chapitre. 2

Système de numération et codage de l'information

II.1. Système de numération

Introduction :

Le processeur est le système automatique de traitement de l'information, ces informations peuvent prendre plusieurs formes :
Texte écrit, parole prononcée, image observée, grandeur physique mesurée, image observée, nombre…etc.
Cependant, ces différentes formes de l'information ne sont pas compréhensibles par le processeur (machine).
La machine est conçue de tell sorte qu'elle comprenne une seule forme de l'information : (0) et (1). Le système conventionnel de comptage est bien le système décimal (base 10) incompatible avec la machine. Cela nous conduit à utiliser d'autre système de numération plus souple à l'utilisation.

II.1.1. Différents systèmes de numération

Les systèmes de numération les plus utilisés sont:

- Le système binaire
- Le système hexadécimal
- Le système décimal
- Le système octal

Notion de base :

Définition:

La base est le nombre qui sert à définir un système de numération.

Exemple : le système décimal $\longrightarrow$ base 10

le système octal $\longrightarrow$ base 8

Représentation d'un nombre dans une base quelconque β :

Un nombre peut être écrit sous la forme suivante :

$$\sum_{1=n}^{i=0} b_i\,\beta^i = b_n\beta^n + b_{n-1}\beta^{n-1} + \cdots b_2\beta^2 + b_1\beta^1 + b_0\beta^0$$

Où b_i : est le chiffre de la base de rang (i).

β^i est la puissance de la base β avec un exposant de rang (i).

Exemple : base 10.

Le nombre 2015 est de rang n =3, Il s'écrit alors : $2.10^3 + 0.10^2 + 1.10^1 + 5.10^0$

$= 2015$

II.1.1.1. Système décimal

Le système décimal est le système le plus couramment utilisé dans nos habitudes. Chaque chiffre peut avoir 10 valeurs différentes :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, pour cette raison que la base de ce système est appelée base 10.

Dans l'exemple précédent, le nombre 2015 s'écrit :

$2015 = 2.10^3 + 0.10^2 + 1.10^1 + 5.10^0$

Les chiffres : 2, 0, 1, 5 sont multipliés par une puissance de 10 appelée poids du chiffre.

- 3 : le plus fort poids
- 0 : le plus faible poids.

Remarque : Si on change la position du poids, la représentation de ce nombre sera erronée, d'où le système de numération de position.

II.1.1.2. Système octal

Le système octal utilise un système de numération ayant une base égale à 8.

Octal : en latin octo qui veut dire, huit.

Dans ce système les symboles se limitent à 8 seulement : 0, 1, 2, 3, 4, 5, 6, 7.

Exemple :

$(2015)_8 = 2*8^3 + 0*8^2 + 1*8^1 + 5*8^0$

$= 2*512 + 0*64 + 1*8 + 5*1$

$= 1037.$

II.1.1.3. Système binaire

Le nombre est représenté par deux valeurs : 0 et 1, d'où la dénomination : binaire, sa base est égale à 2.

Exemple : le nombre 2015 est trop long à représenter dans la base 2, prenons le nombre 215

$(215)_2 =$ 128 + 64 + 16 + 4 + 2 + 1

$= 2^6 + 2^5 + 2^4 + 2^2 + 2^1 + 2^0$

$= 1*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 1*2^0$

$= (11100111)_2$

II.1.1.4. Système hexadécimal

Le système hexadécimal utilise les 16 symboles suivants :

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

Le système a pour la base la valeur 16.

Exemple : $(5AF)_{16}$, ce nombre est écrit dans la base 16, il peut être décomposer comme suit :

$(5AF)_{16} = 5*16^2 + A*16^1 + F*16^0$, le rang étant égale à 2 (n =2, i = 0, 1, 2).

Calculons la valeur de ce nombre en décimal : $16^2 = 256$, $16^1 = 16$, et $16^0 = 1$

Ainsi : A = 10 et F = 15 en décimal, le nombre $(5AF)_{16}$ est égal donc à 1455 en décimal.

II.1.2. Correspondance entre base (2), base (10) et base (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

Tab.1.

Le nombre $(5AF)_{10}$ peut se décomposer comme suit :

$(5AF) = 5*16^2 + A*16^1 + F*16^0$

En remplaçant A et F par leur équivalent en base 10, on obtient :

$(5AF)_{10} = 5*16^2 + A*16^1 + F*16^0$

$= (1455)_{10}$

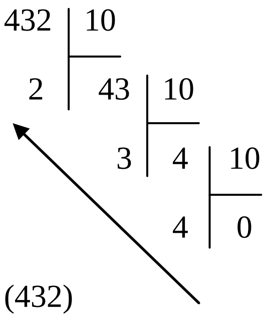
II.1.3. Conversion et changement de base

La conversion de nombre est une opération qui consiste à passer d’un système de numération vers un autre.

a) Conversion décimal vers binaire

Prenons un nombre quelconque exprimé dans la base 10.

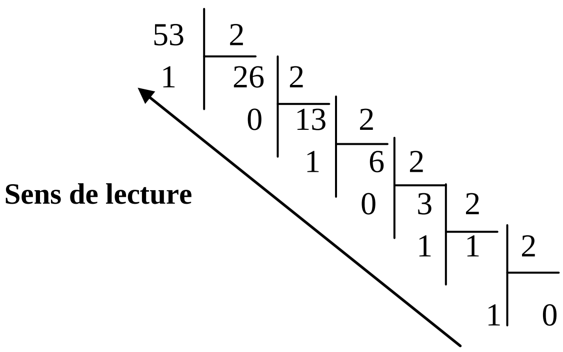
Soit le nombre : 432



(432)

A partir de ce principe, pour obtenir l’expression binaire d’un nombre exprimé en décimal, il suffit de réaliser des divisions successives de ce nombre par 2 jusqu’à ce que le quotient obtenu soit égale à 0. Les restes de ces divisions sont prises de bas en haut représentant le nombre binaire.

Exemple : $(53)_{10} = (?)_2$



Le nombre $(53)_{10} = (110101)_2$

b) Conversion décimal / octal et décimal/binaire

$(149)_{10}$ = ? base 8 et base 2

149 | 8
5 | 18 | 8
2 | 2 | 8
2 | 0

$(225)_8$

149 | 2
1 | 74 | 2
0 | 37 | 2
1 | 18 | 2
0 | 9 | 2
1 | 4 | 2
0 | 2 | 2
0 | 1 | 2
1 | 0

Sens de lecture

= $(10010101)_2$

c) Conversion simplifiée : de la base (8) à la base (2).

$(225)_8$ = ? $(\ \ )_2$,

	2	2	5
En binaire :	010	010	101

= (010 010 101) = $(010010101)_2$ c'est bien vérifié.

Exemple de conversion binaire/octal et octal/binaire.

Binaire :	101	110	100	011
Octal :	(5	6	4	$3)_8$

Octal : $(\underline{3} \quad \underline{7} \quad \underline{4})_8$

Binaire : $(011 \quad 111 \quad 100)_2$

d) Relation entre les nombres binaires et les nombres hexadécimaux.

Exemple : soit le nombre binaire $(110001001011)_2$ = ? ($)_{16}$

Binaire ⟶ hexadécimal

$\underline{1100} \quad \underline{0100} \quad \underline{1011} = (A4B)_{16}$

A 4 B

Hexadécimal ⟶ Binaire

Soit le nombre : (2AF4), sa représentation en binaire est donnée par :

$\underline{2} \quad \underline{A} \quad \underline{F} \quad \underline{4}$

$(0010 \quad 1010 \quad 1111 \quad 0100)_2$

II.1.4. Représentation des nombres signés

Les nombres positifs et négatifs dans le système décimal sont représentés par le signe (+) et (-), respectivement pour les nombres positifs et négatifs. Dans une machine informatique où le traitement des nombres s'effectue en binaire, la représentation est différente. Dans le système binaire par convention, la valeur (0) est attribuée au signe (+) et le (1) au signe (-).

Exemple :

0	100001

Signe nombre

a) Nombre binaire signé

Le bit le plus significatif est utilisé pour représenter le signe du nombre.

- Nombre négatif : si le bit de plus fort poids est égale à(1).
- Nombre positif : si le bit de plus fort poids est égale à(0).

Les autres bits codent la valeur absolue du nombre.

Ex : (-24) et (-128), en binaire signé (bs).

Sur 8 bits, (-24) : 10011000

En effet : 24 s'écrit en binaire sur 8 bits : 00011000, et comme le nombre est négatif on affecte un (1) au bit de plus fort poids. Le nombre -24 est alors représenté par : 10011000

Sur 8 bits, (-128) : le nombre 128 est représenté sur 9 bits.

En effet : $128 = 2^7 = 10000000$, le bit qui indique le signe (-) est donc le 9ème bit de plus fort poids, d'où (-128) = 110000000.

II.2. Codage des nombres

Définition : un code est une correspondance arbitraire entre un ensemble de symboles et un ensemble d'objets. Certains codes permettent d'effectuer des opérations arithmétiques, d'autres permettent de détecter des erreurs lors d'une transmission de données voir les corriger.

II.2.1. les codes pondérés

Définition : un code est dit pondéré, si la position de chaque symbole dans chaque mot correspond à un poids fixe.

a) Code binaire :

   Les codes étudiés précédemment sont des codes pondérés. Dans une base donnée, chaque bit est affecté d'un poids proportionnel à sa position. Il existe d'autres types de codes pondérés notamment les représentations décimales codées binaires.

b) Code BCD (Binary Coded Decimal)

   Il s'agit d'une représentation des nombres en base 10, où chaque chiffre est codé en binaire. Il faut quatre bits pour représenter les 10 chiffres de la base 10. Chaque bit d'un groupe de 4 est affecté de son poids naturel.

   On écrit ainsi par exemple:

   $(421)_{10} = (0100\ 0010\ 0001)_{BCD}$ au lieu de : $(110100101)_2$ en binaire naturel.

   En code BCD un nombre de (n) chiffres occupe toujours 4n bits.

II.2.2. les codes non pondérés

a) Code ASCII (American Standard Code for Information Interchange)

   C'est le code le plus utilisé dans la transmission entre une unité centrale et ses périphériques. Il sert à coder des lettres, des chiffres et un certains nombres d'ordres qui correspondent souvent aux touches du clavier (par exemple la touche ENTREE).

Ces symboles sont codés en binaire sur 7 bits, ce qui permet $2^7 = 128$ possibilités.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Tab.2. Code ASCII et sa correspondance en décimal et en Hexadécimal.

b) Codes adjacents

Deux chiffres ou nombres consécutifs ont toujours des représentations qui ne diffèrent que par un seul bit, on dit qu'il s'agit d'un code adjacent.

Si l'adjacence est complète (avec retour au point de départ) on parle de code cyclique. ou code de Gray. Tab.2.

c) Code correcteur d'erreurs

Dans la transmission numérique, il est toujours possible qu'un bit soit modifié. Des codes ont été donc développés pour détecter et éventuellement corriger des erreurs.

- Ajout d'un bit de parité :

Un bit supplémentaire est ajouté à l'information que l'on souhaite transmettre, ce bit représente la parité du nombre total des (1).

Exemple :

Information	Bite de parité
1010	0
1000	1
1110	1

Tab.3.

On détecte ainsi l'erreur qui porte sur un seul bit.

- Code (p) parmi (n) :

Dans ce type de code, les chiffres de 0 à 9 sont représentés par des combinaisons toujours (p) bits à (1) parmi les (n) bits codant le chiffre.

Exemple : code (2) parmi (5).

Code (2) parmi (5) A B C D E	décimal	Binaire standard a b c d e
11000	0	0000
00011	1	0001
00101	2	0010
00110	3	0011
01001	4	0100
01010	5	0101
01100	6	0110
10001	7	0111
10010	8	1000
10100	9	1001

Tab.4

	Gray décimal	Gray décimal	Binaire pur
0	0000	0001	0000
1	0001	0101	0001
2	0011	0111	0010
3	0010	1111	0011
4	0110	1011	0100
5	0111	1010	0101
6	0101	1110	0110
7	0100	0110	0111
8	1100	0100	1000
9	1101	0000	1001
10	1111		1010
11	1010		1011
12	1011		1100
13	1011		1101
14	1001		1110
15	1000		1111

Tab.5. Code Gray pour la numérotation hexadécimale et décimale.