

# Numération et codage

## Table des matières

1. Définition.....	2
1.1. Unité de codage.....	2
1.2. Unité de transfert.....	2
1.3. Mots binaires.....	2
2. Codage.....	3
2.1. Base 10 ou décimale.....	3
2.2. Base 2 ou binaire.....	3
2.3. Décimale → binaire.....	3
2.3.1. la division successive par 2.....	3
2.3.2. le tableau.....	4
2.4. Base 16 ou Hexadécimale.....	4
2.5. Décimale → Hexadécimale.....	5
2.5.1. la division successive par 16.....	5
2.5.2. le tableau.....	5
3. Un peu d'histoire sur les encodages.....	5
3.1. Le code US-ASCII.....	5
3.2. Les encodages ISO-latin.....	7

Aujourd’hui nos ordinateurs, téléphones et autres appareils savent manipuler aussi bien des nombres et du texte que des images, de la vidéo ou de la musique... Mais comment représenter, au sein d’un système numérique, cette diversité des objets du monde réel ou virtuel ?

Quelles sont les techniques utilisées pour représenter numériquement les grandeurs qui nous entourent ?



# 1. Définition

## 1.1. Unité de codage

Les composants constituant un système informatique réagissent, de manière interne, à des signaux « **tout ou rien** ». On représente les deux états stables ainsi définis par les symboles « **0** » et « **1** » ou encore par « **L** » (Low) et « **H** » (High).

Le système de numération adaptée à la représentation de tels signaux est **la base 2**, on parle alors de **codage binaire**.

L'unité de codage de l'information est un élément ne pouvant prendre que les valeurs 0 ou 1 ; le **bit**.

## 1.2. Unité de transfert

Pour les échanges de données, les informations élémentaires (bits) sont manipulées par groupes qui forment ainsi des mots binaires. La taille de ces mots est le plus souvent un multiple de  $8 = 2^3$ .

L'unité de transfert utilisée pour les échanges de données est le mot de 8 bits appelé **octet** (byte en anglais).

**Exemples :** (2 octets)

1111 0011

1010 1111

## 1.3. Mots binaires

Dans un mot binaire, le bit situé le plus à gauche est le bit le plus significatif, **MSB (Most Significant Bit)**, celui situé le plus à droite est le bit le moins significatif, **LSB (Less Significant Bit)**.

**Exemple :**

<b>MSB</b>								<b>LSB</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>
<b>bit de poids fort</b>				<b>bit de poids faible</b>				
<b>octet</b>								

La capacité en octets des différents constituants tels que circuits mémoires, disques durs, ... est souvent importante : il devient indispensable d'utiliser **des unités multiples de l'octet**.

ko	(kB)	=	kilo-octet	(kiloByte)	=	$10^3$	octets	=	1000	octets
Mo	(MB)	=	Méga-octet	(MegaByte)	=	$10^6$	octets	=	1000	ko
Go	(GB)	=	Giga-octet	(GigaByte)	=	$10^9$	octets	=	1000	Mo
To	(TB)	=	Téra-octet	(TeraByte)	=	$10^{12}$	octets	=	1000	Go

(k, M, G, T, ... = multiple du système international , b=bit, B=Byte, bi=binary)

Exemple : Votre FAI<sup>1</sup> vous annonce un débit descendant de 8 600 kbits/s. *Quelle sera le temps théorique minimal de téléchargement d'une application de taille égale à 25 Mo ?*

<sup>1</sup> Fournisseur d'Accès Internet

## 2. Codage

### 2.1. Base 10 ou décimale

Les nombres que nous utilisons habituellement sont ceux de la base 10 (système décimal).

Nous disposons de dix chiffres différents de 0 à 9 pour écrire tous les nombres.

D'une manière générale, toute base N est composée de N chiffre de 0 à N-1.

Soit un nombre décimal  $N = 2348$ . Ce nombre est la somme de 8 unités, 4 dizaines, 3 centaines et 2 milliers.

Nous pouvons écrire  $N = (2 \times 1000) + (3 \times 100) + (4 \times 10) + (8 \times 1)$

$$2348 = (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (8 \times 10^0)$$

10 représente la base et les puissances de 0 à 3 le rang de chaque chiffre.

Quelque soit la base, le chiffre de droite est celui des unités.

Celui de gauche est celui qui a le poids le plus élevé.

### 2.2. Base 2 ou binaire

Elle possède 2 symboles (0 et 1) appelés également **digits** ou plus communément **bits**<sup>2</sup>.

Comme en base 10, à chaque bit du mot en base 2 correspond une puissance de 2

Puissances	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Valeurs en décimal	128	64	32	16	8	4	2	1
exemple	0	1	1	0	1	1	1	0
Valeur en décimale correspondante	$64 + 32 + 8 + 4 + 2 = 110$							

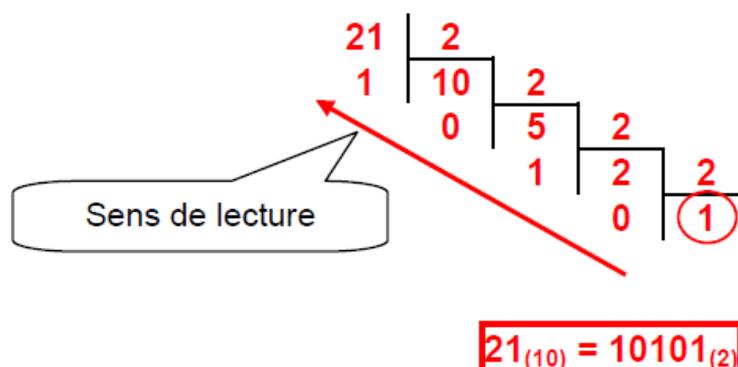
### 2.3. Décimale → binaire

Des indices ou un préfixe peuvent être utilisés pour les nombres binaires :  $110011_{(2)}$ ,  $1101_{(BIN)}$ ,  $\%111000$ .

Pour coder un nombre décimal en binaire, on peut utiliser plusieurs méthodes.

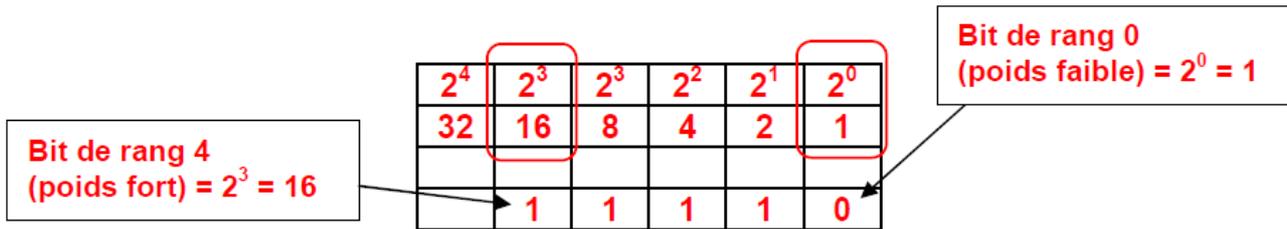
**Exemple** : codage des nombres  $21_{(10)}$  et  $30_{(10)}$  en binaire.

#### 2.3.1. la division successive par 2



<sup>2</sup> abréviation de **binary digit**

### 2.3.2. le tableau



$$30 = 16 \times 1 + 8 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 0$$

$$30_{(10)} = 11110_{(2)}$$

### 2.4. Base 16 ou Hexadécimale

Le binaire, s'il est très représentatif du codage interne des machines, reste très délicat et fastidieux à manipuler. Les programmeurs ont très vite ressenti la nécessité d'utiliser une représentation plus rapide des nombres binaires.

Binaire %	Hexadécimal H ou Ox	Décimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Dans le système hexadécimale les dix premiers symboles correspondent à ceux utilisés dans le système décimal : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9, et les six derniers correspondent aux premières lettres de l'alphabet latin : A, B, C, D, E et F, lesquelles valent respectivement 10, 11, 12, 13, 14 et 15 en base 10.

Puissances	$16^4$	$16^3$	$16^2$	$16^1$	$16^0$
Valeurs en décimal	65536	4096	256	16	1
exemple	0	0	0	6	B
Valeur en décimale correspondante	$6 \times 16 + 11 \times 1 = 107$				

## 2.5. Décimale → Hexadécimale

Des indices ou un préfixe peuvent être utilisés pour les nombres hexadécimaux :  $15_{(16)}$ ,  $23_{(HEX)}$ ,  $0x55F$ ,  $\$AF4$ ,  $\&h38$ ,  $\#44B$ .

**Remarque :**

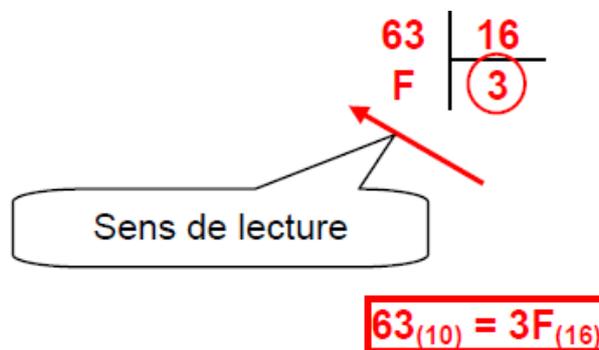
- Le préfixe  $0x$  est utilisé dans le langage C, C++ et JAVA
- $\$$  est utilisé dans le langage Pascal
- $\&h$  dans le langage Basic
- le  $\#$  dans le HTML.

Une autre écriture courante est l'ajout du suffixe « h » à la fin du nombre ( $F15Ah$  par exemple).

Pour coder un nombre décimal en hexadécimale, on peut utiliser plusieurs méthodes.

**Exemple :** codage des nombres  $63_{(10)}$  et  $80_{(10)}$  en hexadécimale.

### 2.5.1. la division successive par 16



### 2.5.2. le tableau

$16^2$	$16^1$	$16^0$
256	16	1
	5	0

$$80 = 5 \times 16 + 0 \times 1$$

$$80_{(10)} = 50_{(16)}$$

## 3. Un peu d'histoire sur les encodages

### 3.1. Le code US-ASCII

Une grosse part des informations manipulées par les systèmes numériques concerne le langage parlé ou écrit matérialisé sous formes de textes, eux-mêmes constitués de caractères typographiques. Comment coder universellement ces caractères et permettre ainsi l'échange d'informations entre machines et/ou utilisateurs, quelle que soit la langue utilisée ?

Le jeu de caractères codés ASCII<sup>3</sup> (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) ou code américain normalisé pour l'échange d'informations, est la norme de codage de caractères en

<sup>3</sup> inventé par Bob BERNER en 1961

informatique la plus connue, la plus ancienne et la plus largement compatible.

Le code ASCII est un code sur 7 bits (valeurs 0 à 127), il permet de définir :

- des caractères imprimables universels : lettres minuscules et majuscules, chiffres, symboles,...
- des codes de contrôle non imprimables : indicateur de saut de ligne, de fin de texte, codes de contrôle de périphériques, ...

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Table ASCII sur 7 bits

Exemple : *Décrypter la chaîne ASCII ci-dessous représentée sous la forme d'une suite d'octets :*

0101 0011 0101 0100 0101 1001

### 3.2. Les encodages ISO-latin

Dans les années 1990, pour satisfaire les besoins des pays européens, ont été définis plusieurs encodages alternatifs, connus sous le nom de ISO-latin, ou encore ISO-8859-1. Idéalement, on aurait pu et certainement dû définir un seul encodage pour représenter tous les nouveaux caractères; mais entre toutes les langues européennes, le nombre de caractères à ajouter était substantiel, et cet encodage unifié aurait largement dépassé 256 caractères différents, il n'aurait donc pas été possible de tout faire tenir sur un octet.

Mais on a préféré préserver la "bonne propriété" du modèle un caractère = un octet, ceci afin de préserver le code existant qui aurait sinon dû être retouché ou récrit.

Dès lors il n'y avait pas d'autre choix que de définir plusieurs encodages distincts; par exemple pour le français on a utilisé à l'époque ISO-latin-1; pour le russe ISO-latin-5.

ISO-8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
<b>8x</b>	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
<b>9x</b>	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
<b>Ax</b>	NBSP	ı	ç	£	¤	¥	ı	§	¨	©	ª	«	¬	-	®	¯
<b>Bx</b>	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
<b>Cx</b>	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
<b>Dx</b>	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
<b>Ex</b>	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
<b>Fx</b>	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Jeu de caractères étendus pour la France