

CODAGE DE L'INFORMATION

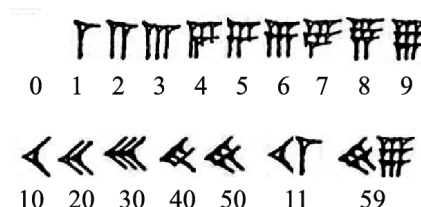
1. Les systèmes de numération

Depuis l'antiquité, les hommes ont essayé de représenter les nombres entiers par des symboles. Il existe essentiellement deux grands principes :

1. chaque symbole possède une valeur, et on ajoute ces valeurs pour obtenir le nombre. Le système de numération des romains repose essentiellement sur ce principe (XXXII=32), avec en plus un aspect soustractif (IX=9).
2. on utilise un nombre petit de symboles (les chiffres) dont la valeur dépend de la position. Chaque décalage vers la gauche du symbole le multiplie par une certaine quantité appelée la **base**. Par exemple, en écriture décimale 2345 signifie $5+4\times 10+3\times 100+2\times 1000$.

Le second système possédant de nombreux avantages par rapport au premier, notamment pour effectuer les opérations, c'est lui qui s'est peu à peu imposé. Nous utilisons actuellement le système décimal (la base est 10) mais ce ne fut pas toujours le cas.

Les babyloniens utilisaient un système **sexagésimal**, ce qui signifie que la base était 60. C'est pourquoi une minute fait 60 secondes !



En informatique, les base 2 (écriture **binaire**) et base 16 (écriture **hexadécimale**, où les chiffres représentant 11, 12,...15, sont notés A,B,...,F) sont très utilisées.

2. Les bases fréquemment utilisées

▪ Le décimal (Base 10) : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Le système décimal est le système universellement utilisé. C'est la base de référence, ce qui signifie qu'un nombre est de manière implicite décimal dès lors qu'il est écrit sans précision de sa base.

▪ Le binaire (Base 2) : {0, 1}

C'est la base de numération couramment utilisée en électronique. C'est un système à base 2 qui est donc composé des caractères 0 et 1. Chacun de ces chiffres est appelé 'Bit', contraction des mots Binary Unit ou Binary Digit.

▪ L'hexadécimal (Base 16). Utilise des chiffres et des lettres : {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Ce système à base 16 est le plus utilisé en électronique numérique car il permet une manipulation de quartets en représentation compacte. Ce qui, dans les systèmes actuels à grande capacité mémoire par exemple, est un avantage non négligeable. La base 16 est une forme contractée de la base 2.

2.1. Notation

Lorsqu'on écrit un nombre, il faut spécifier dans quelle base il est écrit, sinon il y a des risques de confusion. Par convention, on indique la base en indice :

$3A9_{(16)} = (3A9)_{16}$ **base 16 : hexadécimal**

$238_{(10)} = (238)_{10}$ **base 10 : décimal**

$0100\ 1101_{(2)} = (0100\ 1101)_2$ **base 2 : binaire**

En programmation, la notation permettant de voir que l'on est en hexadécimal dépend du langage :

Exemple avec $(AE4F)_{16}$:

Langage	Préfix	Exemple
C, C++, java	0x	0xAE4F
Pascal	\$	\$AE4F
Basic	&h	&hAE4F
HTML	#	#AE4F

2.2. Vocabulaire

On définit alors par « bit », la plus petite unité d'information manipulable par un circuit logique qui est 0 ou 1. Il est possible de traiter des mots de plusieurs bits. Un mot de 4 bits est appelé un quartet, de 8 bits un octet (Byte) et de 16 bits un Word.

MSB			LSB			Bit de rang n
b_5	b_4	b_3	b_2	b_1	b_0	
1	0	1	1	0	1	(2) Base
32	16	8	4	2	1	Poids
2^5	2^4	2^3	2^2	2^1	2^0	

On appelle le bit de rang 0 (b_0) le **bit de poids faible (LSB)** et le bit de rang le plus élevé (ici b_5) le **bit de poids fort (MSB)**.

La formule du poids P est : $P = \text{Base}^{\text{Rang}}$

La formule du nombre de combinaisons possibles est : $N = \text{Base}^{\text{Nbre de Bit}}$

3. Conversion d'un nombre exprimé dans une base B en base 10

$$N(10) = a_0.B_0 + a_1.B_1 + a_2.B_2 + \dots + a_{(n-2)}.B_{(n-2)} + a_{(n-1)}.B_{(n-1)}$$

Avec :

- B : La base du nombre à convertir
- n : Le nombre de chiffres ou de bits du nombre exprimé en base B
- a_i : Le chiffre ou bit de rang i du nombre exprimé en base B
- N : Le nombre exprimé en base B

Exemple :

Soit $N = 483_{(16)}$ à convertir en base 10.

D'après les notations précédentes :

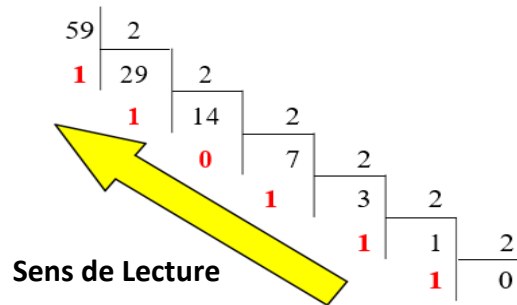
B	n	a_0	a_1	a_2
16	3	3	8	4

$$(N)_{10} = 3 \times 16^0 + 8 \times 16^1 + 4 \times 16^2 = 3 + 128 + 1024 = (1155)_{10} = (483)_{16}$$

4. Passage du décimal vers une base n

Pour convertir un nombre décimal en une base n , on divise ce nombre par n , puis tant que le quotient obtenu est supérieur ou égal à n , on divise à nouveau le résultat par n . Le nombre cherché est donné par les restes des divisions successives et le résultat de la dernière division. Le poids fort du nombre cherché est le résultat de la dernière division et le poids faible le reste de la première division.

Exemple : Soit $N = 59_{(10)}$ à coder en binaire (base 2)



$$N = (59)_{10} = (11\ 1011)_2$$

On peut écrire le résultat sur un octet : $N = (59)_{10} = (0011\ 1011)_2$

5. Passage du binaire à l'hexadécimal

Il faut séparer le nombre exprimé en binaire en paquets de 4 bits en commençant par les bits de poids faibles. Ensuite, on calcule le poids total de chaque paquet de 4 bits pour obtenir le nombre en hexadécimal.

Exemple : soit $N = (0101\ 0101\ 0110\ 1010)_2$ à exprimer en hexadécimal

$(0101)_2$	$= (4 + 1)_{10}$	$= (5)_{10}$	$= (5)_{16}$
$(0101)_2$	$= (4 + 1)_{10}$	$= (5)_{10}$	$= (5)_{16}$
$(0110)_2$	$= (4 + 2)_{10}$	$= (6)_{10}$	$= (6)_{16}$
$(1010)_2$	$= (8 + 2)_{10}$	$= (10)_{10}$	$= (A)_{16}$

$$(0101\ 0101\ 0110\ 1010)_2 = (556A)_{16}$$

6. Passage de l'hexadécimal au binaire

Il faut convertir chaque "chiffre" du nombre hexadécimal en un paquet de 4 bits.

Exemple : Soit $N = 1F4A_{(16)}$ à exprimer en binaire

$(1)_{16}$	$= (1)_{10}$	$= (0001)_2$
$(F)_{16}$	$= (15)_{10}$	$= (1111)_2$
$(4)_{16}$	$= (4)_{10}$	$= (0100)_2$
$(A)_{16}$	$= (10)_{10}$	$= (1010)_2$
$(1F4A)_{16}$	$= (0001\ 1111\ 0100\ 1010)_2$	$= (1\ 1111\ 0100\ 1010)_2$

7. Le code ASCII

L'*American Standard Code for Information Interchange*, ou code ASCII ou ASCII, est une norme informatique pour le codage des caractères. En adoptant le même codage, les systèmes informatiques conçus par n'importe quel fabricant savent ainsi échanger du texte, des nombres, des signes de ponctuation et bien d'autres symboles.

Si l'ASCII, né dans les années soixante, est désormais détrôné par d'autres formes de codage plus étoffées et plus récentes comme l'Unicode, il a servi de base à ses successeurs. Ces derniers restent souvent compatibles avec lui.

Propriétés de l'ASCII :

- L'ASCII se code sur 7 bits.
- Le code ASCII permet de définir 128 codes numériques, donc 128 caractères.
- Les 32 premiers codes, de 0 à 31, ne sont pas des caractères imprimables mais des caractères "de contrôle". Par exemple le code 13 représente un retour à la ligne, et le code 7 fait produire un bip à certains ordinateurs, ce qui s'avérait utile sur les premiers IBM PC pour signaler une erreur, par exemple.
- À partir du code 32, suivent des signes de ponctuation et quelques symboles mathématiques comme ! ou + ou / , puis les chiffres arabes de 0 à 9, ainsi que les 26 lettres de l'alphabet latin, en capitales puis en minuscules.
- Les codes 65 à 90 représentent les lettres de A à Z en capitale, et les codes 97 à 122 les mêmes lettres en minuscule, de a à z. Pour passer d'une lettre capitale à sa minuscule, il suffit d'ajouter 32 en décimal au code ASCII (ou +20 en hexadécimal, ou de modifier le sixième bit en binaire). Donc en décimal, 65 (la lettre A) + 32 = 97 (la lettre a)

Table ASCII :

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

L'ASCII étendu :

Le code ASCII ayant été établi par des Américains, dont la langue ne comporte pas d'accents, les caractères accentués du français et les signes diacritiques nécessaires à d'autres langues n'y étaient pas prévus. C'est ce qui a conduit les organismes de normalisation à définir d'autres codages plus étoffés, qui ajoutent à l'ASCII les caractères qui manquent à une langue, un pays ou une culture.

Par commodité, et comme les 128 premiers caractères restent souvent identiques à ceux de l'ASCII, on parle alors d'ASCII étendu. Cette appellation un peu fourre-tout recouvre en fait différents types de codage, comme la norme ISO-8859-1 sur 8 bits (pour un codage maximal de 256 caractères), qu'on appelle aussi Latin-1 ou Europe occidentale.