L'unités :

Un ordinateur manipule des nombres binaires sous forme de 8 bits appelés Octet :

Kilo-octets (ko)=2^{10} octes=1024 octets

Mega-octets(Mo)=2^{20} octes =1024 ko

Giga-octets(Go)=2^{30} octets=1024 Mo

Terra-octets(To)=2^{40} octets=1024 Go

Peta-octets(Po)=2^{50} octets=1024 To

Mot binaire

Un mot binaire de n digits est un ensemble ordonné de n variables binaires.

Digit ou Binary digit (BIT) = nombre valant 0 ou 1

Pour n variables binaires => 2^n combinaisons possibles, c'est à dire 2^n écritures différentes de mots.

Exemple :

Avec n = 3

on a $2^3 = 8$ combinaisons :

« 000, 001, 010, 011, 100, 101, 110, 111 »

1.2. Numération

La numération permet de représenter un mot (ou nombre) par la juxtaposition ordonnée de variables (ou symboles) pris parmi un ensemble. Connaître la numération revient à connaître le mécanisme qui permet de passer d'un mot à un autre (comptage, opération)

❖ Les systèmes de numération les plus courants sont :

- **Système décimal :** il comprend **10** symboles appelés chiffres : {**0,1,2,3,4,5,6,7,8,9**}
- **Système binaire :** il comprend **2** symboles appelés BIT (Binary digIT) : **0** et **1**
- **Système octal :** il utilise **8** symboles qui sont les chiffres de **0 à 7**,
- **Système hexadécimal : 16** symboles : Les chiffres de **0 à 9** et les lettres A**, B, C, D, E, F**

❖ La **Base** : le nombre de symboles que possède le système de numération

- Lorsqu'un mot (ou nombre) est écrit, la position respective des symboles détermine leurs poids.

Exemple 1.1 :

359,02 : nombre du système décimal système à base 10
Composé de 5 symboles

Ecriture : $359,02 = 3x100 + 5x10 + 9x1 + 0x0,1 + 2x0,01$

Ou encore : $359,02 = 3x10^{2} + 5x10^{1} + 9x10^{0} + 0x10^{-1} + 2x10^{-2}$

3 => chiffre du poids fort
2 => chiffre du poids faible
Les poids des rangs sont des puissances de 10

1.3. Codage

Définition :

Le codage est l'opération de transformation de l'écriture d'un nombre décimal dans une base **B** quelconque.

Ecriture dans une base B :

Tout entier décimal **N** peut s'écrire dans une base **B** quelconque.

Un nombre **N** s'écrit en juxtaposant n symboles :

$$N = (a_{n-1}\, a_{n-2} \ldots a_0)_B \quad 0 \leq a_i \leq B-1$$

Ce nombre N a pour valeur décimale :

$$N = a_{n-1}.B^{n-1} + a_{n-2}.B^{n-2} + \ldots + a_1.B^1 + a_0.\, B^0$$

$$0 \leq N \leq B^n-1$$

Ecriture dans une base B :

$N = a_{n-1}.B^{n-1} + a_{n-2}.B^{n-2} + \dots + a_1.B^1 + a_0.B^0$; $0 \leq N \leq B^{n-1}$

- Cette forme est appelée forme polynomiale.
- L'élément a_i est le symbole de rang i et son poids est B^i.
- a_{n-1} est le symbole le plus significatif (de poids le plus fort)
- a_0 est le symbole le moins significatif (de poids le plus faible)
- Les termes **B^i** sont appelés **coefficients de pondération** ou Poids
- N est codé sur n bits

Codages courants :

Codage	B	a_i
Binaire	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 7
Hexadécimal	16	0, 1, 2, 3, 4, 5, 7, 8, 9, A, B, C, D, E, F

1.4. Codage binaire (B = 2)

- Système ayant une base 2 : disposant de 2 symboles {0,1}
- Codage binaire se fait généralement en utilisant un nombre fixe de bits :

  Format de 4 bits, de 8 bits, de 16 bits, de 32 bits…

  Ces formats sont des puissances de 2.
- Format de n bits : permet de représenter tous les nombre entiers N compris entre 0 et 2^n-1.
- Format de n bits : on parle alors de mot de n bits.

Exemple :

$(1011)_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0$ $(1011)_2 = 1x8 +$

$0x4 + 1x2 + 1x1 = (11)_{10}$

-Le nombre binaire **1011** représente le nombre décimal **11**

-le chiffre **1 à gauche** représente le bit de **poids fort**

- le chiffre **1 à droite** représente le bit de **poids faible**

- Les poids des rangs sont des puissances de **2**

- On dit que le mot de 4 bits est pondéré **8-4-2-1**

Comptage binaire

Si on utilise 4 bits pour coder les nombre décimaux :

- Il existe 16 combinaisons (2^4=16)
- On peut alors compter de 0 à 2^4 - 1 = 15
- Le bit de poids faible (celui de droite) sera pondéré 2^0 = 1
- Le bit de poids fort (celui de gauche) sera pondéré 2^3 = 8

Tableau de la suite des nombres binaires :

N	a_3	a_2	a_1	a_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1

N	a_3	a_2	a_1	a_0
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Méthode de codage d'un nombre N en binaire **<u>Méthode 1 :</u>**

- On sait que $0 \leq N \leq B^n-1$ et $B = 2$

- On détermine le nombre n de bits minimum pour coder N.

- On positionne les bits a_i à 0 ou 1 de telle façon que $a_{n-1}.2^{n-1} + a_{n-2}.2^{n-2} + \ldots + a_1.2^1 + a_0.2^0 = N$

Méthode 1 :

Exemple : On veut coder 18

- Codage sur 1 bit (n = 1) : on compte de 0 à 2^1-1 = 1,
- Codage sur 2 bits (n = 2) : on compte de 0 à 2^2-1 = 3,
- Codage sur 3 bits (n = 3) : on compte de 0 à 2^3-1 = 7,
- Codage sur 4 bits (n = 4) : on compte de 0 à 2^4-1 = 15,
- Codage sur 5 bits (n = 5) : on compte de 0 à 2^5-1 = 31,

=> Il faut 5 bits pour coder le nombre 18 :

16	8	4	2	1
1	0	0	1	0

18 = $(10010)_2$

<u>Méthode 2 :</u>

Méthode des divisions euclidiennes successives

Soit $N = a_{n-1}.B^{n-1} + a_{n-2}.B^{n-2} + \ldots + a_1.B^1 + a_0.\ B^0$ avec B = 2

Effectuons les divisions euclidiennes successives de N par 2 jusqu'à ce que le quotient devienne nul :

$$(a_{n-1}.2^{n-1} + a_{n-2}.2^{n-2} + \ldots + a_1.2 + a_0) = \underbrace{(a_{n-1}.2^{n-2} + a_{n-2}.2^{n-3} + \ldots + a_1)} \times 2 + a_0$$

$$\overbrace{(a_{n-1}.2^{n-2} + a_{n-2}.2^{n-3} + \ldots + a_1)} = \underbrace{(a_{n-1}.2^{n-3} + a_{n-2}.2^{n-4} + \ldots + a_2)} \times 2 + a_1$$

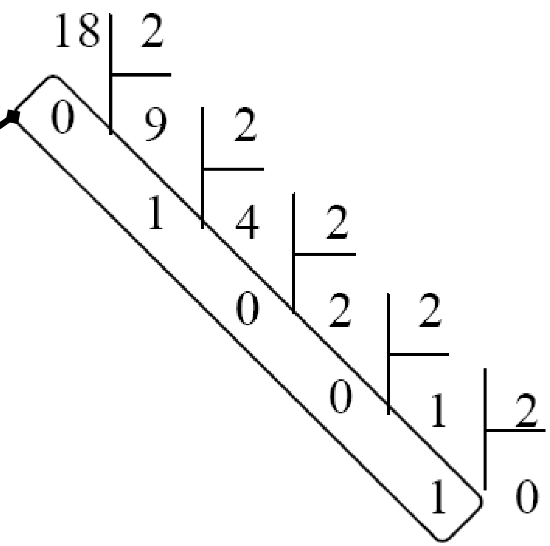
$$\vdots$$

$$\overbrace{(a_{n-1})} = (0) \times 2 + a_{n-1}$$

Méthode 2 :

Les restes des divisions successives forment le mot binaire du poids le plus faible au poids le plus fort.

Exemple : On veut coder 18



18 = $(10010)_2$

1.5. Codage octal (B = 8)

- Système ayant une base 8 : disposant de 8 symboles {0 à 7}
- Format de n chiffres : permet de représenter tous les nombre entiers décimaux compris entre 0 et 8^n-1.

Exemple :

Mot octal de 3 chiffres. Il sera pondéré 8^2-8^1-80 soit : 64-8-1 (721)8 =

$7 \times 8^2 + 2 \times 8^1 + 1 \times 8^0$

(721)8 = 7×64 + 2×8 + 1×1 = $(465)_{10}$

Le nombre $(721)_8$ représente le nombre décimal 465

Codage d'un nombre N en octal :

Comme pour la conversion en binaire, on effectue ici la méthode des divisions euclidiennes successives par 8

Exemple :

convertir le nombre décimal 99 en octal

$99 = 8 \times 12 + 3$ (chiffre de poids faible)

$12 = 8 \times 1 + 4$

$1 = 8 \times 0 + 1$ (chiffre de poids fort)

$\Rightarrow \quad 99 = (143)_8$

Passage du Binaire en Octal (Encodage) :

Il suffit de faire des regroupements de 3 bits sur le mot binaire.
Un mot binaire de 3 bits permet de coder les nombres entiers décimaux compris entre 0 et 7:

$$N = \underbrace{a_{n-1}.2^{n-1} + a_{n-2}.2^{n-2} + a_{n-3}.2^{n-3}}_{a'_{n-1}.8^{n-1}} + \ldots + \underbrace{a_3.2^3}_{a'_1.8^1} + \underbrace{a_2.2^2 + a_1.2^1 + a_0.2^0}_{a'_0.8^0}$$

$$N = a'_{n-1}.8^{n-1} + \ldots \quad a'_1.8^1 + a'_0.8^0$$

Exemple :

Soit le nombre binaire : 1110111

1 110 111 = 001 110 111

1 6 7

$(1110111)_2 = (167)_8$

1.6. Codage hexadécimal (B = 16)

- Système ayant une base 16 : disposant de 16 symboles {0 à 9 et de A (10) à F (15)}

- Format de n chiffres : permet de représenter tous les nombre entiers décimaux compris entre 0 et 16^n-1.

Voici l'équivalence des codages dans les 2 numérations (décimal, hexadécimale) :

N	Héxa	N	Héxa	N	Héxa	N	Héxa	N	Héxa
0	0	**5**	5	**10**	A	**15**	F	**20**	14
1	1	**6**	6	**11**	B	**16**	10	**21**	15
2	2	**7**	7	**12**	C	**17**	11	**26**	1A
3	3	**8**	8	**13**	D	**18**	12	**32**	20
4	4	**9**	9	**14**	E	**19**	13	**100**	64

Exemple :

Mot hexadécimal de 4 symboles. Il sera pondéré 16^3-16^2-16^1-16^0 soit : 4096-256-16-1

$$(20AC)_{16} = 2\times16^3 + 0\times16^2 + A\times16^1 + C\times16^0$$

$$(20AC)_{16} = 2\times4096 + 0\times256 + 10\times16 + 12\times1 = (8364)_{10}$$

Le nombre $(20AC)_{16}$ représente le nombre décimal 8364

Codage d'un nombre N en octal :

Comme pour la conversion en binaire, on effectue ici la méthode des divisions euclidiennes successives par 16.

Exemple : convertir le nombre décimal 92 en hexadécimal

92 = 16 × 5 + 12 (chiffre de poids faible = C)

5 = 16 × 0 + 5 (chiffre de poids fort) 92 = $(5C)_{16}$

Passage du Binaire en Hexadécimal (Encodage) :

Il suffit de faire des regroupements de 4 bits sur le mot binaire.
Un mot binaire de 4 bits permet de coder les nombres entiers décimaux compris entre 0 et 15.

Exemple :

Soit le nombre binaire : 1110111 111 0111 = 0111 0111

7 7

$$(1110111)_2 = (77)_{16}$$

Les opérations en binaire :

a. L'addition

On procède comme en décimal. Quand le résultat de la somme d'une colonne est supérieure à 1
(utilise plus de 1 bit), on passe ce bit au voisin de gauche.

Exemple : 1011 + 1001

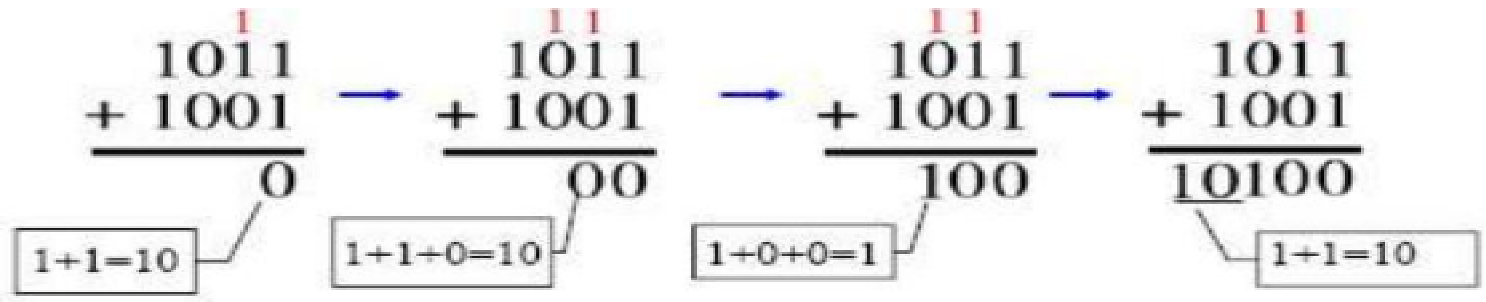
0+0=0

0+1=1

1+0=1

1+1=0 **Retenir 1**

1+1+1=1 **Retenir 1**



Les opérations en binaire :

Exercice :

Effectuer les additions suivantes :

1100 + 0011=
1111 + 0101=

Les opérations en binaire :

b. La soustraction

Dans la soustraction binaire, on peut procéder comme en décimal :

- Quand la quantité à soustraire est supérieure à la quantité dont on soustrait, on « emprunte » 1 au voisin de gauche.
- En binaire, le « 1 » emprunté va ajouter « 2 » à la quantité dont on soustrait, tandis qu'en décimal il ajoute « 10 ».

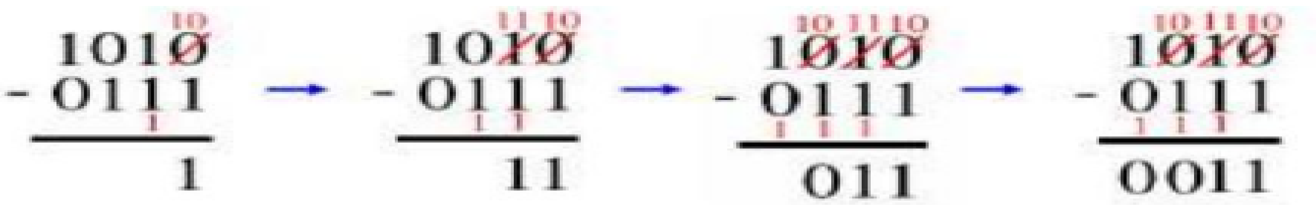
Les opérations en binaire :

b. La soustraction

.

- 0-0=0
- 1-0=1
- 1-1=0
- 0-1=1 emprunt

Exemple : 1010 – 0111



Les opérations en binaire :

Exercice :

Effectuer les soustractions suivantes:

1111 - 0101=

1100 - 0011=

Les opérations en binaire :

c. La multiplication

Dans la multiplication binaire, on procède comme en décimal.

Exemple : 1101 × 101

```
   1101          1101          1101          1101
  ×101          ×101          ×101          ×101
  ----   →      ----    →     ----    →     ----
   1101          1101          1101          1101
                00000         00000         00000
                             110100        110100
                                          -------
                                          1000001
```

Les opérations en binaire :

c. La multiplication :

Exercice N°1 :

Effectuer les multiplications suivantes:

1111 * 10=
1100 *110=

Exercice N°2

Multiplier 10011011 et 11001101 en binaire.

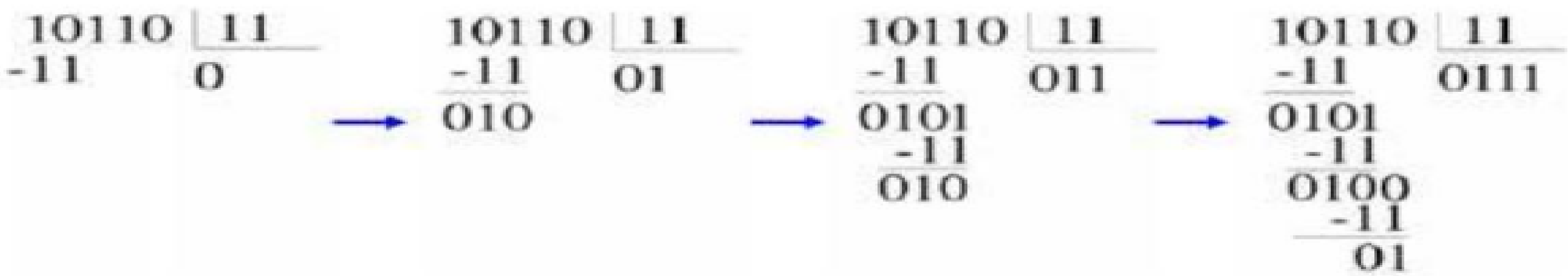
Les opérations en binaire :

d. La division

- La division binaire s'effectue à l'aide de soustractions et de décalages, comme la division décimale, sauf que les digits du quotient ne peuvent être que 1 ou 0.
- Le bit du quotient est 1 si on peut soustraire le diviseur, sinon il est 0.

Pour l'instant, on ne fait que la division entière.

Exemple : 10110 / 11



Les opérations en binaire :

Exercice :

A=201 = 11001001
B=16 = 10000

Calculer en binaire A/B

1.7. Codage BCD (Binary Code Decimal)

- Il faut prendre le codage binaire naturel des 10 chiffres décimaux.

- Chaque chiffre d'un nombre décimal est donc codé sur un mot binaire de 4 bits, car il faut 4 bits pour coder jusqu'à 9 . Or 4 bits permettent de coder les nombres de 0 à 15 : Il ne faut donc pas tenir compte du codage dépassant 9.

0 : 0000 1 : 0001 2 : 0010 3 : 0011 4 : 0100

5 : 0101 6 : 0110 7 : 0111 8 : 1000 9 : 1001 10 à 15 : interdit

- Exemple : Codage d'un entier 234 en BCD : $(0010\ 0011\ 0100)_{BCD}$

Opération d'addition :

Binaire naturel

9	=	1	0	0	1
+ 4	=	0	1	0	0
13	=	1	1	0	1

Binaire naturel → BCD

13	=	(hors norme)	1	1	0	1
(+6)	=	(décalage)	0	1	1	0
13	=		$(0\ 0\ 0\ 1\ 0\ 0\ 1\ 1)_{BCD}$			

1.8. Codage Complément à 2

Définitions :

- Le complément à 2 = codage des nombres signés (+ et -),

- Nombre binaire de n bits → Codage des nombres entiers décimaux de 0 à 2^n-1

- Codage des nombres signés N à partir de n bits : $-2^{n-1} \leq N \leq 2^{n-1}-1$:

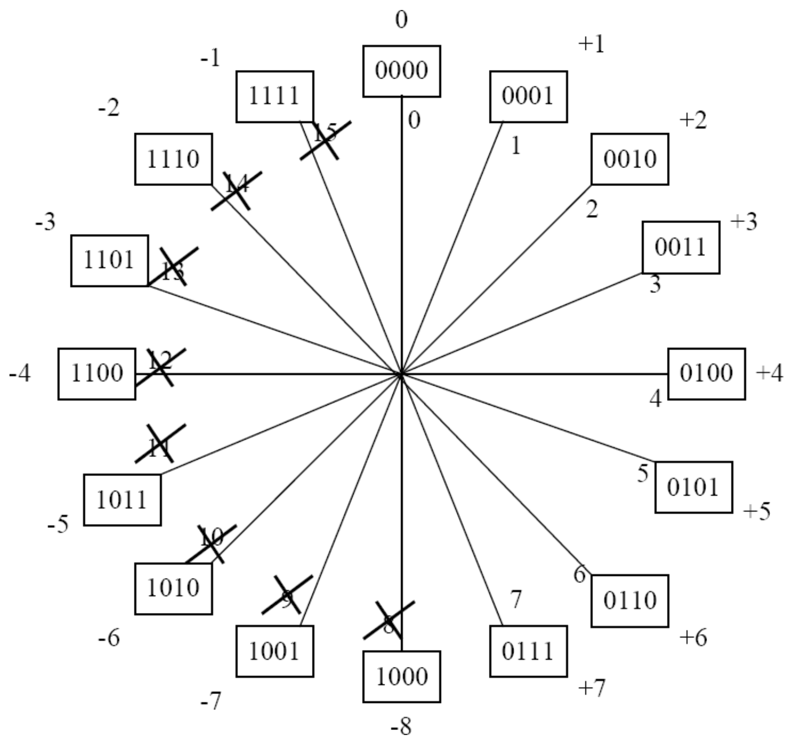
    - Codage du nombre sur (n-1) bits

    -1 bit de signe (celui de gauche : 0 : positif , 1 : négatif)

<u>Exemple : n = 4 :</u>

Il est possible de coder :

- Nombres non signés sur 4 bits : 24=16 : 0 à 16
- Nombres signés sur 4 bits : on « coupe en 2 » 16 / 2 = 8 : -8 à +7



Comment trouver le complément à 2 d'un nombre binaire (l'opposé du nombre décimal) :

➢ Méthode 1 :

on affecte de poids négatif le bit de poids fort:

$(1101)_{C_2} = 1\times(-2^3) + 1\times2^2 + 0\times2^1 + 1\times2^0$

$(1101)_{C_2} = 1\times\mathbf{(-8)} + 1\times\mathbf{4} + 0\times\mathbf{2} + 1\times\mathbf{1} = -3$

Méthode 2 :(**le complément à 2)**

- on prend le complément à 1 du nombre (les 0 deviennent des 1 et les 1 des 0)
- on ajoute 1

Exemple :

	$(0110)_2 = 6$	:	Nombre 6
	$(1001)_2 = \overline{6}$	:	complément à 1 de 6
+1 :	$(0001)_2$	:	On ajoute 1
	$(1010)_2$	:	On obtient (-6)

Comment trouver le complément à 2 d'un nombre binaire (l'opposé du nombre décimal) :

➢ Méthode 3 :

-1. On recopie le nombre binaire de la droite vers la gauche jusqu'au premier 1

- 2. On complémente les autres (les 0 deviennent des 1 et les 1 des 0) :

$(0110)_2 = 6$

$(1010)_2 = -6$

Code de Gray

Le code de Gray, également appelé binaire réfléchi, est un type de codage binaire permettant de ne modifier qu'un seul bit à la fois quand un nombre est augmenté d'une unité.

Le nom du code vient de l'ingénieur américain Frank Gray qui déposa un brevet sur ce code en 1953.

Code de Gray

- **Codage décimal Codage binaire classique Codage Gray ou binaire réfléchi**

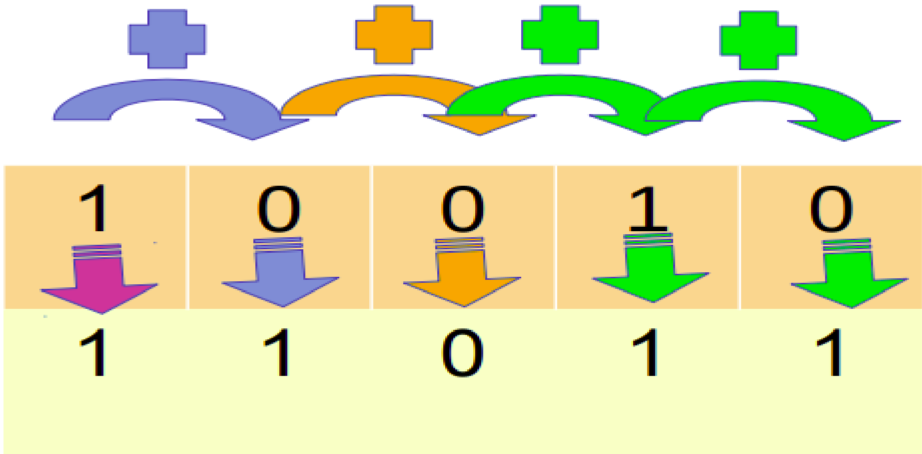
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

Code de Gray

Une méthode pour produire la suite des nombres de gray qui peut être intéressante consiste à partir du binaire, de faire l'addition décalé SANS les retenus et de supprimer le bit de poids faible

Code de Gray

Binaire => gray code



De gauche à droite faire la somme des bits adjacents sans retenue

- 10010 =>(11011)gray

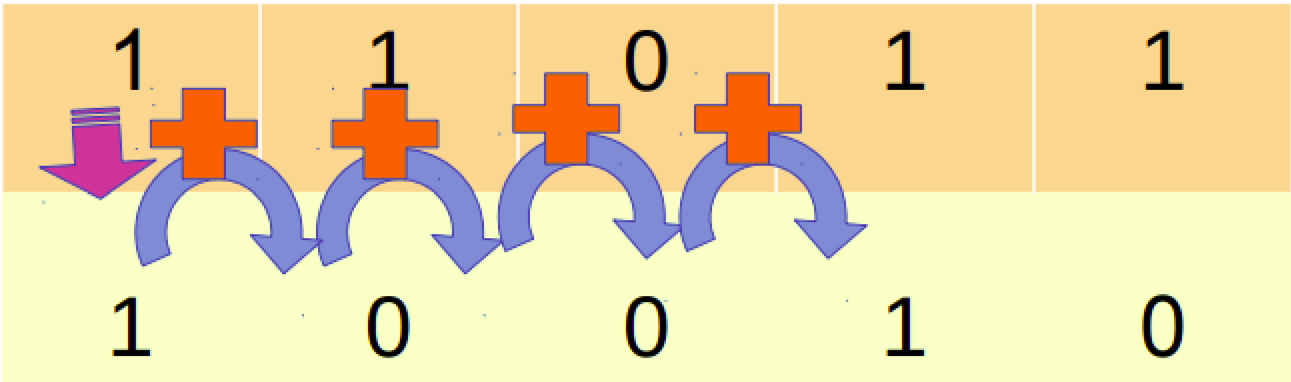
Code de Gray

Exercice

- Convertir le nombre
  - 1010 =>( )gray

Code de Gray

gray code => Binaire



De gauche à droite faire la somme des bits adjacents sans retenue

- ❖ (11011)gray = (10010)

Code de Gray

Exercice

- Convertir le nombre
- $(1010)_{gray} \Rightarrow (\ )_2$

Codage des caractères

Les caractères englobent : les lettres alphabétiques (A, a, B, b,..) , les chiffres , et les autres symboles (> , ; / : ….) .

- ❖ Le codage le plus utilisé est le ASCII (American Standard Code for Information Interchange)
- ❖ 7 bits ->128 caractères :
- ❖ 26 lettres majuscules A - Z
- ❖ 26 lettres minuscule a - z
- ❖ 10 chiffres 0 à 9
- ❖ 33 caractères de ponctuation
  - ✓ sp,! ” #$%& ’ ()*+,-. /< = >?@ [ ] ^_` { | } ~
- ❖ 33 caractères de contrôle :
  - ✓ null, etx, bel, bs, ht, lf, vt, ff, cr, …, del

Codage des caractères

Le code ASCII

Chaque caractère a un « code » unique Entier entre 0 et 255

Exemple :

- ✓ E 69
- ✓ x 120
- ✓ e 101
- ✓ m 109
- ✓ p 112
- ✓ l 108
- ✓ e 101

32		64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o

48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_	127	

Codage des caractères

ASCII étendu

- Dans ce codage chaque caractère est représenté sur 8 bits .
- Avec 8 bits on peut avoir 2 8 = 256 combinaisons
- Chaque combinaison représente un caractère
  - ✓ Exemple :
- Le code 65 $(01000001)_2$ correspond au caractère A
- Le code 97 $(01100001)_2$ correspond au caractère a
- Le code 58 $(00111010)_2$ correspond au caractère :
- Actuellement il existe un autre code sur 16 bits , se code s’appel UNICODE .

Codage des caractères

Par exemple :

- ‘A’ = 6510 = 0100 00012
- ‘B’ = 6610 = 0100 0010
- ...
- ‘a’ = 9710 = 0110 0101
- ‘ ‘ = 3210 = 0010 0000
- ‘0’ = 4810 = 0011 0000
- ‘1’ = 4910 = 0011 0001
- ‘2’ = 5010 = 0011 0010
- …
- ‘9’ = 5710 = 0011 1001

Codage des caractères

- Actuellement il existe un autre code sur 16 bits , se code s'appel UNICODE .

- UNICODE 16 bits -> 65 536 caractères

- Ce code contient en plus de tous les caractères européens, 42 000 caractères asiatiques.

- Le code ASCII est contenu dans les 128 premiers caractères d'UNICODE.

Codage des caractères

Il existe deux formats :

1) 16 bits (UCS-2)   ISO/IEC 10646

2) ou 32 bits (UCS-4)

- ❖ UCS-2 équivalent à UNICODE 2.0
- ❖ UCS-4 inclut :

- ✓ Caractères musicaux
- ✓ Symboles mathématiques
- ✓ Écritures anciennes telles que les hiéroglyphes.

Codage des caractères

Exercice :

Donnez le code ASCII de la chaine de caracteres : “Bonjour ” en decimal ,en binaire et en hexadecimal