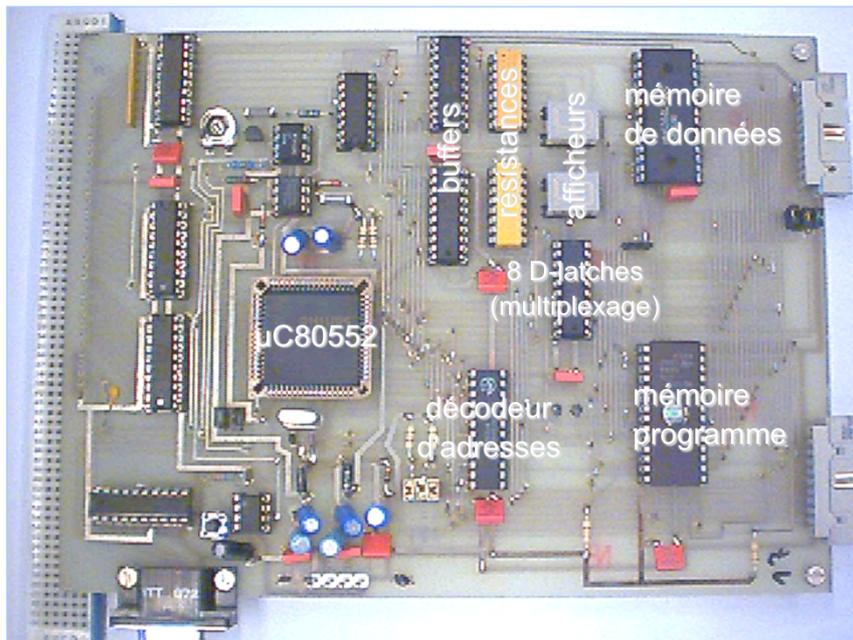


# **Chapitre 10: L'électronique numérique, c'est quoi?**

## **10.1 - Introduction**

## L'électronique numérique: c'est quoi?



2

### L'électronique numérique, c'est quoi?

L'électronique numérique: c'est ça!

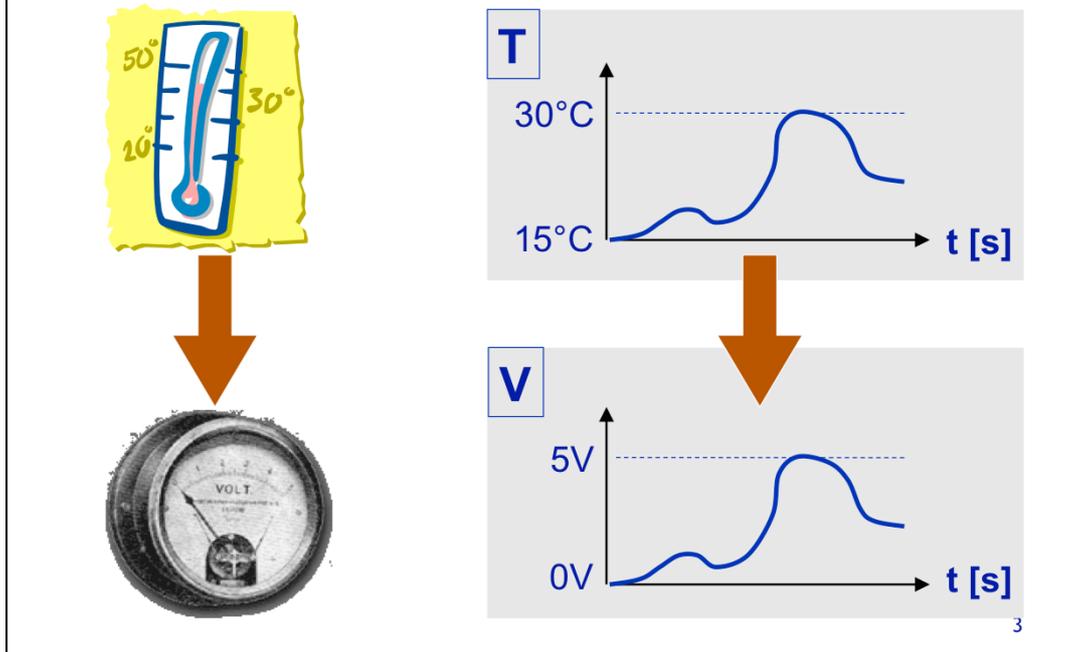
"Ca", c'est la **carte à microcontrôleur 80C552** (\*) que vous utiliserez au laboratoire cette année pour illustrer la partie "numérique" du cours ELECH301.

Cette carte est, comme tout montage électronique, un assemblage de composants sur un circuit imprimé. Son principe de fonctionnement est tout-à-fait similaire au principe d'autres systèmes numériques que sont par exemple votre PC ou votre GSM.

Pour tenter de comprendre ce principe, nous allons maintenant entamer un cheminement en plusieurs chapitres...

(\*) un microcontrôleur est une variante de microprocesseur

## Pour rappel: en électronique analogique, la représentation de l'info est directe



### **En électronique analogique, la représentation de l'information est beaucoup plus directe**

Pour rappel: en électronique analogique, on représente une information (en principe analogique) en traduisant directement cette information sous forme d'une tension (ou d'un courant), analogique également.

La traduction de l'information en signal électrique est donc "directe" puisque les deux grandeurs sont de même nature: analogique.

En électronique numérique au contraire, la tension (bien que fondamentalement analogique) est considérée comme une grandeur logique.

## En électronique numérique, l'info est représentée sous forme de bits et d'octets

The image shows two vertical advertisement panels for computer hardware. The left panel features a blue background with white and yellow text. It highlights a 'VERITABLE' system with an 'intel inside' logo, a '3.0 GHz' processor, and a 'pentium 4' chip. Below this, it mentions 'Processeur Intel Pentium 4 avec technologie Hyper-Threading' and 'Front Side Bus 800 MHz'. Further down, it lists a 'PIONEER Graveur DVD/CD 8x multistandard' and an 'ATI Carte graphique Radeon 9800 XL 128Mo DDR SDRAM interface mémoire 256bits'. At the bottom, it says 'DVD-ROM 16x'. The right panel also has a blue background and lists various components: 'Disque dur 200 Go' (Seagate Ultra ATA 100, 7200 RPM), 'SAMSUNG 512 Mo' (DDR 400MHz, 64-bit), 'MEDION Carte tuner TV et FM stéréo', 'Recompensé par le Reddot Design Award Vainqueur 2003', '6x USB 2.0', '2x FireWire', '8 canaux audio', 'Carte réseau installée', and 'Fax-Modem 56K v.9x PCI'. Several terms like '200 Go', '512 Mo', '128Mo', '256bits', and '10/100 MBit' are circled in orange.

4

### En électronique numérique, l'information est représentée sous forme de bits et d'octets

Un PC est un exemple de système numérique. Les images ci-dessus sont extraites d'une publicité pour un PC en 2004. Les notions de *bit* et d'*octet* y sont omniprésentes ("Mo" est l'abréviation de mégaoctet et "Go" de gigaoctet). Elles sont effectivement typiques de l'électronique numérique.

Les notions de *bit* et d'*octet* sont liées à la manière dont l'information (à traiter par le système électronique) est représentée dans le signal électrique.

**La différence fondamentale entre électronique analogique et électronique numérique, c'est la manière dont l'information est représentée dans le signal électrique.**

## Premier avantage du numérique: l'information ne se dégrade pas...

---



5

Avant d'expliquer en détail ce que sont ces bits et ces octets, c'est-à-dire le principe même de l'électronique numérique, voyons d'abord quels en sont les avantages fondamentaux:

### **Premier avantage du numérique: l'information ne se dégrade pas**

En électronique analogique, tout traitement de l'information (y compris une simple copie) dégrade cette information.

En électronique numérique au contraire, l'information peut être traitée ou copiée (même un grand nombre de fois) sans être dégradée (\*).

Cette "non-dégradation" apporte un avantage décisif à l'électronique numérique, notamment dans le fait que l'information peut être dupliquée et propagée beaucoup plus facilement et de manière quasiment illimitée.

L'échange de fichiers numériques entre utilisateurs, opération banale s'il en est aujourd'hui, est une illustration de ce phénomène. Cette possibilité de copie et diffusion quasi illimitée de l'information numérique pose notamment des problèmes de droits d'auteurs quand cette information constitue une œuvre artistique. Ce type de problème ne se posait pas de manière aussi aigüe lorsque les supports étaient analogiques.

(\*) Cette affirmation est en fait approximative: plutôt que de dire que l'information n'est pas dégradée par un traitement ou une copie, il serait plus exact de dire que l'information est dégradée mais qu'elle peut être restaurée. Nous y reviendrons.

## Second avantage du numérique: le traitement peut être très complexe



6

### **Second avantage du numérique: le traitement peut être très complexe**

Le second avantage d'une représentation numérique (=sous forme de bits et d'octets) de l'information, c'est qu'on peut faire subir à cette information des traitements beaucoup plus complexes qu'en analogique.

Ceci est directement lié au fait qu'on peut copier l'information sans la dégrader (voir page précédente):

- en analogique, le nombre de traitements successifs applicables à un signal est limité puisque chacun de ces traitements dégrade un peu l'information
- en numérique au contraire, le nombre de traitements successifs est virtuellement illimité, ce qui permet des traitements beaucoup plus complexes qu'en analogique.

## Troisième avantage du numérique: on peut programmer!

---



### Troisième avantage du numérique: on peut programmer!

Notre propos ici n'est pas d'affirmer que programmer est amusant, mais plutôt d'insister sur le fait qu'en électronique numérique il existe des circuits qui peuvent être programmés (entre autres: les microprocesseurs)...

Un "programme", c'est un ensemble d'instructions (c'est-à-dire d'informations) qui **définissent** le traitement à effectuer sur les informations (données).

Un circuit qui peut être programmé n'a donc pas de fonction fixe, ce qui lui confère une flexibilité énorme en termes d'applications.

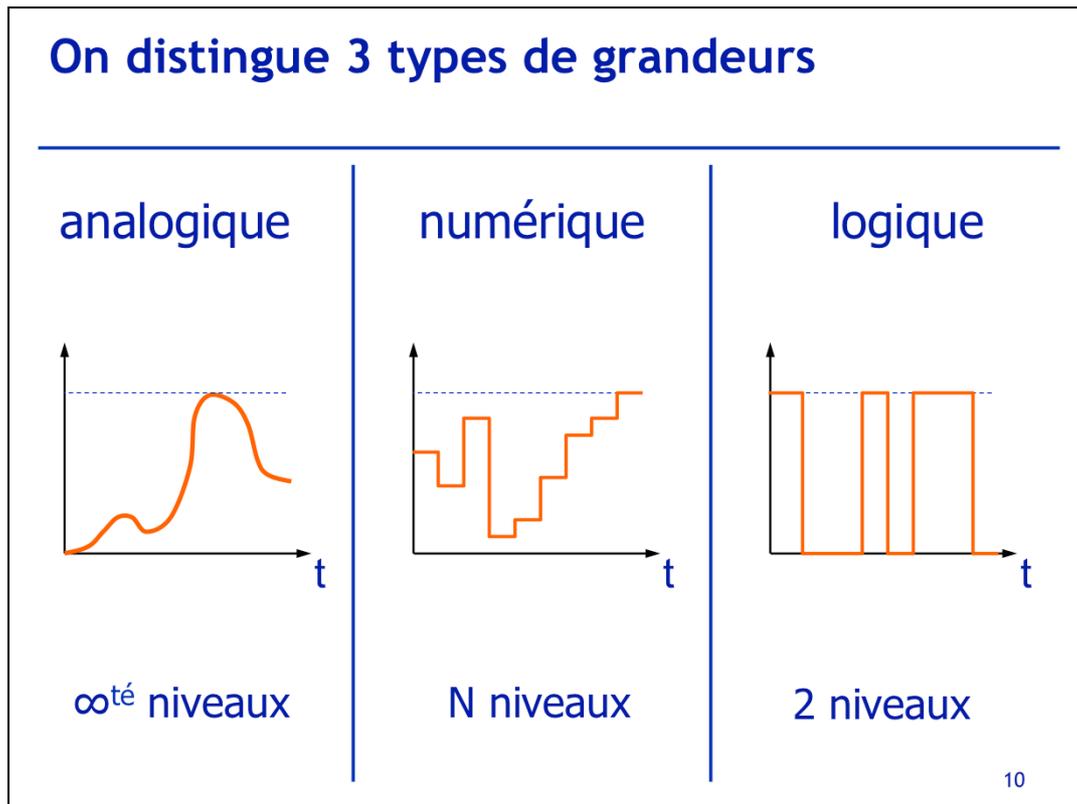
Cette possibilité (même si elle est devenue banale) est remarquable: en numérique, on peut donc indiquer via un programme, c'est-à-dire après sa fabrication, ce qu'un circuit doit faire.

# Chapitre 10: L'électronique numérique, c'est quoi?

## 10.2 - Bits et octets

12/09/12

9



### On distingue trois types de grandeurs

De manière tout-à-fait générale (indépendamment de l'électronique), on distingue trois types de "grandeurs" ou de "variables":

- on dit d'une variable qu'elle est **analogique** lorsqu'elle varie de manière *continue*, ou encore qu'elle peut prendre une *infinité* de valeurs différentes sur un intervalle.
- on dit d'une variable qu'elle est **numérique** lorsqu'elle varie de manière discrète (ou quantifiée), ou encore qu'elle ne peut prendre qu'un nombre *fini* de valeurs différentes (appelées **états**) sur un certain intervalle.
- on dit d'une variable qu'elle est **logique** lorsqu'elle ne peut prendre que *deux* valeurs différentes. Une variable **logique** est donc un cas particulier de variable numérique (2 états).

Pour comprendre vraiment ce qu'est l'électronique numérique, il est fondamental de bien faire la distinction entre:

- l'information qu'on veut représenter
- et le signal électrique qui est le moyen concret de représenter cette information

Comme le montrent les pages suivantes:

- l'information en elle-même peut être soit analogique, soit numérique, soit logique
- le signal électrique (tension ou courant) est toujours de nature fondamentalement analogique

## Une grandeur *analogique* varie de manière continue

---



11

### Une grandeur analogique varie de manière continue

Voici quelques exemples de grandeurs analogiques:

- la plupart des grandeurs physiques sont analogiques: température, temps, vitesse, position, pression, etc (\*)
- une variable  $x$  réelle (ensemble  $\mathbb{R}$ )

(\*) nous négligeons ici les effets quantiques (qui disent précisément qu'à petite échelle ces grandeurs sont quantifiées... donc en fait numériques)

## Une grandeur *numérique* varie de manière discrète (N états)



6 états

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
<b>O</b>	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>
<b>] ^</b>	<b>- `</b>	<b>a</b>		
<b>k</b>	<b>l</b>	<b>m</b>	<b>n</b>	<b>o</b>
<b>y</b>	<b>z</b>	<b>{</b>	<b> </b>	<b>}</b>

26 états



52 états

12

### Une grandeur numérique varie de manière discrète (N états)

Toute variable qui ne peut prendre qu'un nombre fini de valeurs est une variable numérique. Ces valeurs en nombre fini sont appelées "**états**".

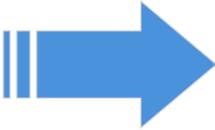
Voici quelques exemples de variables numériques:

- une variable désignant une lettre de l'alphabet (26 états)
- une variable désignant une carte dans un jeu de cartes (52 états)
- l'état d'un feu rouge (3 états)
- une variable désignant le jour de l'année (365 états)
- une variable désignant un signe du zodiaque (12 états)
- une variable  $x$  entière (ensemble  $\mathbb{N}$  ou  $\mathbb{Z}$ , pour peu qu'on se limite à un intervalle)

Il existe en fait énormément de variables numériques dans la vie courante...

## Une grandeur *logique* varie entre 2 états

---

 vrai	 on	 gauche
 faux	 off	 droite

13

### Une grandeur logique varie entre 2 états

Une grandeur logique est un cas particulier de grandeur numérique: celui où il n'y a que deux valeurs possibles.

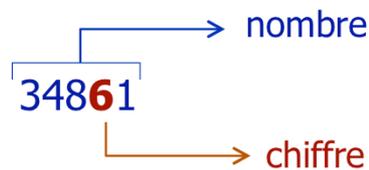
Une des grandeurs logiques les plus évidentes, c'est l'opposition "vrai/faux". Une grandeur qui est soit vraie soit fausse est une grandeur logique booléenne, en référence à l'algèbre de Boole qui est la branche des mathématiques traitant des propositions logiques (c'est-à-dire qui sont soit vraies soit fausses).

Mais il existe de très nombreux autres exemples de grandeurs logiques. En particulier toutes les oppositions de deux concepts peuvent être traduites sous forme de grandeur logique. Voici quelques exemples:

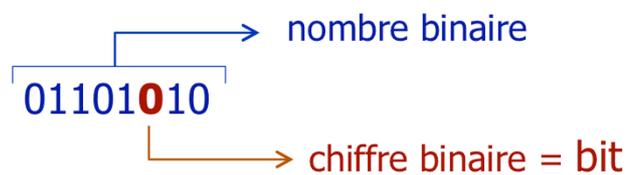
- une lampe est soit allumée soit éteinte
- un interrupteur également
- etc

## Un *bit* est d'abord un chiffre binaire...

décimal = 10 valeurs : 0 1 2 3 4 5 6 7 8 9



binaire = 2 valeurs : 0 1



14

Voyons maintenant ce qu'est un bit.

### Un bit est d'abord un chiffre binaire...

Revenons un instant sur une évidence: nous utilisons pour compter le *système décimal* (encore appelé *base 10*): en d'autres termes nous utilisons des *chiffres* qui peuvent prendre 10 valeurs différentes (de 0 à 9).

Pour compter au-delà de 9, nous combinons plusieurs chiffres pour former des *nombres* appelés nombres décimaux.

Il est possible d'utiliser d'autres bases: rien n'empêche de compter et de calculer en utilisant des chiffres qui peuvent chacun prendre davantage ou au contraire moins de 10 valeurs.

C'est ce qu'on fait en électronique numérique, où l'on utilise principalement le système binaire (ou base 2) dans lequel chaque chiffre ne peut prendre que deux valeurs différentes: 0 et 1.

Un bit n'est rien d'autre qu'un chiffre en base 2 ("Bit" est d'ailleurs l'acronyme de "binary digit", ce qui signifie bien "chiffre binaire").

En combinant plusieurs bits, on forme des nombres binaires, ce qui est bien utile pour compter au-delà de... 1.

## Un *bit* est aussi une grandeur logique

---

 vrai	 on	1
 faux	 off	0

15

### Un bit est aussi une grandeur logique

Indépendamment du fait qu'un bit est un chiffre, un bit ne peut prendre que deux valeurs différentes. Un bit est donc également une variable logique.

Bien que le bit (valant 0 ou 1) et la variable booléenne (valant "vrai" ou "faux") ne sont pas des notions rigoureusement identiques, il est très courant de confondre en pratique ces deux notions en associant:

- d'une part l'état "1" à l'état "vrai"
- d'autre part l'état "0" à l'état "faux"

Un bit est donc à la fois une variable logique (une variable à 2 états) et une variable mathématique (un chiffre), deux types de variables auxquelles s'appliquent des traitements a priori différents. Cette double personnalité confère au bit des propriétés tout-à-fait intéressantes: le bit jette en quelque sorte un "pont" entre la logique et les mathématiques.

## Un mot de $N$ bits est un groupe de $N$ bits et peut représenter $2^N$ états

1 bit	2 bits	3 bits	4 bits	$N$ bits
		000	0000	1000
		001	0001	1001
	00	010	0010	1010
0	01	011	0011	1011
1	10	100	0100	1100
	11	101	0101	1101
		110	0110	1110
		111	0111	1111
				↓
2 états	4 états	8 états	16 états	$2^N$ états

16

### Un "mot de $N$ bits" est un groupe de $N$ bits

Comme on l'a dit, un bit ne permet de compter que jusqu'à...1. Il est donc courant qu'on associe plusieurs bits soit pour représenter un nombre plus grand (on a donc dans ce cas constitué un nombre binaire) soit pour représenter une variable *numérique* (à plus de deux états).

Un groupement de  $N$  bits est appelé un "mot de  $N$  bits". Le nombre de bits est le plus souvent un multiple de 2.

### Un mot de $N$ bits peut représenter $2^N$ états

Combien d'états peut-on représenter avec  $N$  bits? Il est facile de montrer (par dénombrement: voir image ci-dessus) que  $N$  bits permettent de représenter  $2^N$  états différents.

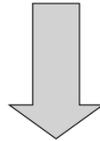
Il est intéressant dans ce contexte de mémoriser les principales puissances de 2:

nombre de bits $N$	nombre d'états $2^N$	multiple
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	
8	256	
9	512	
10	1024	=1K ("Kilo")
12	4096	
16	65536	
20	1048576	=1M ("Méga")
24	16777216	
30	1073741824	=1G ("Giga")
32	environ $4 \cdot 10^9$	
64	environ $1,8 \cdot 10^{19}$	

## Un octet (ou *byte*) est un mot de 8 bits

---

00000000



$2^8 = 256$  états

11111111

17

### Un octet est un mot de 8 bits

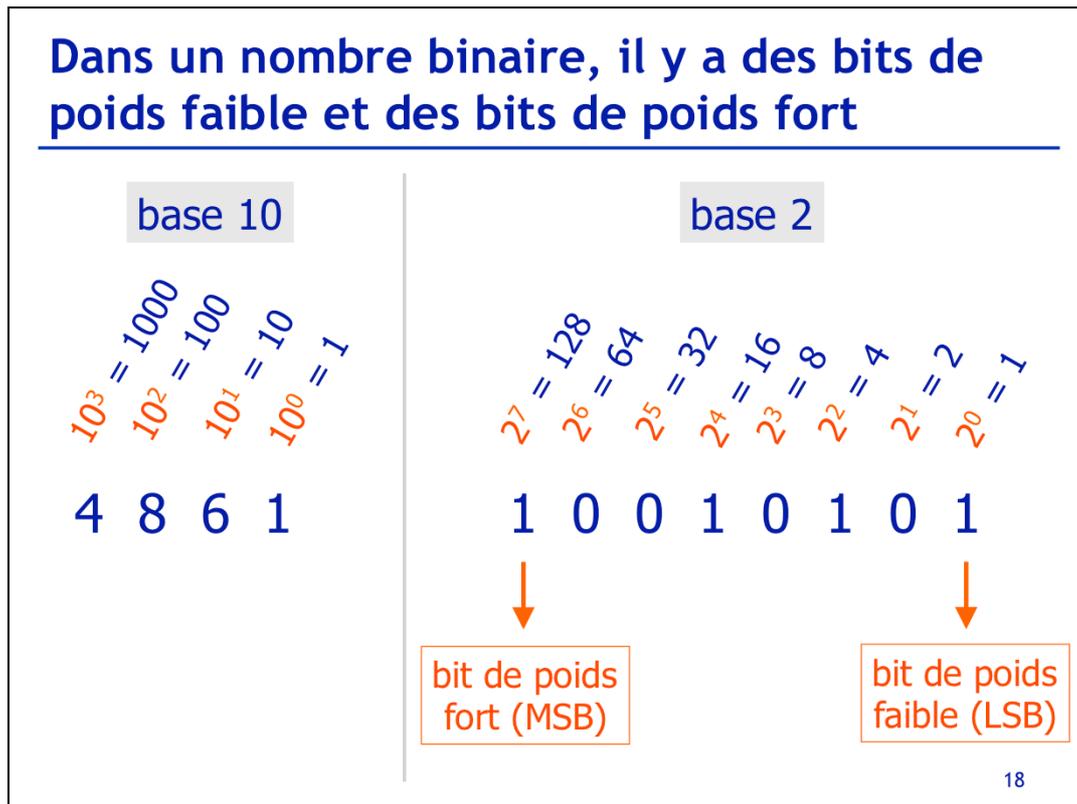
En pratique, on utilise très souvent des mots de 8 bits. Ceux-ci ont donc reçu le nom particulier d'octet.

Un octet permet de représenter  $2^8=256$  états différents, qui peuvent être par exemple:

- les nombres entiers de 0 à 255
- les nombres entiers de -128 à 127
- un jeu complet de caractères (majuscules, minuscules, ponctuation, etc) sur un PC
- etc

Le terme "byte" est la traduction anglaise de "octet". Un byte vaut donc 8 bits: attention à ne pas confondre les deux notions (bit et byte) ! L'unité "Kb" désigne un kilobyte.

On utilise également souvent des groupes d'octets: typiquement des mots de 16, 24, 32 ou 64 bits.



Pour rappel (à gauche): dans un nombre décimal, à chaque chiffre est associée une puissance de 10 (puisque l'on est en base 10) qui représente le "poids" de ce chiffre au sein du nombre:

- les unités ont un poids de  $1=10^0$
- les dizaines ont un poids de  $10=10^1$
- les centaines ont un poids de  $100=10^2$
- etc

Un nombre binaire, par exemple un octet, est construit selon le même principe: à chaque bit (chiffre binaire) est associé un poids de valeur  $2^B$  (où B représente la place du bit dans le nombre binaire, avec B=0 pour le bit le plus à droite). On a donc:

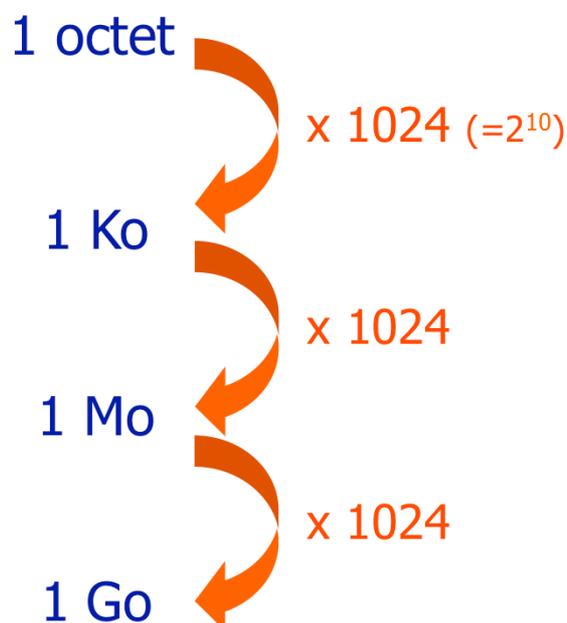
- le chiffre le plus à droite a un poids  $2^0=1$
- le chiffre suivant (vers la gauche) a un poids  $2^1=2$
- le chiffre suivant a un poids  $2^2=4$
- etc
- dans un mot de N bits, le chiffre le plus à gauche a un poids  $2^{N-1}$

On appelle "**bits de poids faible**" le ou les bits situés le plus à droite du nombre binaire et "**bits de poids fort**" le ou les bits situés le plus à gauche du nombre binaire.

En anglais, on appelle en particulier:

- LSB** (Least Significant Bit = bit le moins significatif) le bit le plus à droite (de poids 1)
- MSB** (Most Significant Bit = bit le plus significatif) le bit le plus à gauche (de poids  $2^{N-1}$ )

## Ko, Mo, Go sont les multiples de l'octet



19

### **Ko, Go, Mo sont les multiples de l'octet**

Comme un octet ne permet de stocker que 256 états différents, il est courant d'utiliser d'énormes quantités d'octets dès qu'on doit stocker des informations (voir page suivante).

Pour désigner de grandes quantités d'octets, on utilise donc les multiples "kilo", "méga" et "giga" (comme on le fait pour les grandeurs physiques).

En électronique numérique, ces préfixes n'ont cependant pas exactement la même signification que pour les grandeurs physiques:

- un Kilo-octet désigne  $2^{10}=1024$  octets (et non 1000) [\*]
- un méga-octet désigne 1024 kilo-octets (soit  $1024^2 = 2^{20} = 1048576$  = environ 1 million d'octets)
- un giga-octet désigne 1024 méga-octets soit ( $1024^3 = 2^{30} = 1073741824$  = environ 1 milliard d'octets)

Pourquoi se compliquer la vie en utilisant un rapport de 1024 plutôt que le rapport de 1000 classiquement utilisé pour les grandeurs physiques?

Si l'on a choisi un rapport de 1000 pour les grandeurs physiques, c'est parce que c'est un chiffre rond. Mais ce chiffre n'est rond qu'en base 10. Or comme on utilise la base 2 en électronique numérique (où 1000 ne tombe pas rond!), on a préféré utiliser le nombre rond le plus proche mais en base 2, c'est-à-dire 1024: 1024 est en effet la dixième puissance de 2, ce qui s'écrit en binaire: 1000000000.

N.B.: les préfixes "kilo", "méga" et "giga" peuvent aussi être utilisés pour les bits (un kilobit = 1024 bits).

[\*] Pour faciliter la distinction, on réserve parfois "kilo" (avec une minuscule) au facteur 1000 et "Kilo" (avec une majuscule) au facteur 1024. Certains auteurs ont même défini les unités alternatives suivantes: le "kibi" (qui vaut 1024 alors que le "kilo" vaut 1000), le "mébi", le "gibi", etc. On parlerait donc de "kibibyte". Mais ces définitions sont très peu utilisées en pratique.

## L'octet sert d'unité de mesure des quantités d'information numérique



page texte brut  
2Ko



RAM  
4 Go



carte flash  
2Go à 32Go



CD: 650Mo



DVD: 4,7 à 17Go



HD: 500Go 20

### L'octet sert d'unité de mesure des quantités d'information numérique

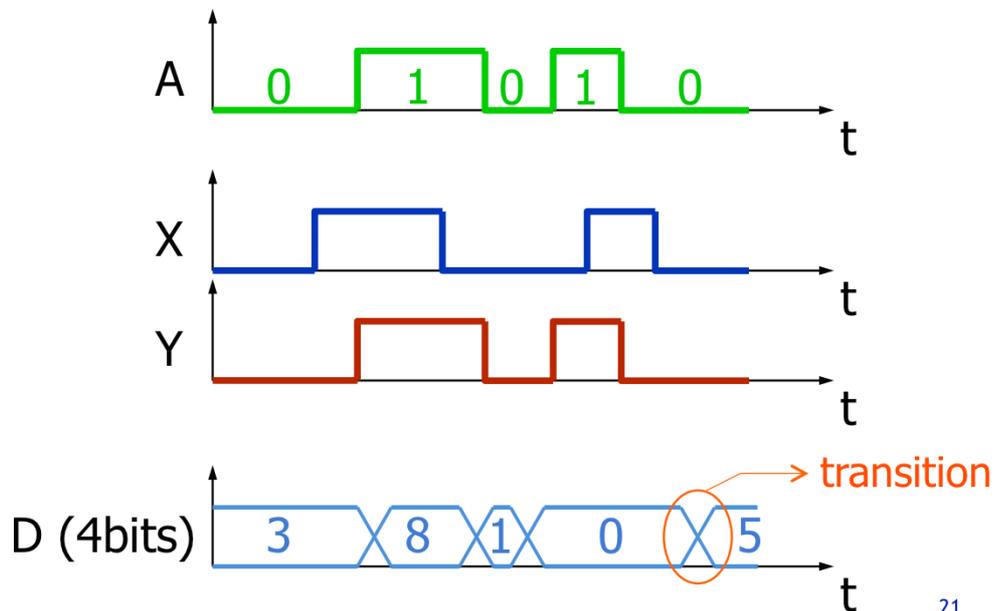
En pratique, un octet peut aussi être vu comme l'unité de base permettant d'évaluer une quantité d'information mise sous forme numérique.

On voit ci-dessus des supports numériques courants et la quantité d'information nominale qu'ils peuvent contenir.

Voici quelques exemples de quantités de données tirées du monde de l'informatique (chiffres actualisés en 2011, voir aussi la figure 9.1):

- une page de texte "brut" (sans mise en page): environ 2Ko
- mémoire RAM d'un PC: typiquement quelques Go
- disquette (obsolète): 1,44Mo
- disquette de grande capacité type "Zip" (obsolète): 250Mo
- carte mémoire de type "flash" (p.ex. pour appareil photo numérique): entre 2Go et 32Go
- disque dur: quelques centaines de Go à quelques To
- CD-ROM: 650Mo
- DVD: 4,7Go à 17Go (suivant la technologie)

## Un *chronogramme* représente la valeur d'un ou plusieurs bits en fonction du temps



21

### Un chronogramme représente la valeur d'un ou plusieurs bits en fonction du temps

En électronique numérique, on s'intéresse souvent à la variation des signaux au cours du temps. Un chronogramme est simplement un graphe montrant l'évolution d'un bit ou plusieurs bits au cours du temps.

Les trois premiers chronogrammes montrent l'évolution d'un seul bit chacun (A, X et Y désignent les signaux correspondants): on y voit bien que chaque signal ne peut prendre que deux valeurs (0 ou 1).

Le dernier chronogramme montre l'évolution de la valeur d'un mot de 4 bits. Pour éviter de devoir représenter les 16 niveaux, on indique simplement la valeur du mot (dans un hexagone plus ou moins étiré). Les transitions (= changement de valeur du mot) sont indiquées par un "croisement" des lignes supérieure et inférieure.

## **Chapitre 10: L'électronique numérique, c'est quoi?**

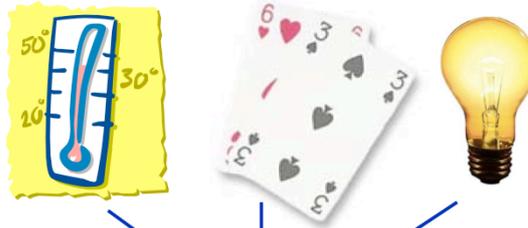
### **10.3 - Principe de l'électronique numérique**

12/09/12

23

## Le principe fondamental de l'électronique numérique consiste à...

1) représenter toute information sous forme de bits



2) représenter chaque bit sous forme de signal électrique (analogique!)

00010010

0V

5V

24

### Le principe fondamental de l'électronique numérique consiste à...

#### 1) représenter toute information sous forme de bits

...que cette information soit logique, numérique ou analogique

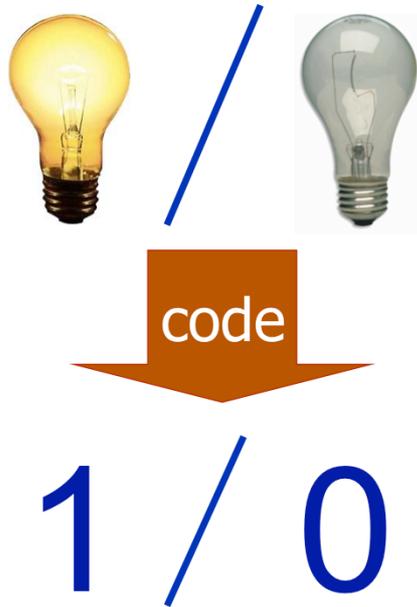
#### 2) représenter chaque bit sous forme de signal électrique

Chaque bit issu de l'étape 1 est traduit sous forme de signal électrique. La traduction se fait en associant à chaque état du bit (0 ou 1) une valeur bien précise du signal électrique, par exemple 0V et 5V respectivement pour les états 0 et 1.

Ces deux étapes vont maintenant être décrites en détail.

## 1) représenter une information logique: 1 bit + 1 code

---



25

Voyons d'abord comment représenter une information (quelle que soit sa nature) sous forme de bits.

Trois cas sont possibles suivant que l'information est logique, numérique ou analogique...

### **1) représenter une information logique: 1 bit + 1 code**

Si l'information est elle-même logique (c'est-à-dire à 2 états), la représenter sous forme de bits est trivial: un bit suffit.

Pour pouvoir interpréter ce bit, il faut définir le code utilisé, c'est-à-dire la manière dont l'association est faite.

Pour coder l'état d'une ampoule, on peut par exemple décider d'utiliser le code "éteint=0 et allumé=1". On aurait aussi bien pu utiliser le code contraire. Le code doit donc être explicitement connu pour pouvoir interpréter l'état du bit: la notion de **codage** de l'information (sous forme de bits) est donc fondamentale.

## 1) représenter une information numérique: N bits + 1 code

00000



11111



000000



111111

$2^5 = 32$  états  $<$  52 états  $<$   $2^6 = 64$  états

26

### 1) représenter une information numérique: N bits + 1 code

Supposons maintenant que l'information est numérique, c'est-à-dire que la variable à coder possède un nombre d'états fini mais supérieur à 2.

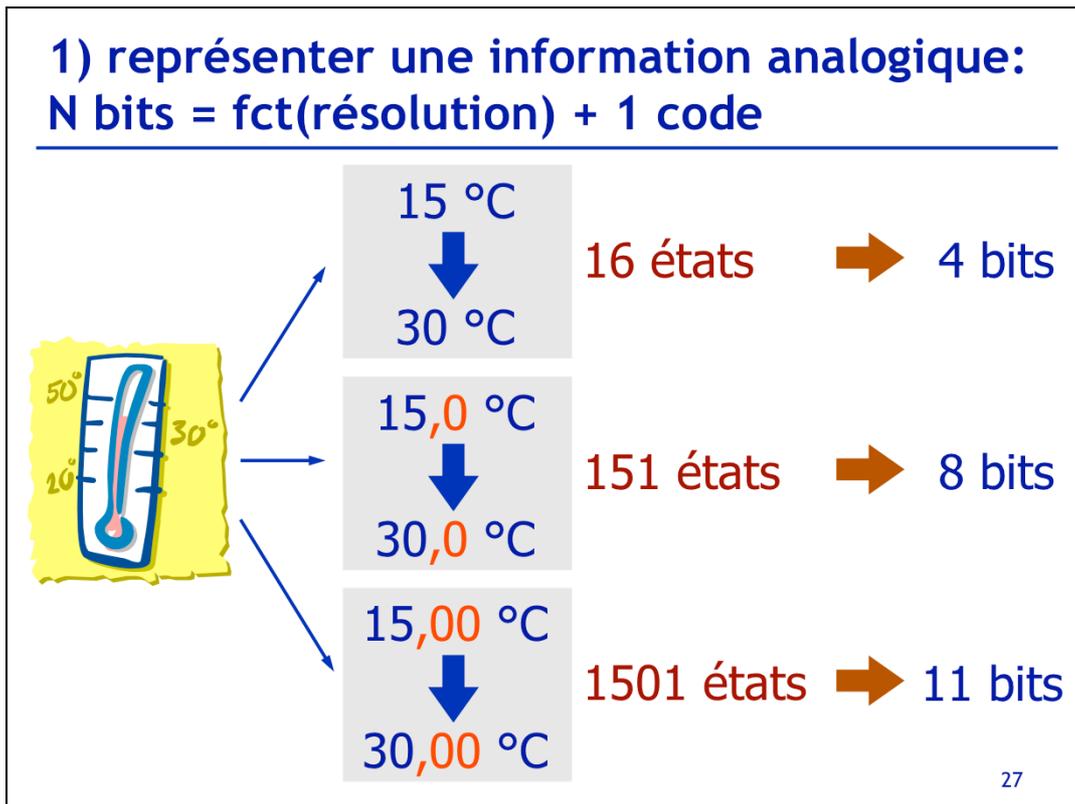
Pour coder cette information sous forme de bits, il suffit de prendre suffisamment de bits.

Puisque un mot de N bits permet de représenter  $2^N$  états:

- pour une information possédant 3 ou 4 états: il faut 2 bits ( $2^2=4$ )
- pour une information possédant entre 5 et 8 états: il faut 3 bits ( $2^3=8$ )
- pour une information possédant entre 9 et 16 états: il faut 4 bits ( $2^4=16$ )
- ...
- pour une information possédant entre 129 et 256 états: il faut 8 bits ( $2^8=256$ )
- ...
- pour une information possédant entre  $(1+2^{N-1})$  et  $2^N$  états: il faut N bits

Exemple: pour représenter la valeur d'une carte dans un jeu de cartes (52 états), il faut 6 bits puisque:  $2^5 < 52 < 2^6$

Dans ce cas-ci aussi, il faut préciser le code utilisé.



**1) représenter une information analogique: N bits =fct (résolution) + 1 code**

Comment peut-on prétendre représenter une information analogique (c'est-à-dire susceptible de varier parmi un nombre infini de valeurs) sous forme d'un nombre fini de bits (susceptible de ne coder que  $2^N$  valeurs)?

En réalité, une variable analogique n'est jamais connue avec une précision illimitée (en d'autres termes: avec un nombre infini de chiffres après la virgule). C'est à la fois impossible et inutile en pratique.

Considérons le cas d'une température: suivant le problème qu'on traite, on peut être amené à la connaître à 1 degré près, ou à 0,1 degré près, ou à 0,01 degré près... mais jamais avec une précision illimitée (ce qui serait par ailleurs impossible à mesurer).

On est donc de fait ramené à considérer un nombre fini de valeurs (donc une information numérique)... même pour une variable analogique;

Le nombre de bits nécessaires dépend donc de la résolution choisie pour l'information analogique. Une fois cette résolution connue, il suffit d'utiliser la règle de la page précédente pour déterminer le nombre de bits requis.

## Pourvu qu'on convienne d'un code, on peut tout représenter en numérique

- code ASCII →
- code BCD
- code GRAY
- complément à 2
- nombres flottants
- etc

MSB \ LSB	0	1	2	3	4	5	6	7
	000	001	010	011	100	101	110	111
0	0000 NUL	DLE	SP	0	@	P	~	p
1	0001 SOH	DC1	!	1	A	Q	a	q
2	0010 STX	DC2	"	2	B	R	b	r
3	0011 ETX	DC3	#	3	C	S	c	s
4	0100 EOT	DC4	\$	4	D	T	d	t
5	0101 ENQ	NAK	%	5	E	U	e	u
6	0110 ACK	SYN	&	6	F	V	f	v
7	0111 BEL	ETB	'	7	G	W	g	w
8	1000 BS	CAN	(	8	H	X	h	x
9	1001 HT	EM	)	9	I	Y	i	y
A	1010 LF	SUB	*	:	J	Z	j	z
B	1011 VT	ESC	+	;	K	[	k	}
C	1100 FF	FS	,	<	L	\	l	
D	1101 CR	GS	=	=	M	]	m	{
E	1110 SO	RS	.	>	N	^	n	~
F	1111 SI	US	/	?	O	_	o	DEL

28

### Pourvu qu'on convienne d'un code, on peut tout représenter en numérique

En conclusion, tout type d'information (logique, numérique ou analogique) peut être représenté sous forme de bits et d'octets.

quelques exemples de codes: code ASCII, code BCD, code GRAY, code en complément à 2, codage des nombres flottants, etc

## 2) représenter un bit sous forme de signal électrique

1	→	5V	3V	0V	1A
0	→	0V	0V	5V	0A

29

### 2) Représenter un bit sous forme de signal électrique

La seconde étape de la représentation d'une information en électronique numérique consiste à traduire chaque bit issu de l'étape 1 en signal électrique.

Il suffit pour cela d'associer à chacun des deux états du bit une valeur précise d'un signal électrique. Il s'agit également à cette étape de convenir d'un "code" ou d'une convention entre la représentation "logique" (bit) et la représentation "électrique" de l'information.

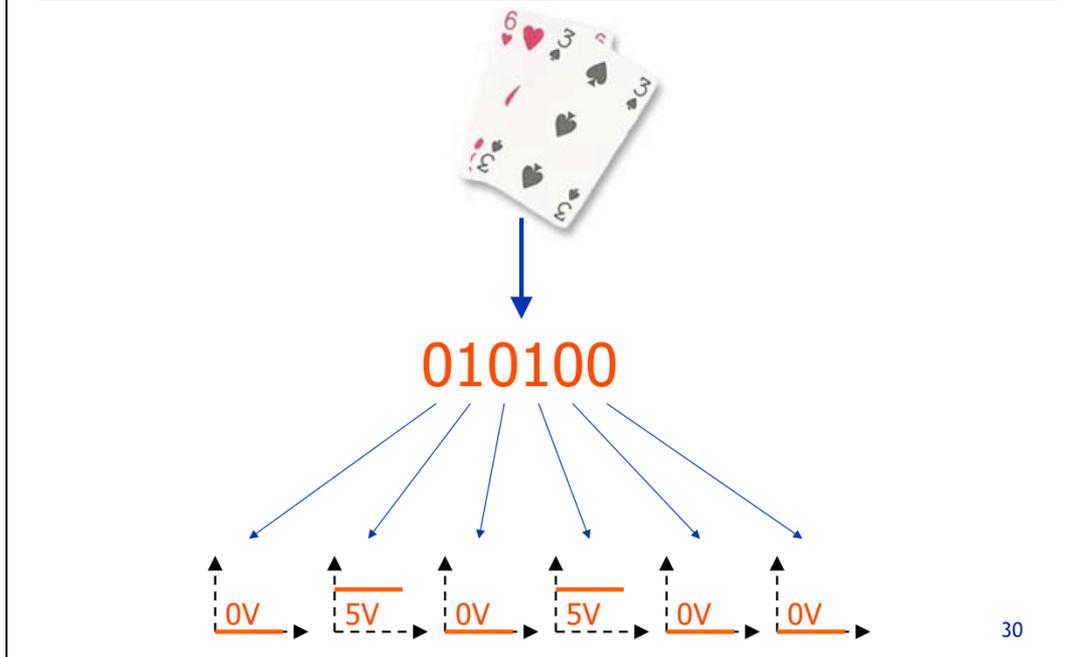
Plusieurs codes sont possibles. Le plus connu est celui de la "logique normale", qui associe un niveau de tension élevé à l'état logique 1 et un niveau de tension faible à l'état logique 0. Les niveaux les plus typiques sont 5V et 0V, mais d'autres valeurs peuvent être utilisées (par exemple 3V/0V ou encore +15V/-15V, etc)

On peut également travailler en "logique inverse", c'est-à-dire associer l'état logique 1 au niveau de tension le plus faible. Cette convention est évidemment moins intuitive.

Enfin on peut utiliser d'autres signaux électriques que la tension: on peut par exemple utiliser un courant.

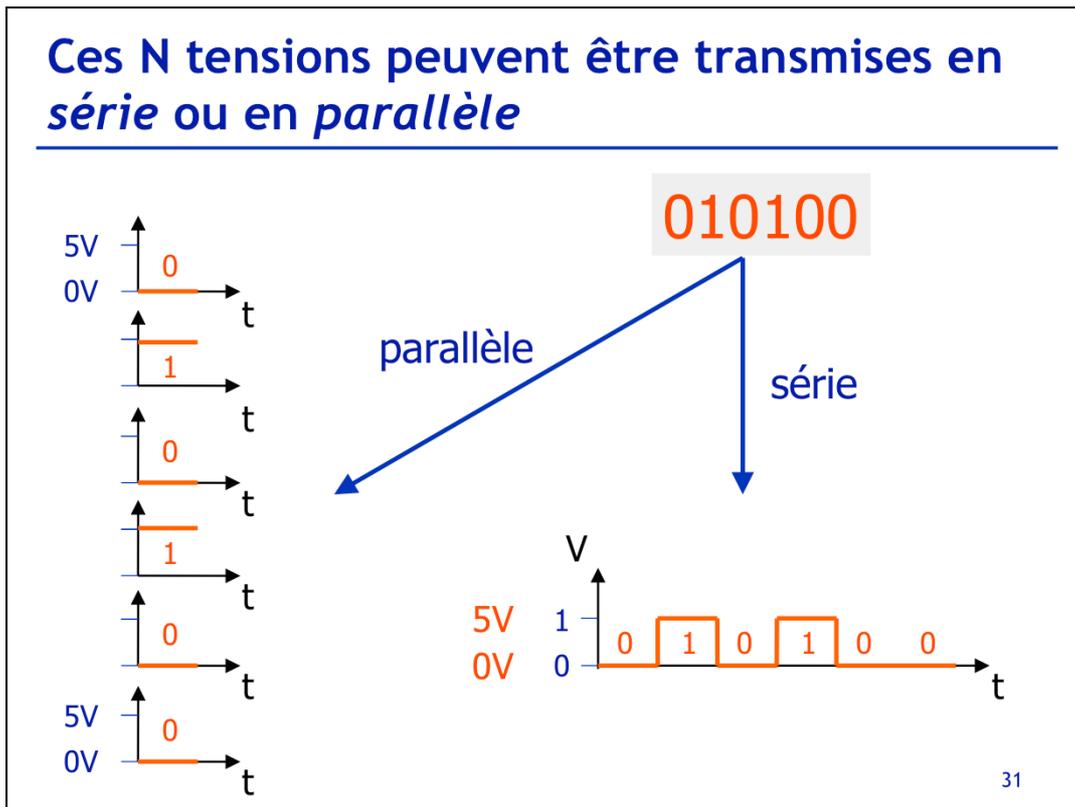
Remarque: lors de cette opération, on fixe par convention le fait que le signal électrique ne peut prendre que deux valeurs. On choisit donc d'interpréter le signal électrique (fondamentalement analogique) comme une valeur... logique. C'est le principe fondateur de l'électronique numérique. C'est également ce principe qui permet la non-dégradation de l'information et confère des avantages majeurs à l'électronique numérique.

## Une information est donc représentée par N tensions "logiques"



### Une information est donc représentée par N signaux électriques "logiques"...

Si l'on fait la synthèse des deux étapes expliquées dans les pages précédentes, on voit donc qu'en électronique numérique une information est représentée, en général, par N tensions. Chacune de ces tensions représente un bit, et peut donc prendre deux valeurs différentes suivant l'état de ce bit.



31

#### Ces N tensions peuvent être transmises en série ou en parallèle

Chacune des tensions représentant un bit est en fait une ddp entre deux fils.

En pratique, il existe deux manières de transmettre ces N ddps:

-la transmission *parallèle* (à gauche) consiste à utiliser N paires de fils [\*] **simultanément**: les ddps respectives, prises simultanément, représentent l'information.

-la transmission *série* (à droite) consiste au contraire à utiliser une seule paire de fils et à lui faire prendre **successivement** les N ddps qui représentent, au total, l'information.

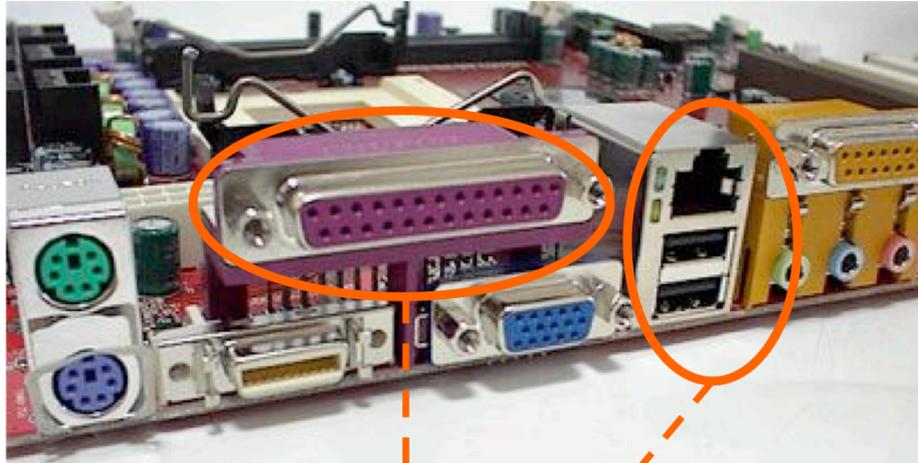
Chaque type de transmission a ses avantages et inconvénients:

-la transmission parallèle est en principe plus rapide mais elle demande davantage de fils [\*]

-la transmission série est en principe plus lente mais elle ne demande que deux fils

[\*] Dans une transmission parallèle, on peut parfois se contenter de N+1 fils (au lieu de 2N): les N ddps peuvent être mesurées par rapport à un même fil commun (=masse = 0V).

## Ces N tensions peuvent être transmises en série ou en parallèle: exemple



port parallèle ←

→ ports série  
USB, FireWire

32

### Ces N tensions peuvent être transmises en série ou en parallèle: exemple

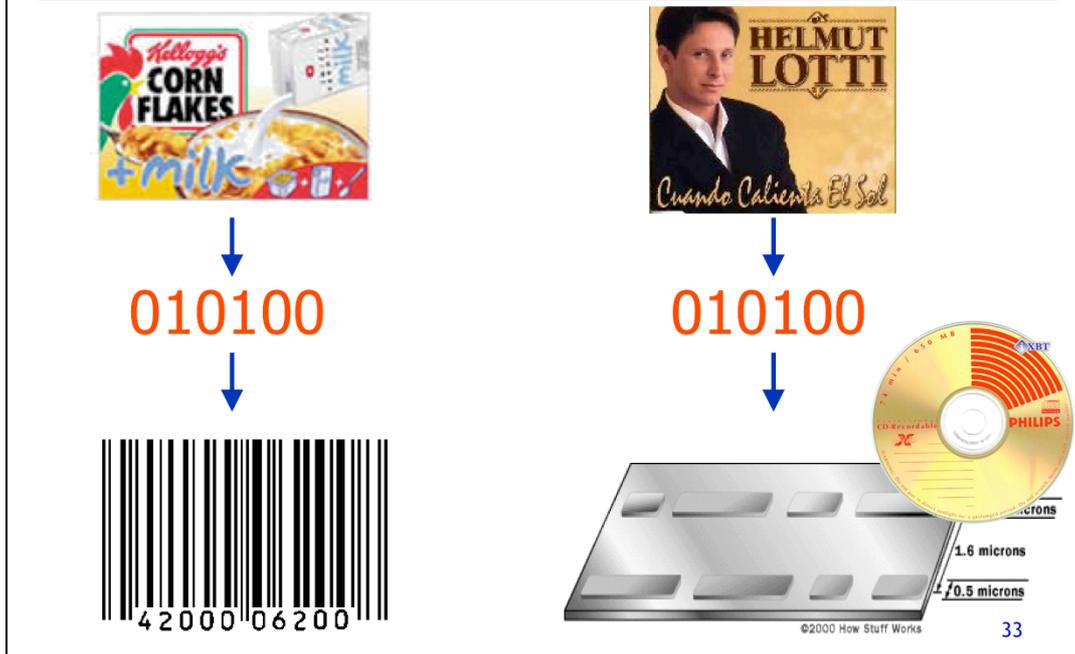
L'image ci-dessus montre les connecteurs présents à l'arrière d'un PC.

Il existe effectivement:

- un connecteur (comportant de nombreux conducteurs) pour la transmission parallèle
- un ou plusieurs connecteurs, dont les ports "USB", pour la transmission série

N.B.: USB signifie: Universal **S**erial Bus

## Le support d'une information numérique n'est pas forcément électrique



Pour terminer: une petite remarque par rapport à la "deuxième étape" (celle consistant à traduire les bits sous forme de signal électrique):

### **Le support d'une information numérique n'est pas forcément électrique**

Voici deux exemples dans lesquelles l'information est effectivement traduite sous forme numérique (étape 1) mais "implémentée" (=réalisée concrètement) dans un autre support qu'un signal électrique:

- dans un code-barres, l'information numérique est implémentée au moyen de la couleur (blanc ou noir)
- sur un CD ou un DVD, l'information numérique est implémentée au moyen de zones plus ou moins réfléchissantes qui sont lues par un laser

Ces deux exemples sont des "supports numériques". L'information qu'ils portent est très facile à transformer en signal électrique.