Représentation de l'information

Il y a toujours de multiples manières de représenter l'information. Il existe des techniques différentes pour représenter la même information.

Cherche sur internet une représentation peut commune de l'information (autre que l'alphabet romain) et découvre les bases de son fonctionnement, c'est-à-dire ses règles d'écriture et de transmission, ses caractères, etc. (par exemple le langage morse)

Autre modèle de représentation

Les sociétés anciennes disposaient de système de comptage particulier :

Par exemple la numération babylonienne

Chez les Babyloniens (environ 2000 ans av J.C.), les symboles utilisés sont le clou pour l'unité et le chevron pour les dizaines. C'est un système de positon.

A partir de 60, la position des symboles entre en jeu :

204 : 

7392 : 

Le nombre 60 constitue la base de ce système.

Traitement de l'information

Certaines informations peuvent être traitées de deux manières distinctes :

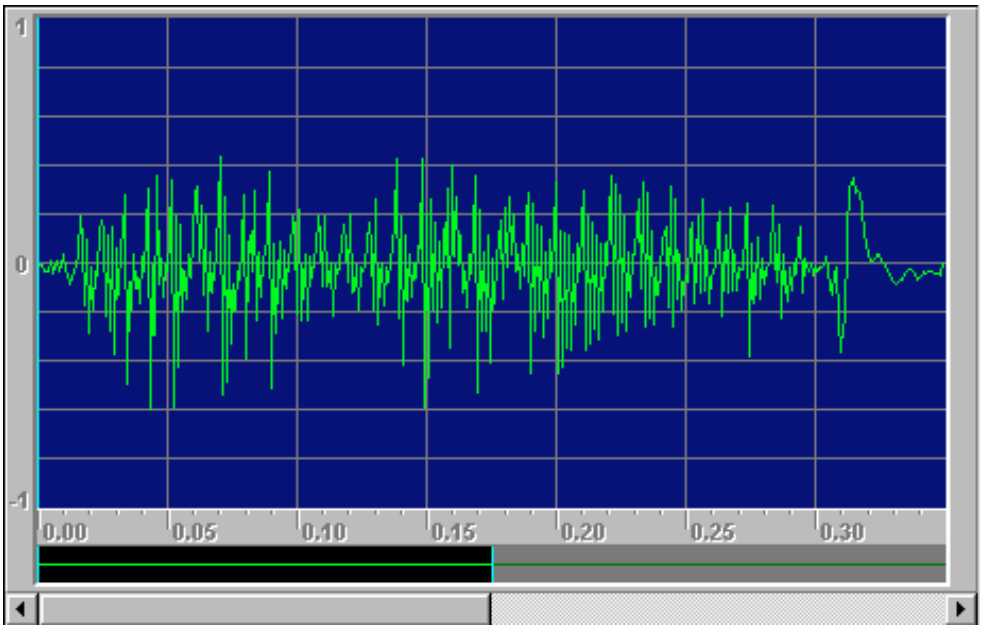
Nous pouvons en faire un affichage « analogique ».

Dans ce cas les valeurs ne sont pas séparées par des sauts : entre deux valeurs A et B il existe un nombre infini d'autres valeurs.

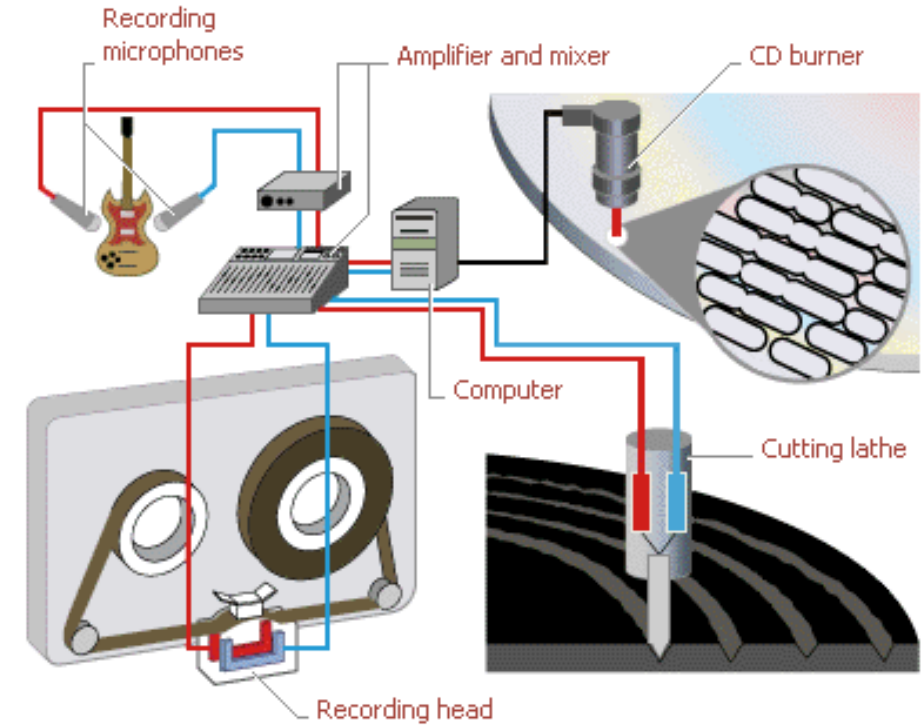
Nous pouvons également en faire un affichage « digitale » (numérique).

Dans ce cas une valeur est représentée par une chaîne finie de symboles appelés digits. Il est impossible de représenter digitalement tous les nombres existants entre deux points d'une échelle analogique

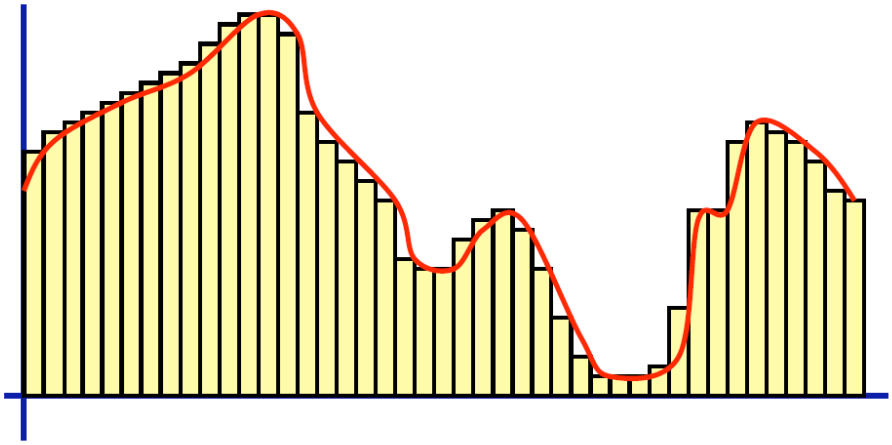
Un autre exemple est l'enregistrement du son



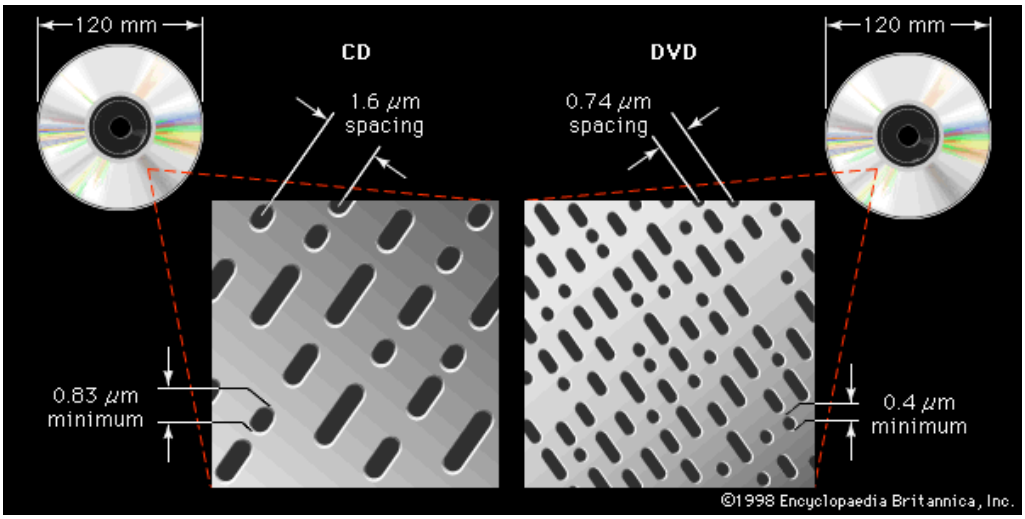
Représentation d'une onde sonore



Les différentes méthodes d'enregistrement du son (analogique ou numérique)



Pour enregistrer sur un CD, le son est échantillonné 44'100 fois par seconde. La valeur de chaque échantillon est stockée en binaire, à l'aide de 16 digits (bits): il n'y a que 65'536 valeurs possibles (2^{16})



Si le temps maximal d'enregistrement sur un CD est de 74 minutes, le nombre maximal de bits stockés dans un CD est donc de :

(44100 échantillons/sec)*(16 bits)*(2 canaux)*(74*60 sec)= 6'265'728'000 bits = 783'216'000 bytes

(1 byte = 8 bits)

Toute information dans un système informatique est représentée sous la forme d'un paquet de bits.

La différence entre un type d'information et un autre est donnée seulement par le contexte : la même séquence de bits peut représenter un nombre entier, un nombre réel, un caractère, une instruction, un son, etc

information = bits + contexte

Représentation des nombres naturels

Le système de numération romain a été heureusement remplacé par un système de numération de position dans une base choisie (normalement la base 10)

Exemple :

MCMLIII = 1953

$1953 = 1x10^3 + 9x10^2 + 5x10^1 + 3x10^0$

1953 = 1x1000 + 9x100 + 5x10 + 3x1

1953 = 1000 + 900 + 50 + 3

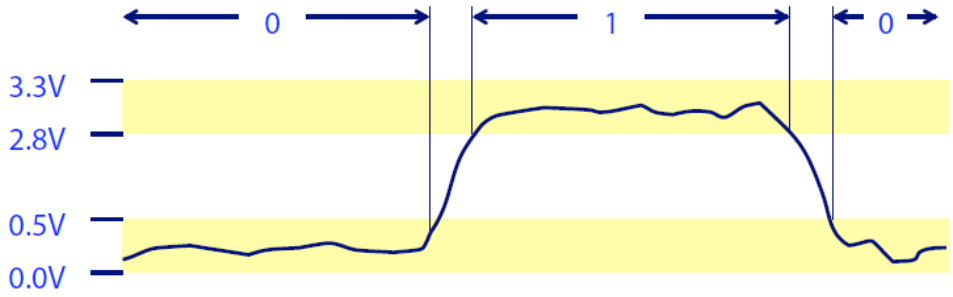
Dans ce mode de représentation, en base n on utilise n symboles (chiffres) différents. Mais la valeur du chiffre change selon sa position.

Si, dans la vie courante, nous utilisons la base 10, les ordinateurs utilisent la base 2.

Les problèmes de la base 10 sont :

- Difficulté de stockage
- Difficulté de transmission des 10 niveaux de signal nécessaires
- Difficulté d'implémentation des fonctions logiques et arithmétiques

Par contre, la base 2 est facile à stocker, à l'aide d'éléments électroniques bistables, et sa transmission est fiable, même sur des environnements bruyants et imprécis.



Conversion

Passage de binaire à décimal

Méthode :

$01101011 = 0x2^7+1x2^6+^{1x25}+0x2^4+1x2^3+0x2^2+1x2^1+1x2^0$

$= 64+32+8+2+1$

$= 107$

Exercice :

Convertir les valeurs suivantes : 10100010, 00110110, 11110000, 11111110, 10010110

Passage d'hexadécimal à décimal

Méthode :

$A8CE = 10x16^3+8x16^2+12x16^1+14x16^0$

$= 40960+2048+192+14$

$= 43214$

Exercice :

Convertir les valeurs suivantes : CAFE, 1234, B524, 3EF2, AA11, ACDC

Passage de décimal à binaire

Méthode :

Pour trouver le binaire d'un nombre décimale il faut utiliser le modulo (le restant d'une division)

193 = ?

193	2						
1	96	2					
	0	48	2				
		0	24	2			
			0	12	2		
				0	6	2	
					0	3	2
						1	1

S'écrira donc : 11000001_2

Exercice :

Convertir les valeurs suivantes : 33, 122, 255, 77, 128, 93

Passage de décimal à hexadécimal

Méthode :

923 = ?

923	16	
11	57	16
	9	3

$923 = 39B_{16}$

Passage de binaire à hexadécimal

Méthode :

Il peut être intéressant de passer par une valeur intermédiaire en décimale

1011	1010	0010	1110
11	10	2	13
B	A	2	E

Exercice :

Convertir ne hexadécimal le code suivant : 1001 1110 0101 0011

Addition de nombre binaire

Méthode :

On procède comme dans le système décimal, une valeur supérieure à 1 fait reporter le ressentant sur la colonne suivante.

	1	0	0	1	1	1	1
	1	0	1	0	0	1	0
1	0	0	1	0	1	0	1

Exercice :

Additionner

1011 + 0101	110010 + 010111	011010 + 101110 + 110011

Format de la représentation

Les ordinateurs ont un format pour représenter les nombres, c'est à-dire un nombre de chiffres préétabli. Pour cette raison, il est parfois utile d'écrire également les zéros à gauche

Exemple : en base 2, sur 4 chiffres, on peut représenter les naturels entre 0 et 15

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

Il est possible d'excéder la capacité de représentation d'un ordinateur (overflow), lors d'une addition par exemple :

10 + 9 = 19

```
  1010
+1001
------
 10011
```

Bit perdu à cause du dépassement de capacité

On appelle généralement "taille d'un mot" le nombre de bits utilisés par un ordinateur pour stocker un entier.

La plupart des ordinateurs possèdent des mots 32 bits, bien que la tendance est d'aller vers les 64 bits.

Le nombre de bits d'un mot limite donc la taille maximale de la mémoire d'un ordinateur.

Si un ordinateur utilise des mots de 32 bits, la taille maximale de sa mémoire est de 2^{32} bytes.

C'est-à-dire 4 gigabytes : $2^{32} = 2^2 x 2^{30} = 4GB$

Crédit : D.Etiemble « représentaiton de l'information », Eduardo Sanchez « représentation de l'information » et ??? « Codage »