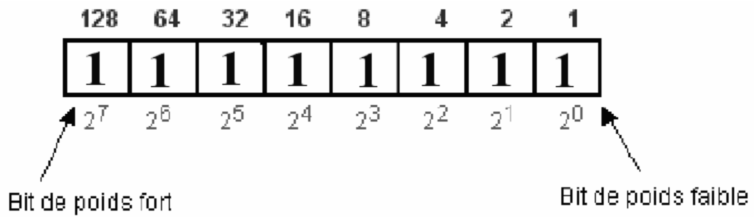
C’est un système de numération positionnel en base 16, utilise les dix premiers chiffres et les 6 premières lettres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

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

2 Conversion entre les systèmes

2.1 Conversion binaire en décimal

Dans un nombre binaire, la valeur d'un bit, appelée poids, dépend de la position du bit en partant de la droite. A la manière des dizaines, des centaines et des milliers pour un nombre décimal, le poids d'un bit croît d'une puissance de deux en allant de la droite vers la gauche comme le montre la figure suivante :



Il suffit donc de faire la somme des poids de chaque bit à 1.

Le nombre ci-dessus est égal à 128 + 64 + 32 + 16 + 4 + 2 + 1 = 255

Exemple :

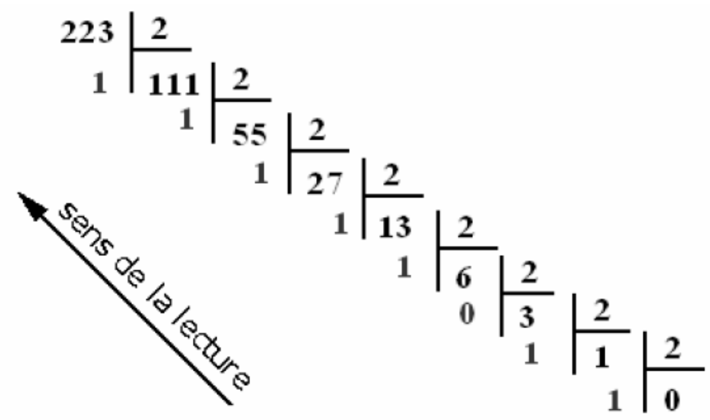
$(\mathbf{1010})_2 = \mathbf{1} \times 2^3 + \mathbf{0} \times 2^2 + \mathbf{1} \times 2^1 + \mathbf{0} \times 2^0$ $(\mathbf{1010})_2 = \mathbf{1} \times 8 + \mathbf{0} \times 4 + \mathbf{1} \times 2 + \mathbf{0} \times 1$ donc : $(\mathbf{1010})_2 = (\mathbf{10})_{10}$

2.2 Conversion décimal en binaire

Pour obtenir l'expression binaire d'un nombre exprimé en décimal, il suffit de diviser successivement ce nombre par 2 jusqu'à ce que le quotient obtenu soit égal à 0. Le nombre cherché est donné par les restes successifs des divisions pris du bas vers en haut.

Exemple : Conversion d'un nombre décimal 223 en binaire

$(223)_{10} = (11011111)_2$



$(\mathbf{223})_{10} = (\mathbf{11011111})_2$

2.3 Conversion hexadécimal en décimal

Exemple : Convertir le nombre $(5AC)_{16}$ en décimal.

Le nombre $(\mathbf{5AC})_{16}$ peut se décomposer comme suit : $(\mathbf{5AC})_{16} = \mathbf{5} \times 16^2 + \mathbf{A} \times 16^1 + \mathbf{C} \times 16^0$

En remplaçant A et C par leur équivalent en base 10, on obtient : $(\mathbf{5AC})_{16} = \mathbf{5} \times 16^2 + \mathbf{10} \times 16^1 + \mathbf{12} \times 16^0$

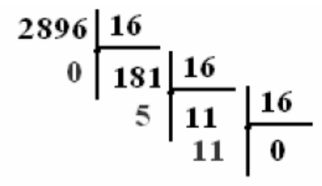
$(\mathbf{5AC})_{16} = \mathbf{5} \times 256 + \mathbf{10} \times 16 + \mathbf{12} \times 1$

Donc : $(\mathbf{5AC})_{16} = (\mathbf{1452})_{10}$

2.4 Conversion décimal en hexadécimal

Pour convertir un nombre décimal en hexadécimal, il suffit d'effectuer des divisions entières par 16 comme en binaire.

Exemple :



$(\mathbf{2896})_{10} = (\mathbf{B50})_{16}$ (11 correspond à la lettre B).

2.5 Conversion binaire en hexadécimal

Pour convertir du binaire en hexadécimal, il suffit de faire correspondre un mot de quatre bits à chaque chiffre hexadécimal en utilisant le tableau suivant :

0000 = 0	0100 = 4	1000 = 8	1100 = C
0001 = 1	0101 = 5	1001 = 9	1101 = D
0010 =2	0110 = 6	1010 = A	1110 = E
0011 = 3	0111 = 7	1011 = B	1111 = F

Tableau de conversion binaire en hexadécimal

Exemple : Convertir 1111110010110001 en hexadécimal.

$(1111110010110001)_2 = (FCB1)_{16}$

Note : $(111010)_2 = (00111010)_2 = (3A)_{16}$

2.6 Conversion hexadécimal en binaire

Cela consiste à associer à chaque caractère du nombre hexadécimal son équivalent binaire sur 4 bits.

Exemple : $(1FA2)_{16} = (0001\ 1111\ 1010\ OO1O)_2$

3 Les opérations en binaire.

Le fait de travailler en base 2 ne change rien aux règles profondes qui lient les nombres. On peut donc dire qu'en binaire, les opérateurs existant en base 10 s'appliquent avec les mêmes règles.

3.1 L'addition.

L'addition en binaire garde les mêmes règles que l'addition en décimal, on y retrouve les propriétés de commutativité et d'associativité des additions décimales.

Mais regardons les additions d'un bit par un autre.

0	0	1	1
+ 0	+ 1	+ 0	+ 1
0	1	1	1 0
			Retenue ↑

Exemple

Retenue		1	1		
			1	1	0
	+			1	0
Somme		1	0	0	0

1	1	1		Retenue
	1	0	1	
+		1	1	
1	0	0	0	Somme

3.2 La Multiplication

La multiplication binaire se réalise comme une multiplication décimale.

Multiplication décimale

		2	3
x		1	1
		2	3
+	2	3	-
	2	5	3

Multiplication binaire

								1	0	1	1	0	1	0
x								1	1	0	0	0	0	1
								1	0	1	1	0	1	0
+							0	0	0	0	0	0	0	-
+						0	0	0	0	0	0	0	-	-
+					0	0	0	0	0	0	0	-	-	-
+				0	0	0	0	0	0	0	-	-	-	-
+			1	0	1	1	0	1	0	-	-	-	-	-
+		1	0	1	1	0	1	0	-	-	-	-	-	-
	1	0	0	0	1	0	0	0	0	1	1	0	1	0

3.3 La soustraction

Encore une fois, la soustraction binaire est identique à sa sœur décimale, tant au niveau des propriétés que de la méthode de calcul.

0	0	1	1
- 0	- 1	- 0	- 1
0	1 0	1	0
	Retenue ↑		

Soustraction décimale

	4	2	9
-	3	5	8
retenues	1		
	0	7	1

soustraction binaire

	1	1	0	1	0	1	1	0	1
-	1	0	1	1	0	0	1	1	0
retenues		1				1	1		
	0	0	1	0	0	0	1	1	1

Conversions

1.1. Décimal vers binaire
Convertir en binaire (base 2) les nombres suivants, en détaillant les calculs :
$(12)_{10}$, $(99)_{10}$, $(421)_{10}$. $(354)_{10}$, $(436)_{10}$, $(1256)_{10}$.

1.2. Binaire vers décimal
Convertir en décimal (base 10) les nombres suivants : $(1001)_2$, $(11101)_2$, $(1101110)_2$.

1.3. Décimal vers hexadécimal
Convertir en hexadécimal (base 16) les nombres suivants : $(12)_{10}$, $(125)_{10}$, $(3247)_{10}$.

1.4. Hexadécimal vers décimal
Convertir en décimal (base 10) les nombres suivants : $(5BC)_{16}$, $(FFF)_{16}$, $(6AF)_{16}$.

Opérations

2.1. Additions Additionner ces nombres binaires et donner le résultat en base 2.
1100 + 1000 1001 + 1010 11001111 + 11100110
Comment vérifier le résultat en faisant le calcul d'une autre façon ?

2.2. Soustractions Mêmes questions avec des soustractions :
1010 – 011 1110 0110 - 1100 1010

2.3. Multiplications Mêmes questions avec des multiplications :
11 1010 × 110 10011 × 0110

2.4. Divisions Mêmes questions avec des divisions entières :
1111 0101 ÷ 1101 100 1000 0111 ÷ 1011

Quelques calculs de synthèse

3.1. Conversions
1. $(88)_{10}=(?)_2$ $(1000\ 1010)_2=(?)_{10}$ $(A1A)_{16}=(?)_{10}$ $(165)_{10}=(?)_{16}$ $(EB5A)_{16}=(?)_2=(?)_{10}$

3.2. Opérations Effectuer les calculs suivants sachant que, en binaire :
A = 11 1100, B = 1 110, C = 101, D = 1 1111. A+B+D A-B B×C A÷C

Conversions

1.1. Décimal vers binaire
Convertir en binaire (base 2) les nombres suivants, en détaillant les calculs :
$(12)_{10}$, $(99)_{10}$, $(421)_{10}$. $(354)_{10}$, $(436)_{10}$, $(1256)_{10}$.

1.2. Binaire vers décimal
Convertir en décimal (base 10) les nombres suivants : $(1001)_2$, $(11101)_2$, $(1101110)_2$.

1.3. Décimal vers hexadécimal
Convertir en hexadécimal (base 16) les nombres suivants : $(12)_{10}$, $(125)_{10}$, $(3247)_{10}$.

1.4. Hexadécimal vers décimal
Convertir en décimal (base 10) les nombres suivants : $(5BC)_{16}$, $(FFF)_{16}$, $(6AF)_{16}$.

Opérations

2.1. Additions Additionner ces nombres binaires et donner le résultat en base 2.
1100 + 1000 1001 + 1010 11001111 + 11100110
Comment vérifier le résultat en faisant le calcul d'une autre façon ?

2.2. Soustractions Mêmes questions avec des soustractions :
1010 – 011 1110 0110 - 1100 1010

2.3. Multiplications Mêmes questions avec des multiplications :
11 1010 × 110 10011 × 0110

2.4. Divisions Mêmes questions avec des divisions entières :
1111 0101 ÷ 1101 100 1000 0111 ÷ 1011

Quelques calculs de synthèse

3.1. Conversions
1. $(88)_{10}=(?)_2$ $(1000\ 1010)_2=(?)_{10}$ $(A1A)_{16}=(?)_{10}$ $(165)_{10}=(?)_{16}$ $(EB5A)_{16}=(?)_2=(?)_{10}$

3.2. Opérations Effectuer les calculs suivants sachant que, en binaire :
A = 11 1100, B = 1 110, C = 101, D = 1 1111. A+B+D A-B B×C A÷C