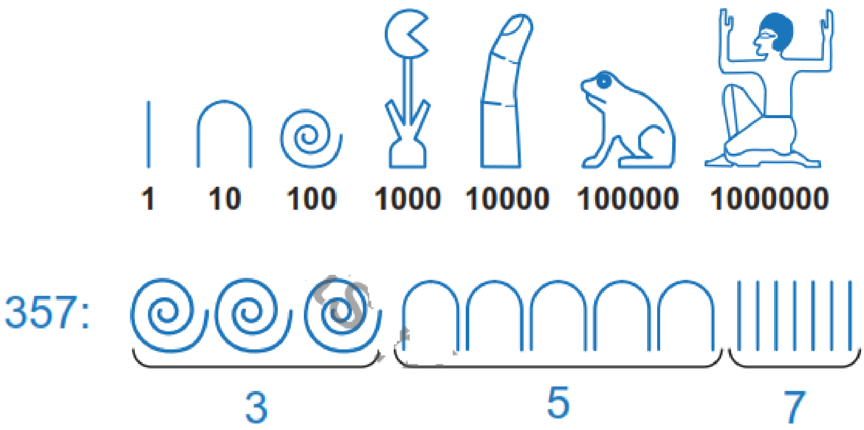
Chapitre1 : SYSTEMES DE NUMERATION ET CODAGE DES INFORMATIONS

1. OBJECTIFS

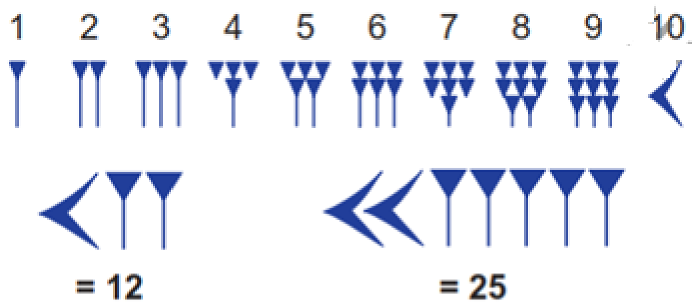
- **Identifier un système de numération ;**
- **Convertir un nombre d'un système de numération en un autre ;**
- **Coder une information dans un format numérique ;**
- **Coder une information dans un format alphanumérique ;**

2. Système de numération

Exemple1 : les Egyptiens utilisaient les sept symboles suivants pour représenter les nombres. Ils ne connaissaient pas le chiffre zéro.



Exemple2 : les Babyloniens utilisaient les symboles ci-dessous pour représenter des nombres



Les Arabes utilisaient les chiffres (0,………,9) pour représenter des nombres, ce sont les mêmes chiffres que nous connaissons actuellement et qui ont beaucoup simplifié la représentation des nombres et les algorithmes des opérations élémentaires.

En ***conclusion*** nous pouvons dire que chaque civilisation avait son propre langage symbolique pour la représentation des nombres.

Vu de tous ces exemples, nous pouvons dire qu'un système de numération est un ensemble de symboles et de règles permettant d'écrire et de nommer les nombres.

Un système de numération est une façon d'énoncer ou d'écrire des nombres. De nombreux systèmes de numérations sont utilisaient en technologie numérique. Les plus courant sont les systèmes ***Décimal*** (base 10), ***Binair***e (base 2), ***Tétral*** (base 4), ***Octal*** (base 8) et ***Hexadécimal*** (base 16).

Le tableau ci-dessous représente un récapitulatif de ces systèmes :

Décimal	Binaire	Tétral	Octal	Hexadécimal
0	0	0	0	0
1	1	1	1	1
2	10	2	2	2
3	11	3	3	3
4	100	10	4	4
5	101	11	5	5
6	110	12	6	6
7	111	13	7	7
8	1000	20	10	8
9	1001	21	11	9
10	1010	22	12	A
11	1011	23	13	B
12	1100	30	14	C
13	1101	31	15	D
14	1110	32	16	E
15	1111	33	17	F

3. Principe de la numération

3.1. Base d'un système de numération

La base d'un système de numération est le nombre de symboles utilisés par le système de numération. Quel que soit la base numérique employée, elle suit la relation suivante :

$$N = \sum_{i=0}^{i=n}(a_i\, b^i) = b_n a^n + \cdots + a_5 b^5 + a_4 b^4 + a_3 b^3 + a_2 b^2 + a_1 b^1 + a_0 b^0$$

Où a_i représente le chiffre de la base de rang i et b^i puissance de la base *b* de rang d'exposant i appelé encore ***poids***. Cette forme de représentation s'appelle ***représentation polynomiale*** ou ***forme polynomiale***

3.2. Le système décimal

C'est le système de numération décimal que nous utilisons tous les jours. C'est le système de **base 10** qui utilise donc 10 symboles différents : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.

Un nombre N (entier positif) exprimé dans le système de numération décimale est défini par la relation ci-dessous :

$N = a_n * 10^n + a_{n-1} * 10^{n-1}$ .... + $a_0 * 10^0$ (où a_n est un chiffre de rang n)

Exemple : $N = (1975)_{10}$
$N = 1 * 10^3 + 9 * 10^2 + 7 * 10^1 + 5 * 10^0$

Les puissances de 10 sont appelés les **poids** ou les **valeurs de position**. Le poids est égal à la base élevée à la puissance de son rang.

	Unité	Dizaine	Centaine	Milliers	10*Milliers	100*Milliers
Chiffre	a0	a1	a2	a3	a4	a5
Rang	0	1	2	3	4	5
Poids	10^0	10^1	10^2	10^3	10^4	10^5

Exercice :

* $N = (6281)_{10}$ =

* $N = (1967)_{10}$ =

* $N = 2 * 10^4 + 8 * 10^3 + 4 * 10^2 + 2 * 10^1 + 9 * 10^0$ =

3.3. Le système binaire

Le système binaire est le système de base 2, c'est à dire qui utilise deux symboles différents : le 0 et le 1. Chacun d'eux est appelé bit (contraction de binary digit) ou élément binaire.
Dans ce système, le poids est une puissance de 2.

Exemple : $N = (10110)_2$

$N = 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$

$N = (22)_{10}$

*** Puissance de 2 :**

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768

Définitions :

Triplet : nombre binaire formé de 3 éléments binaires.

Quartet : nombre binaire formé de 4 éléments binaires.

Octet (byte) : nombre binaire formé de 8 éléments binaires.

Mot (word) : nombre binaire formé de 16, 32 ou 64 éléments binaires.

L.S.B. : bit le moins significatif ou bit de poids faible (élément le plus à droite d'un nombre binaire).

M.S.B. : bit le plus significatif ou bit de poids fort (élément binaire le plus à gauche d'un nombre binaire)

*** *Notations des valeurs binaires :***

Un nombre binaire peut être précédé du signe **%** ou suivi de l'indice de base (2) ou d'un B.

Exemple : % 01000110

$(1000110)_2$

01000110 B.

*** *Cadrage d'un nombre :***

C'est le nombre d'éléments binaires pris pour représenter un intervalle de valeurs. Les éléments binaires les plus significatifs sont situés à droite, les valeurs les moins significatives sont situées à gauche et sont toutes à 0.

Exemple : % 00011011 nombre représenté sur un octet (8 bits)

% 0000000000011011 nombre représenté sur 16 bits.

*** *Valeurs maximum et minimum représentées sur n bits :***

En utilisant n bits, on peut former 2^n nombres différents et le plus grand d'entre eux est égal à (2^n-1).

Exemple : si n = 8 alors : on peut former 256 nombres différents et Nmax = $(2^8 -1) = 255$.

La valeur minimum d'un entier représenté sur n bits est 0 quelque soit le nombre d'éléments binaires.

3.4. Le système octal

Le système de numération octal est de **base 8**, ainsi il utilise 8 symboles différents : 0, 1, 2, 3, 4, 5, 6 et 7.

Dans ce système, le poids est un puissance de 8.

Exemple : $N = (6543)_8$
$N = 6 * 8^3 + 5 * 8^2 + 4 * 8^1 + 3 * 8^0$
$N = (3427)_{10}$

La succession des nombres par ordre croissant est le suivant :

- 1 chiffre : 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, …..

- 2 chiffres : 10, 11, 12, 13, 14, 15, 16, 17, 20, 21......, 27, 30, 31....etc.

*** <u>puissance de 8 :</u>**

n	0	1	2	3	4	5
8^n	1	8	64	512	4096	32768

*** <u>notation d'un nombre octal :</u>**

Un nombre octal peut être précédé du signe **@** ou suivi de l'indice de base (8) ou d'un Q.

Exemple : @ 1672
$(1672)_8$
1672 Q

3.5. LE système hexadécimal

Il est surtout utilisé en informatique. Dans cette base, on utilise 16 chiffres

(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

Exemple d'écriture sous forme polynomiale :

$(3256)_{16}= 3*16^3 + 2*16^2 + 5*16^1 + 6*16^0$
$(9C4F)_{16}= 9*16^3 + 12*16^2 + 4*16^1 + 15*16^0$
$(A2B.E1)_{16}= 10*16^2 + 2*16^1 + 11*16^0 +14*16^{-1}+ 1*16^{-2}$

4. Changement de base

Il s'agit de la conversion d'un nombre écrit dans une base ***B1*** à son équivalent dans une autre base ***B2***.

4.1. Conversion d'un système de base *B* au système décimal

Pour convertir un nombre d'une base ***B*** vers la base décimale, on utilise la méthode dite des **additions** qui consiste à utiliser la représentation du nombre sous ***forme polynomiale***.

Exemple :

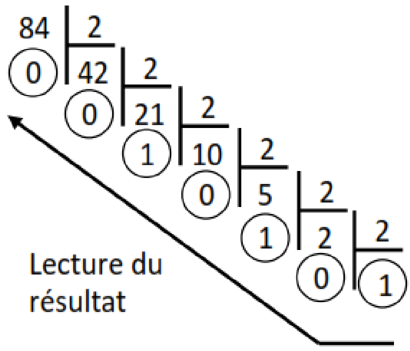
$(1011101)_2= 1*2^6 + 0*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 0*2^1+ 1*2^0=(93)_{10}$
$(231102)_4= 2*4^5 + 3*4^4 + 1*4^3 + 1*4^2 + 0*4^1+ 2*4^0=(2898)_{10}$
$(7452)_8= 7*8^3 + 4*8^2 + 5*8^1+ 2*8^0=(3882)_{10}$
$(D7A)_{16}= 13*16^2 + 7*16^1 + 10*16^0 =(3450)_{10}$

4.2. Conversion d'un nombre décimal entier vers un système de base B

Pour convertir un nombre décimal entier en un nombre de base **B** quelconque, il faut faire des divisions entières successives par la base **B** et conserver à chaque fois le reste de la division. On s'arrête lorsqu'un obtient un résultat inférieur à la base **B.** le nombre recherché **N** dans la base **B** s'écrit de la gauche vers la droite en commençant par le dernier résultat allant jusqu'au premier reste.

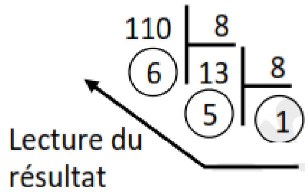
Exemple :

➲ $(84)_{10}=(\ ?\ )_2$



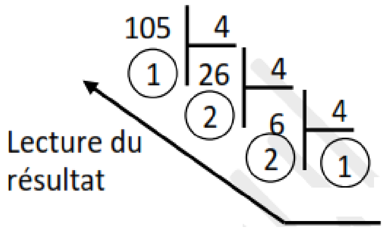
$(84)_{10}=(1010100)_2$

➲ $(110)_{10}=(\ ?\ )_8$



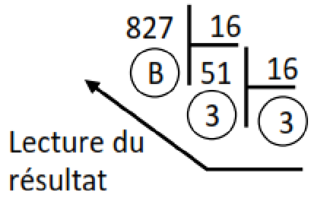
$(110)_{10}=(156)_8$

➲ $(105)_{10}=(\ ?\ )_4$



$(105)_{10}=(1221)_4$

➲ $(827)_{10}=(\ ?\ )_{16}$



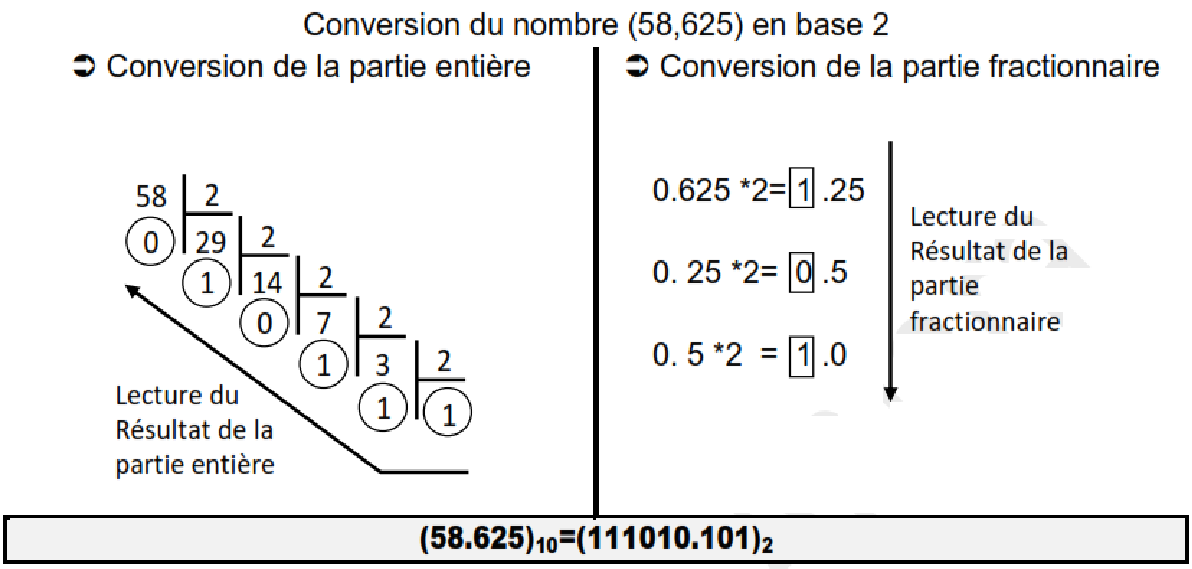
$(827)_{10}=(33B)_8$

4.3. Conversion d'un nombre décimal à virgule vers un système de base B

Pour convertir un nombre décimal à virgule dans une base B quelconque, il faut :

- Convertir la partie entière en effectuant des divisions successives par **B** (comme nous l'avons vu précédemment) ;
- Convertir la partie fractionnaire en effectuant des multiplications successives par **B** et en conservant à chaque fois le chiffre devenant entier.

Exemple :



Remarque : parfois en multipliant la partie fractionnaire par la base **B** on n'arrive pas à convertir toute la partie fractionnaire. Ceci est dû essentiellement au fait que le nombre à convertir n'a pas un équivalent exact dans la base **B** et sa partie fractionnaire est cyclique.

Exemple : $(0.15)_{10} = (\ )$

0.15 *2 = **0** .3
0.3 *2 = **0** .6
0.6 *2 = **1** .2
0.2 *2 = **0** .4
0.4*2 = **0** .8
0.8*2 = **1** .6
0.6 *2 = **1** .2
0.2 *2 = **0** .4
0.4*2 = **0** .8
0.8*2 = **1** .6

➲ $(0.15)_{10}=(0.0010011\underline{\mathbf{1001}})_2$

On dit que le nombre $(0.15)_{10}$ est cyclique dans la base 2 de période **1001.**

4.4. Conversion de la base 2 vers la base 2^n

L'utilisation des bases 2^n (4, 8 et 16) permet de réduire le nombre de symboles à écrire tout en conservant la possibilité de conversion instantanée en binaire.

Pour convertir un nombre de la base 2 vers la base 2^n, il suffit de regrouper les bits par groupes de *n* (2 pour la base tétrale, 3 pour la base octale et 4 pour la base hexadécimale), et de remplacer chacun de ces groupes par le symbole correspondant dans la base d'arrivée.

Exemple : conversion de N_2 = 1100111010101 en base 8 puis en 16.

$$\textit{Base 8}: \ N = 1\ 100\ 111\ 010\ 101_{(2)} = \underbrace{001}_{1_{(8)}}\ \underbrace{100}_{4_{(8)}}\ \underbrace{111}_{7_{(8)}}\ \underbrace{010}_{2_{(8)}}\ \underbrace{101}_{5_{(8)}} = 14725_{(8)},$$

$$\textit{Base 16}: \ N = 1\ 1001\ 1101\ 0101_{(2)} = \underbrace{0001}_{1_{(16)}}\ \underbrace{1001}_{9_{(16)}}\ \underbrace{1101}_{D_{(16)}}\ \underbrace{0101}_{5_{(16)}} = 19D5_{(16)}$$

4.5. Conversion de la base 2^n vers la base 2

Pour la conversion inverse, il suffit de développer chaque symbole de la représentation dans la base 2^n sur n bits.

Exemple : convertir $4A1_{16}$ dans la base

$$4A1_{(16)} = \underbrace{0100}_{\text{4 en base 2}}\ \underbrace{1010}_{\text{A en base 2}}\ \underbrace{0001}_{\text{1 en base 2}} = 010010100001_{(2)}.$$

4.6. Conversion de la base *i* vers la base *j*

- Si *i* et *j* sont des puissances de 2 ($i = 2^n$ et $j = 2^m$) alors on utilise la base 2 comme relais ;

  Exemple : base 8 → base 2 → base 16
- Sinon, on utilise la base 10 comme relais.

  Exemple : base 5 → base 10 → base 2

5. Codage de l'information

Toute information en clair est constituée de trois types de caractères :

- Des textes ensemble de mots eux-mêmes constitués de lettres ;
- Des nombres représentés par des chiffres ;
- des symboles qui peuvent être de nature très diverse (signes de ponctuation, opérateurs arithmétiques, opérateurs logiques, ….. etc).

On distingue deux catégories de codes

- Les codes numériques qui permettent seulement le codage des nombres ;
- Les codes alphanumériques qui permettent le codage d'une information quelconque (ensemble de lettres, de chiffres et de symboles)

5.1. Les codes numériques

5.1.1. Code binaire naturel (pur)

Ce code est utilisé uniquement pour représenter les chiffres décimaux . Ce code est encore appelé ***code 8421***

N	8	4	2	1
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
8	1	0	0	0
9	1	0	0	1

Ce code présente l'inconvénient de changer plus qu'un seul bit quand on passe d'un nombre à un autre immédiatement supérieur.

5.1.2. Code BCD (Binaire Codé en Décimal)

Sa propriété est d'associer 4 bits représentant chaque chiffre en binaire naturel. L'application la plus courante est celle de l'affichage numérique ou chaque chiffre est associé à un groupe de 4 bits portant le code BCD.

Exemple

$$(9\ 4\ 2\ 7)_{10} = (\underline{1001}\ \underline{0100}\ \underline{0010}\ \underline{0111})_{DCB}$$

$$(6\ 8\ 0\ 1)_{10} = (\underline{0110}\ \underline{1000}\ \underline{0000}\ \underline{0001})_{DCB}$$

5.2.3. Les codes non pondérés : codes Gray

Il n'y a pas de poids affecté aux éléments binaires. Son intérêt réside dans des applications d'incrémentation ou un seul bit change d'état à chaque incrémentation.

Conversion binaire / Gray

$$N = (a_{n-1} \ldots a_i \ldots a_0)_2 \quad est \quad équivalent \quad à \quad (g_{n-1} \ldots g_i \ldots g_0)_{CBR}$$

$$g_{n-1} = a_{n-1}$$

$$pour \quad les \quad rangs \quad n-2 \geq i \geq 0 \quad g_i = a_{i+1} \oplus a_i$$

Exemple:

$$(1010)_2 = (?)_{CBR}$$

$$g_3 = a_3 = 1$$
$$g_2 = a_3 \oplus a_2 = 1 \oplus 0 = 1$$
$$g_1 = a_2 \oplus a_1 = 0 \oplus 1 = 1$$
$$\underbrace{g_0 = a_1 \oplus a_0 = 1 \oplus 0 = 1}$$
$$(1010)_2 = (1111)_{CBR}$$

$$(1111)_2 = (?)_{CBR}$$

$$g_3 = a_3 = 1$$
$$g_2 = a_3 \oplus a_2 = 1 \oplus 1 = 0$$
$$g_1 = a_2 \oplus a_1 = 1 \oplus 1 = 0$$
$$\underbrace{g_0 = a_1 \oplus a_0 = 1 \oplus 1 = 0}$$
$$(1111)_2 = (1000)_{CBR}$$

$$(24)_{10} = (?)_{CBR}$$

$$g_4 = a_4 = 1$$
$$g_3 = a_4 \oplus a_3 = 1 \oplus 1 = 0$$
$$g_2 = a_3 \oplus a_2 = 1 \oplus 0 = 1$$
$$g_1 = a_2 \oplus a_1 = 0 \oplus 0 = 0$$
$$\underbrace{g_0 = a_1 \oplus a_0 = 0 \oplus 0 = 0}$$
$$(24)_{10} = (11000)_2 = (10100)_{CBR}$$

Une autre méthode de passage binaire GRAY est :

- Le bit de gauche du mot binaire reste inchangé
- Additionner le bit de poids le plus significatif au bit suivant et reporter le résultat sans tenir compte de la retenue ;
- Continuer l'addition de chaque bit avec le bit suivant jusqu'au bit du poids le plus faible

Exemple :

Binaire : 1 0 1 1 0 1 0

Gray : 1 1 1 0 1 1 1

Le tableau suivant donne l'équivalent gray d'un nombre binaire tel que N≤15.

Décimal	Binaire naturel	Binaire réfléchi
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

Conversion Gray/ Binaire

Pour un mot binaire de format n on prend

$$a_{n-1} = g_{n-1}$$

$$a_i = a_{i+1} \oplus g_i \qquad pour \quad les \quad rangs \quad n-2 \geq i \geq 0$$

Exemple

$(1110)_{Gray} = (?)_2$

$$\underbrace{\begin{aligned} a_3 &= g_3 = 1 \\ a_2 &= a_3 \oplus g_2 = 1 \oplus 1 = 0 \\ a_1 &= a_2 \oplus g_1 = 0 \oplus 1 = 1 \\ a_0 &= a_1 \oplus g_0 = 1 \oplus 0 = 1 \end{aligned}}$$

$(1110)_{Gray} = (1011)_2$

$(01000)_{Gray} = (?)_2$

$$\underbrace{\begin{aligned} a_3 &= g_3 = 0 \\ a_2 &= a_3 \oplus g_2 = 0 \oplus 1 = 1 \\ a_1 &= a_2 \oplus g_1 = 1 \oplus 1 = 0 \\ a_0 &= a_1 \oplus g_0 = 0 \oplus 0 = 0 \end{aligned}}$$

$(0110)_{Gray} = (0100)_2$

$(10001)_{Gray} = (?)_{10}$

$$\underbrace{\begin{aligned} a_4 &= g_4 = 1 \\ a_3 &= a_4 \oplus g_3 = 1 \oplus 0 = 1 \\ a_2 &= a_3 \oplus g_2 = 1 \oplus 0 = 1 \\ a_1 &= a_2 \oplus g_1 = 1 \oplus 0 = 1 \\ a_0 &= a_1 \oplus g_0 = 1 \oplus 1 = 0 \end{aligned}}$$

$(10001)_{Gray} = (11110)_2 = (30)_{10}$

5.1. Les codes alphanumériques : le code ASCII

Le code **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) est un code alphanumérique, devenu une norme internationale. Il est utilisé pour la transmission entre ordinateurs ou entre un ordinateur et des périphériques. Sous sa forme standard, il utilise 7 bits. Ce qui permet de générer $2^7 = 128$ caractères. Ce code représente les lettres alphanumériques majuscules et minuscules, les chiffres décimaux, des signes de ponctuation et des caractères de commande.
Chaque code est défini par 3 bits d'ordre supérieur $\mathbf{b_6b_5b_4}$ et 4 bits d'ordre inférieur $\mathbf{b_3b_2b_1b_0}$. Ainsi le caractère 'A' a pour code hexadécimal 41_H.

Exemple :

A ⇨ $(65)_{ASCII}$ ⇨ $(01000001)_2$ ⇨ $(41)_H$

B ⇨ $(66)_{ASCII}$ ⇨ $(01000010)_2$ ⇨ $(42)_H$

Z ⇨ $(90)_{ASCII}$ ⇨ $(01011010)_2$ ⇨ $(5A)_H$

a ⇨ $(97)_{ASCII}$ ⇨ $(01100001)_2$ ⇨ $(61)_H$

b ⇨ $(98)_{ASCII}$ ⇨ $(01100010)_2$ ⇨ $(62)_H$

z ⇨ $(122)_{ASCII}$ ⇨ $(01111010)_2$ ⇨ $(7A)_H$

[⇨ $(91)_{ASCII}$ ⇨ $(01011011)_2$ ⇨ $(5B)_H$

{ ⇨ $(123)_{ASCII}$ ⇨ $(01111011)_2$ ⇨ $(7B)_H$

ASCII
(American Standard Code for Information Interchange) représente le code de norme américaine pour l'échange de l'information

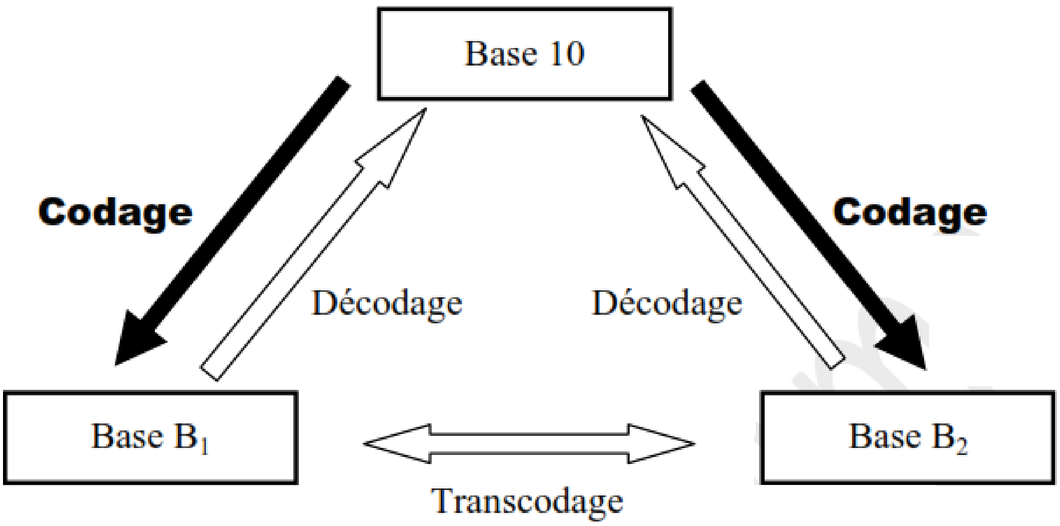
				B6	0	0	0	0	1	1	1	1
				B5	0	0	1	1	0	0	1	1
				B4	0	1	0	1	0	1	0	1
B3	B2	B1	B0	Exemple : E = 100 0101$_{(2)}$ = 69$_{(10)}$. Appuyez sur "ALT", saisissez 69 et relâchez "ALT". Convaincu !								
0	0	0	0		NUL	DLE	SP	0	@	P	`	p
0	0	0	1		SOH	DC1	!	1	A	Q	a	q
0	0	1	0		STX	DC2	~	2	B	R	b	r
0	0	1	1		ETX	DC3	#	3	C	S	c	s
0	1	0	0		EOT	DC4	$	4	D	T	d	t
0	1	0	1		ENQ	NAK	%	5	E	U	e	u
0	1	1	0		ACK	SYN	&	6	F	V	f	v
0	1	1	1		BEL	ETB	'	7	G	W	g	w
1	0	0	0		BS	CAN	(	8	H	X	h	x
1	0	0	1		HT	EM	)	9	I	Y	i	y
1	0	1	0		LF	SUB	*	:	J	Z	j	z
1	0	1	1		VT	ESC	+	;	K	[	k	{
1	1	0	0		FF	FS	,	<	L	\	l	|
1	1	0	1		CR	GS	-	=	M	]	m	}
1	1	1	0		SO	RS	.	>	N	^	n	~
1	1	1	1		SI	US	/	?	O	_	o	DEL

$N=(1001110)_2= 78$

Faire alt 78 pour avoi N

1.1. Le transcodage

Une des applications liée au codage des informations est le passage d'un code à un autre. Cette opération est appelée transcodage.



Le codage des informations se fait au moyen d'un circuit combinatoire appelé **codeur.**
Le décodage des informations se fait au moyen circuit combinatoire appelé **décodeur**.
Un transcodeur est un **décodeur** associé à un **code**