



0001 1011

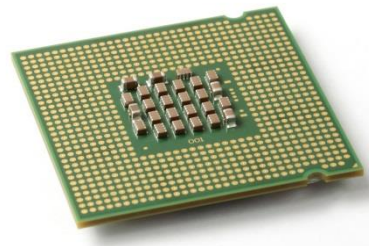
INTRODUCTION

Structure de base de l'ordinateur

- Rôle de l'ordinateur
- Éléments de l'ordinateur



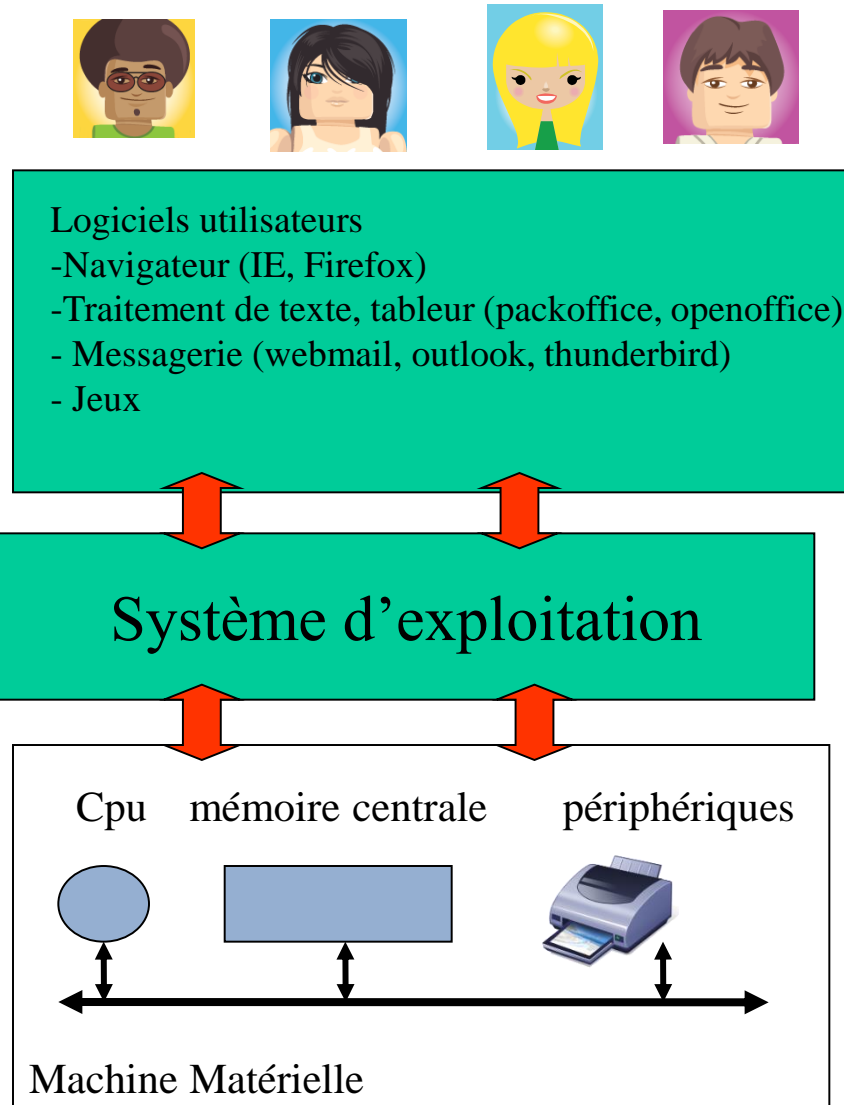
0001 1011



Les différents niveaux de la machine informatique

- On distingue généralement trois couches dans la composition d'une machine informatique :

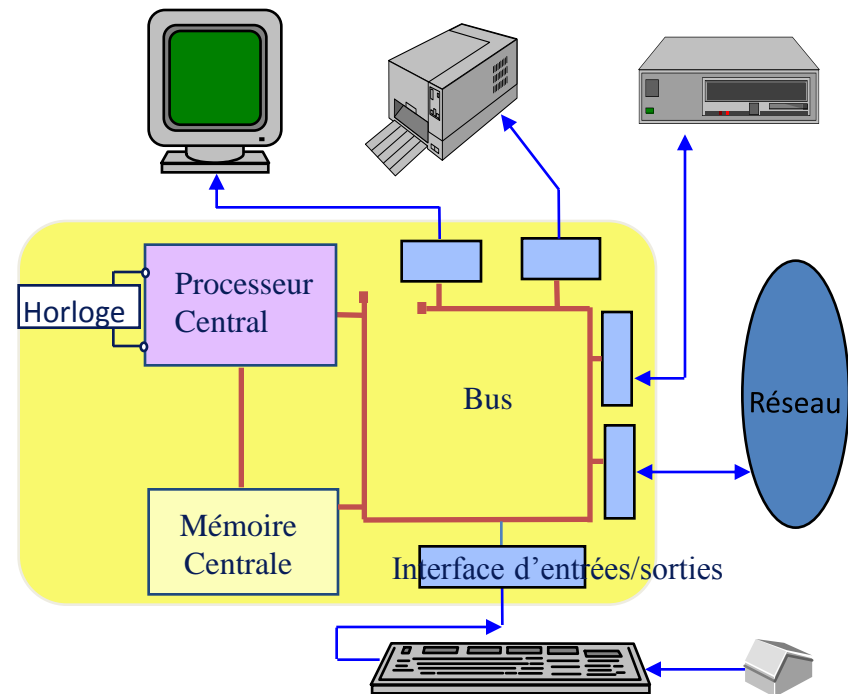
- Les **logiciels des utilisateurs « software »** : ce sont des programmes qui permettent à l'utilisateur de réaliser des tâches sur la machine.
- Le **logiciel de système d'exploitation** : c'est un ensemble de programmes qui se place à l'interface entre le matériel et les logiciels applicatifs. Il permet notamment à ces logiciels applicatifs d'utiliser les ressources matérielles de la machine. Les principaux OS (Operating System) sont notamment Linux, Windows, Mac OS, Unix
- Le **matériel « hardware »** : il correspond à la machine physique, notamment composée du processeur, de la mémoire centrale et des périphériques, l'ensemble communiquant par un bus.



Les fonctions de l'ordinateur

Les composants

- Les fonctions de l'ordinateur sont de permettre à des utilisateurs (via des logiciels informatiques) de :
 - Effectuer du calcul;
 - Stocker des données;
 - Communiquer.
- Pour cela, l'ordinateur est doté d'un ensemble de composants physiques
 - Un élément permettant d'exécuter les instructions d'un programme : c'est le **processeur** (CPU).
 - Des éléments permettant de stocker les données : ce sont les **mémoires** de l'ordinateur.
 - Des éléments permettant la communication entre l'ordinateur et l'être humain : ce sont les **périphériques**.
 - Des éléments permettant aux différents composants (périphériques, processeur, mémoire) de l'ordinateur de communiquer : ce sont les **bus** de l'ordinateur

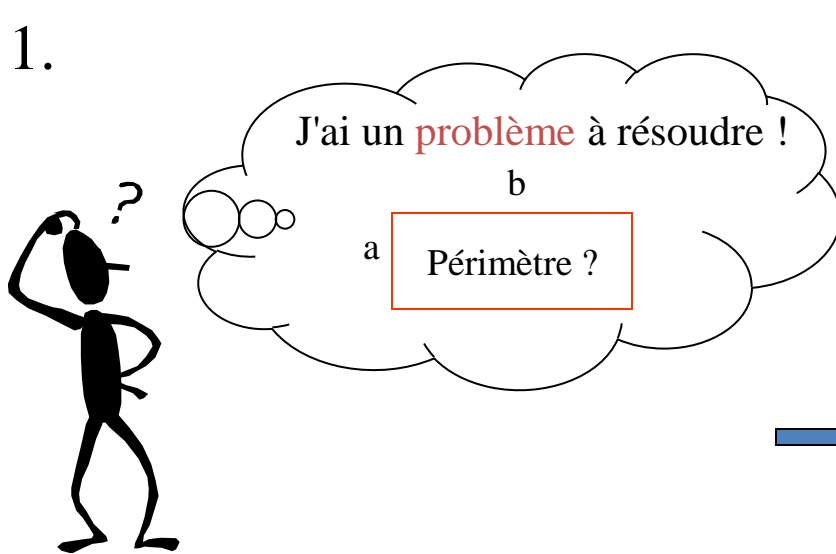


Rôle d'un ordinateur

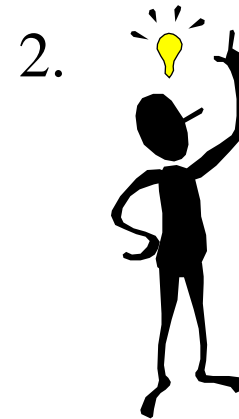
- Exécuter un programme de traitement sur des données en vue de résoudre un problème.
- Deux aspects :
 - L'expression du problème à résoudre, de sa solution dans un **langage compréhensible par l'ordinateur**
 - La **structure de l'ordinateur** qui doit être composé d'éléments permettant le stockage, le traitement, la lecture ou l'écriture des données

A quoi sert donc un ordinateur ?

1.



2.



J'écris une **solution** !
→ **ALGORITHME**

Périmètre := $2a + 2b$

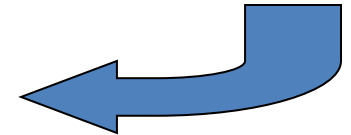
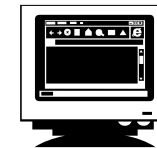


3.

En utilisant un **langage de programmation**, je code la solution pour la faire exécuter par l'ordinateur

→ **PROGRAMME** constitué d'instructions

```
fonction perimetre (a, b : in integer) return  
integer is  
begin  
    perimetre := (2 * a) + (2 * b);  
end;
```



Le codage d'un problème ...

```
fonction perimetre (a, b : in integer) return  
integer is  
begin  
    perimetre := (2 * a) + (2 * b);  
end;
```

Programme en langage de haut niveau
instructions de haut niveau



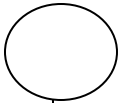
Compilateur

Niveau utilisateur

Système d'exploitation

Gérer et partager le matériel

processeur



Bus

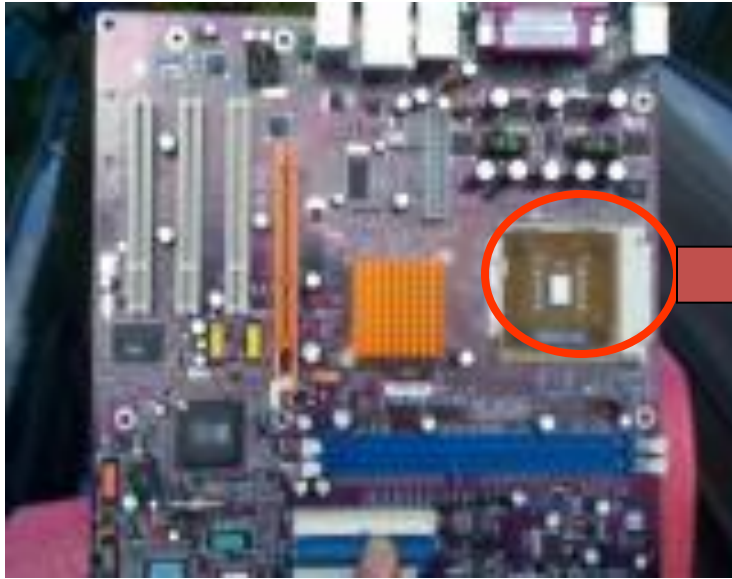
```
01101110111110011  
01111010001011100  
10111101110111111  
00111011110111011  
00111111000111101
```

Mémoire centrale

Machine physique "matérielle"

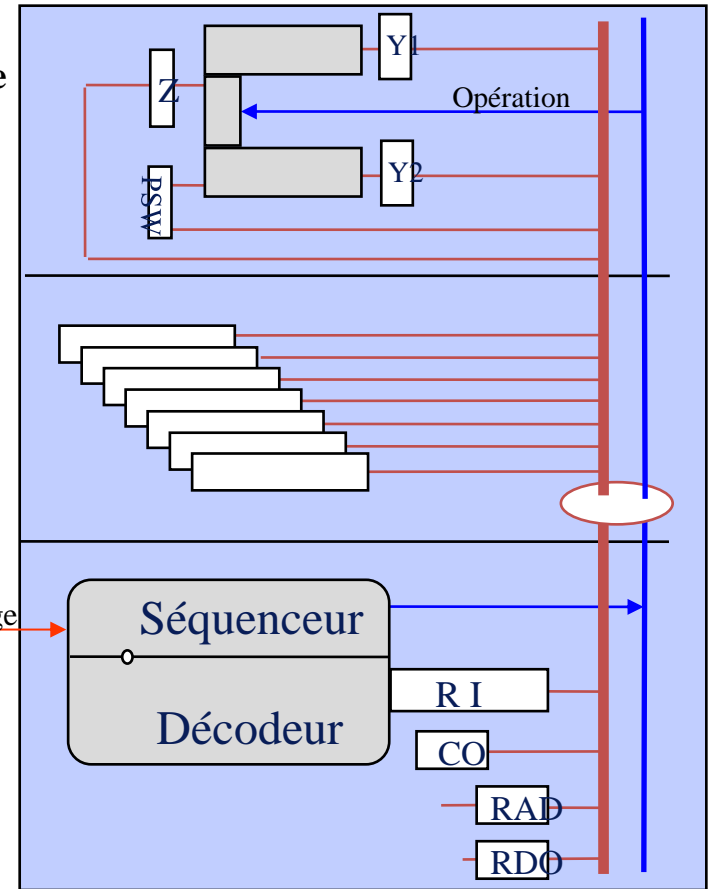
Programme à
exécuter : instructions machine
et valeurs **en binaire**

Composants de l'ordinateur

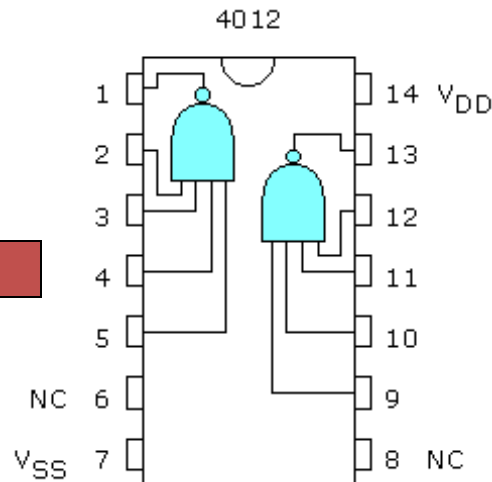
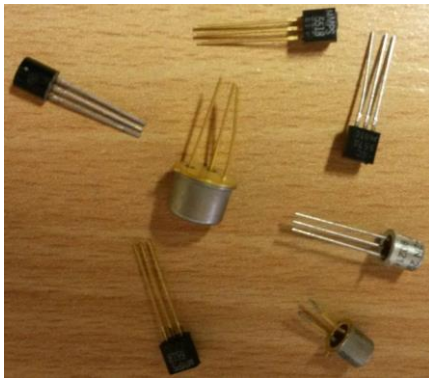


Unité Arithmétique
et logique

Registres



TRANSISTORS



CIRCUITS LOGIQUES OU INTEGRES (PORTES)

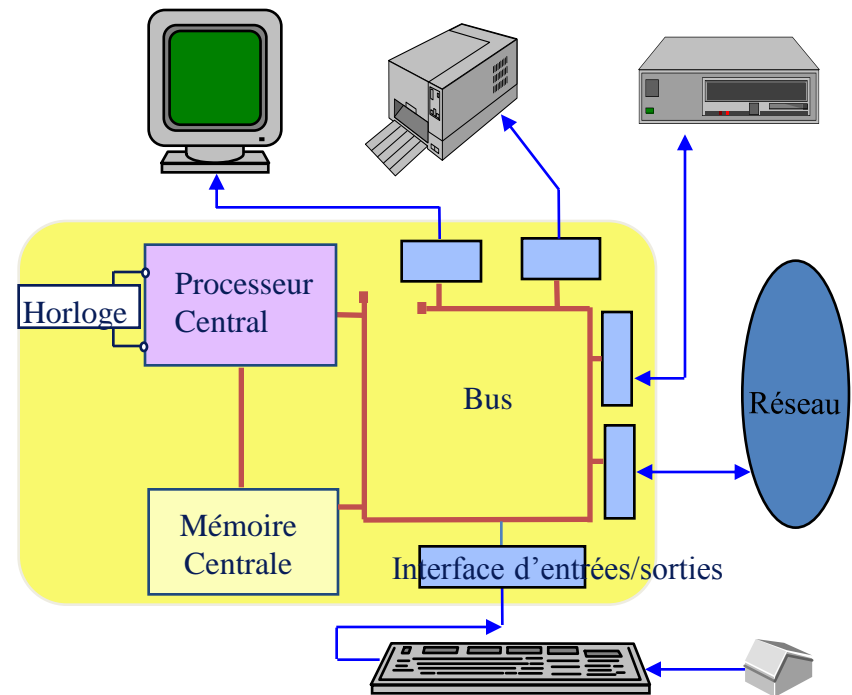
La représentation des informations sur la machine physique

- La donnée de base manipulée par la machine physique est le **bit** (*Binary Digit*) qui ne peut prendre que deux valeurs : 0 et 1
- Ce 0 et 1 correspondent aux deux niveaux de voltage (0-1 et 2-5 volts) admis pour les signaux électriques issus des composants électroniques (transistors) qui constituent les circuits physiques de la machine
- Toutes les informations (nombres, caractères et instructions) ne peuvent être représentées que par une combinaison de 0 et 1 : **chaîne binaire**. Un **octet** est une chaîne de 8 bits.

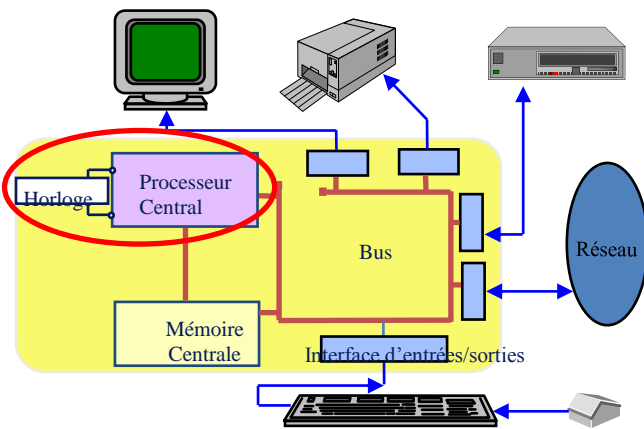
1100 1011

Les composants de l'ordinateur

- L'ordinateur est doté d'un ensemble de composants physiques
 - Des éléments permettant la communication entre l'ordinateur et l'être humain : ce sont les **périphériques**.
 - Un élément permettant d'exécuter les instructions d'un programme : c'est le **processeur** (CPU).
 - Des éléments permettant de stocker les données : ce sont les **mémoires** de l'ordinateur.
 - Des éléments permettant aux différents composants (périphériques, processeur, mémoire) de l'ordinateur de communiquer : ce sont les **bus** de l'ordinateur



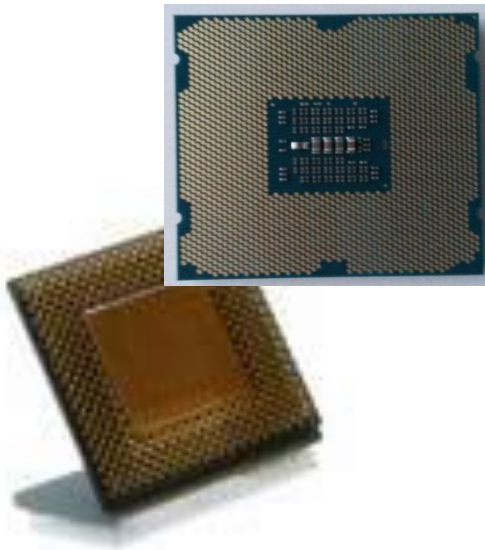
Le processeur



Le **processeur** (**CPU**, pour *Central Processing Unit*) est le cerveau de l'ordinateur. Il permet de manipuler, des données et des instructions codées sous forme binaires.

Le **processeur** est un circuit électronique cadencé au rythme d'une horloge interne qui envoie des impulsions, appelées « **top** ». La **fréquence d'horloge**, correspond nombre d'impulsions par seconde. Elle s'exprime en Hertz (Hz).

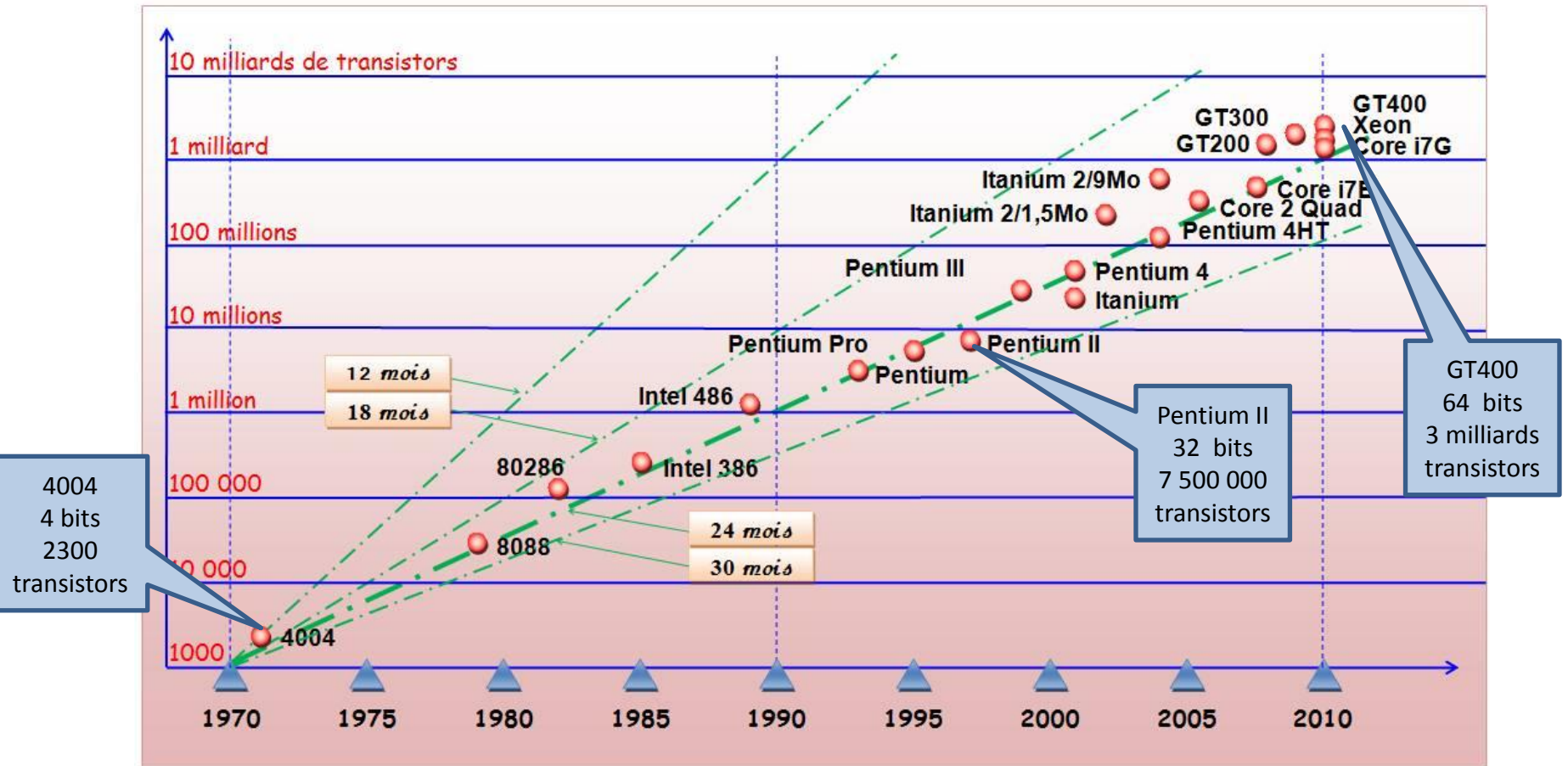
- Ordinateur à 2 GHz → l'horloge envoie 200 000 000 000 battements par seconde.



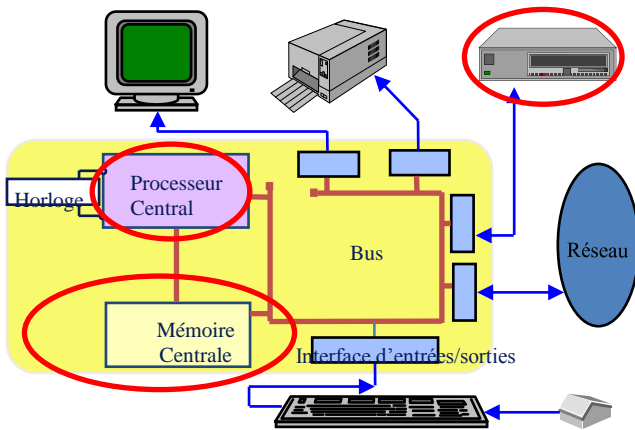
Circuits électroniques composés de millions de transistors placés dans un boîtier comportant des connecteurs d'entrée-sortie, surmonté d'un ventilateur.

→ **circuit intégré ou puce**

Le processeur : performances



La **Loi de Moore** a été exprimée en 1965 par *Gordon Moore*, un des trois fondateurs d'Intel. Elle postule que le nombre de transistors sur une puce double tous les deux ans.



Les mémoires de l'ordinateur

Une « **mémoire** » est un composant électronique capable de stocker temporairement des informations

- Une mémoire est caractérisée par :
 - Sa **capacité**, représentant le volume global d'informations (en bits) que la mémoire peut stocker (par exemple 1 Goctets, soit 2^{30} octets, soit $2^{30} * 8$ bits).
 - Son **temps d'accès**, correspondant à l'intervalle de temps entre la demande de lecture/écriture et la disponibilité de la donnée.
- L'ordinateur contient différents niveaux de mémoire, organisés selon une **hiérarchie mémoire**.

Les grandeurs de l'ordinateur

Capacité – bit - octet

1 octet = 8 bits (byte)	Avant 1998	Après 1998
Kilooctet (Ko)	2^{10} octets = 1024 octets	1000 octets
Mégaoctet (Mo)	2^{20} octets = 1024 Koctets	1000 Koctets
Gigaoctet (Go)	2^{30} octets = 1024 Mcoctets	1000 Mcoctets

Multiples de l'octet :
préfixes SI et mésusages

Nom	Symbole	Valeur	Mésusage ²
kilooctet	ko	10^3	2^{10}
mégaoctet	Mo	10^6	2^{20}
gigaoctet	Go	10^9	2^{30}
téraoctet	To	10^{12}	2^{40}
pétaoctet	Po	10^{15}	
exaoctet	Eo	10^{18}	
zettaoctet	Zo	10^{21}	
yottaoctet	Yo	10^{24}	

Multiples de l'octet :
préfixes binaires

Nom	Symbole	Valeur
kibioctet	kio	2^{10}
mébioctet	Mio	2^{20}
gibioctet	Gio	2^{30}
tébioctet	Tio	2^{40}
pébioctet	Pio	2^{50}
exbioctet	Eio	2^{60}
zébioctet	Zio	2^{70}
yobioctet	Yio	2^{80}

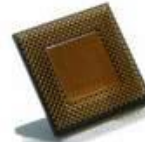
DUT Informatique, Introduction aux

systèmes informatiques

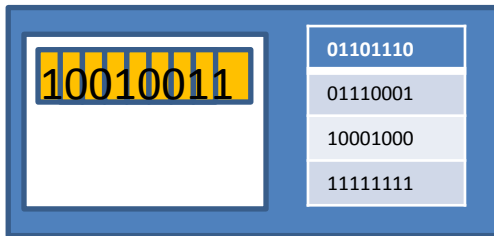
Les mémoires de l'ordinateur

- L'ordinateur contient différents niveaux de mémoire, organisés selon une **hiérarchie mémoire**.

Mémoires vives : mémoires **volatiles** :



Mémoires de masse :
mémoires **permanentes**



REGISTRES
N bits (32, 64)
1 nanoseconde

Mémoires Caches
Koctets (L1,L2)
5 nanosecondes

Mémoires Centrales
Goctets
10 nanosecondes

Mémoires de masse
500 Goctets - Toctets
5 millisecondes

Au plus près du cpu

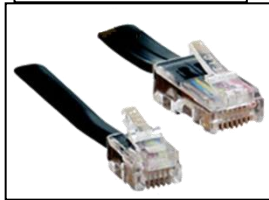
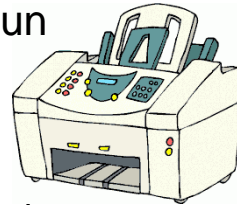
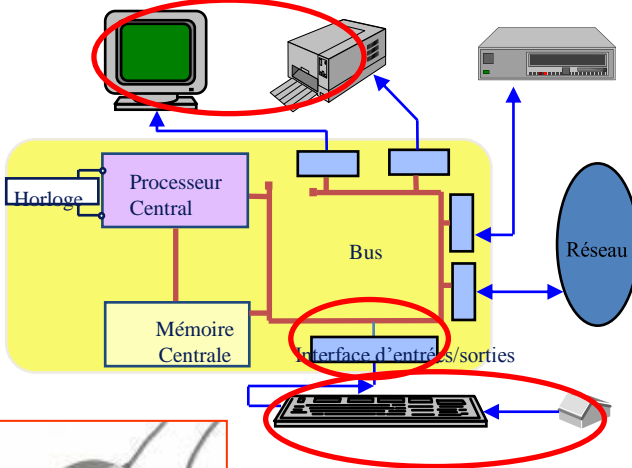
Capacité, vitesse

Au plus loin du cpu

Mémoire volatile : le contenu de la mémoire n'existe que si il y a une alimentation électrique (typiquement les mémoires caches et mémoire centrale)
Mémoire permanente, de masse : mémoire de grande capacité dont le contenu demeure même sans alimentation électrique (typiquement le disque dur)

Périphériques de l'ordinateur

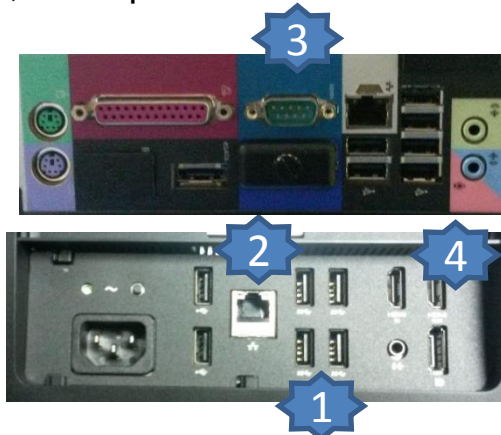
Un périphérique est un matériel électronique pouvant être raccordé à un ordinateur par l'intermédiaire de l'une de ses **interfaces d'entrée-sortie** (interface VGA, HDMI, USB, RJ45.), le plus souvent par l'intermédiaire d'un **connecteur**. L'interface d'entrées-sorties est pilotée par un **driver (pilote d'entrées-sorties)**

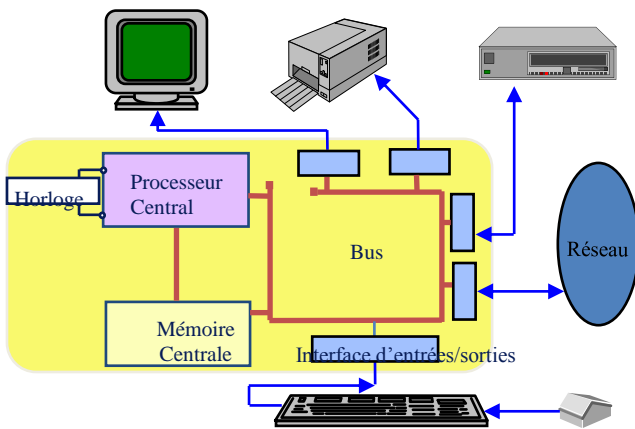


- On distingue habituellement les catégories de périphériques suivantes :
 - **périphériques de sortie**: ce sont des périphériques permettant à l'ordinateur d'émettre des informations vers l'extérieur, tels qu'un écran, une imprimante..
 - **périphériques d'entrée** : ce sont des périphériques capables uniquement d'envoyer des informations à l'ordinateur, par exemple la souris, le clavier, etc.
 - **périphériques d'entrée-sortie** : ce sont des périphériques capables d'envoyer des informations à l'ordinateur et permettant également à l'ordinateur d'émettre des informations vers l'extérieur, par exemple le modem, le disque dur

• Interfaces

1. USB : connexion « à chaud » de périphériques
2. RJ45 : connexion au réseau local filaire
3. VGA : connexion de l'écran
4. HDMI : connexion à un écran haute résolution





Les bus de l'ordinateur



Un « **bus** » est un composant électronique permettant à différents composants de l'ordinateur de s'échanger des informations

- Bus système (*Front Side Bus FSB*) permet la communication entre le processeur et la mémoire centrale.
- Bus d'extension permet aux autres éléments de l'ordinateur de communiquer entre eux.
- Bus série, bus parallèle, largeur de bus



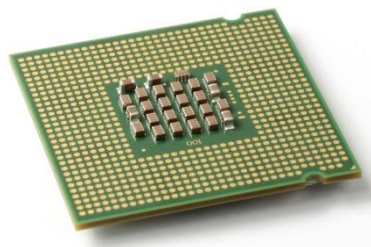


0001 1011

Le langage de l'ordinateur

Représentation des informations

0001 1011



La représentation des informations sur la machine physique

1 bit	0, 1	2^1 états
2 bits	00, 01, 10, 11	2^2 états
3 bits	000, 001, 010, 011, 100, 101, 110, 111	2^3 états
n bits		2^n états

1 octet = 8 bits (byte)	Avant 1998	Après 1998
Kilooctet (Ko)	2^{10} octets = 1024 octets	1000 octets
Mégaoctet (Mo)	2^{20} octets = 1024 Koctets	1000 Koctets
Gigaoctet (Go)	2^{30} octets = 1024 Mcoctets	1000 Mcoctets

La représentation des informations sur la machine physique

$(N)_X = a_n \dots a_1 a_0$ exprimé dans la base $X = a_0 \times X^0 + a_1 \times X^1 + \dots + a_n \times X^n$

Poids fort

Poids faible

base	symboles	valeur
$X = 2$	0, 1	$110 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2$
$X = 10$	0,1,2,3,4,5,6,7,8,9	$110 = 0 \times 10^0 + 1 \times 10^1 + 1 \times 10^2$
$X = 16$	0,1,2,3,4,5,6,7,8,9 A B C D E F	$110 = 0 \times 16^0 + 1 \times 16^1 + 1 \times 16^2$

La représentation des informations sur la machine physique

base	chiffres	Écriture en base 2
10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001
16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9 A (10), B (11), C(12), D(13), E(14), F(15)	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111
8	0, 1, 2, 3, 4, 5, 6, 7	000, 001, 010, 011, 100, 101, 110, 111

$$(AF2)_{16} = (1010\ 1111\ 0010)_2 \quad (72)_8 = (111\ 010)_2$$

La représentation des informations sur la machine physique

Exemple : Convertir $(235,625)_{10}$ en base 2

$$(235,625)_{10} = (11101011, 1010000)_2$$

- Les bases d'expression : conversions utiles

Base 10 \rightarrow base X

- Méthode des divisions :**
- Partie entière : on divise $PE(N)$ par X jusqu'à obtenir un quotient égal à 0

Partie fractionnaire : on multiplie $PF(N)$ par X

$$0,625 * 2 = 1,25$$

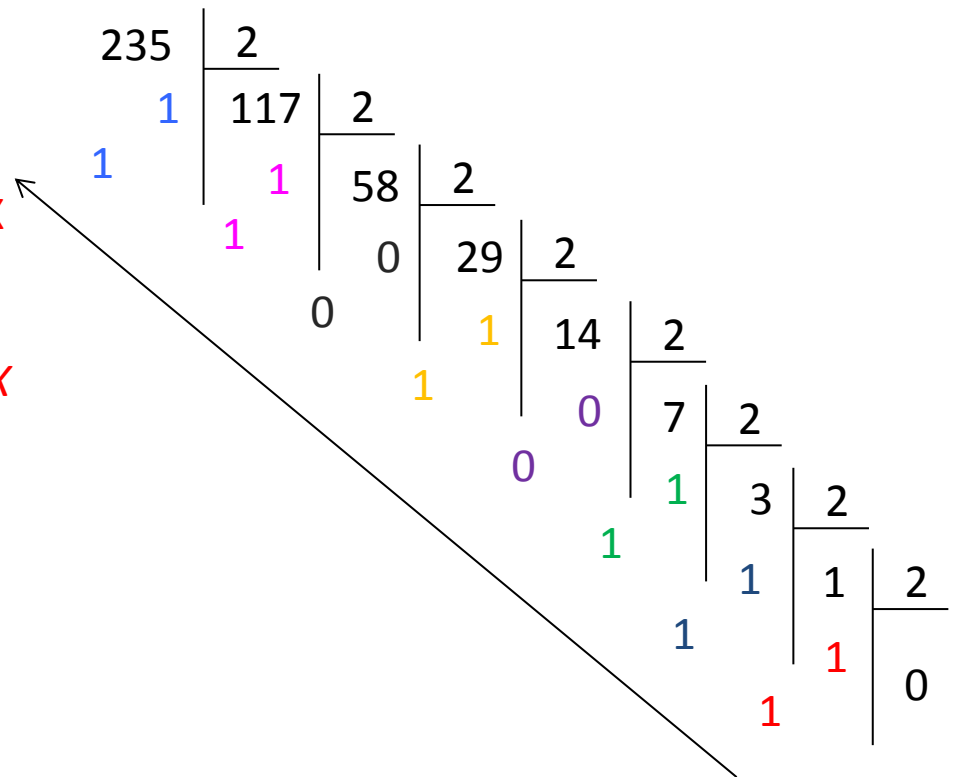
$$0,25 * 2 = 0,5$$

$$0,5 * 2 = 1,0$$

$$0 * 2 = 0,0$$

.....

$$(0,625)_{10} = (0, 1010000)_2$$



La représentation des informations sur la machine physique

- Les bases d'expression : conversions utiles

Base 10 \rightarrow base X

- **Méthode des soustractions** : on soustrait à N la plus grande puissance de X qui lui est inférieure ou égale, puis on répète le processus sur la différence résultante jusqu'à obtenir 0.

Exemple : Convertir $(235,625)_{10}$ en base 2

On sait

$$2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32, 2^6 = 64, 2^7 = 128, 2^8 = 256, 2^9 = 512, 2^{10} = 1024$$

$$2^{-1} = \frac{1}{2} = 0,5, 2^{-2} = \frac{1}{4} = 0,25, 2^{-3} = \frac{1}{8} = 0,125, 2^{-4} = \frac{1}{16} = 0,0625$$

$$235 - 2^7 = 107 ; 107 - 2^6 = 43 ; 43 - 2^5 = 11 ; 11 - 2^3 = 3 ; 3 - 2^1 = 1 ; 1 - 2^0 = 0$$

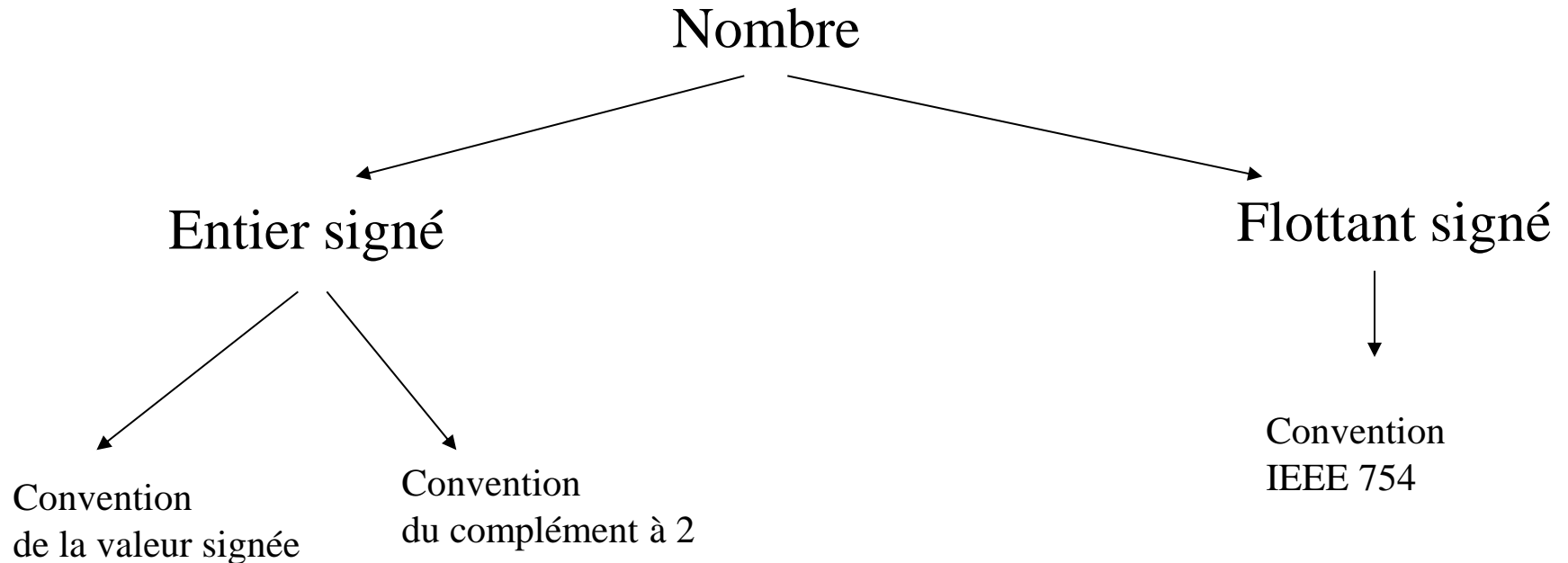
$$235 = 2^7 + 2^6 + 2^5 + 2^3 + 2^1 + 2^0 = (11101011)_2$$

$$0,625 - 2^{-1} = 0,125 ; 0,125 - 2^{-3} = 0$$

$$0,625 = (0,101)_2$$

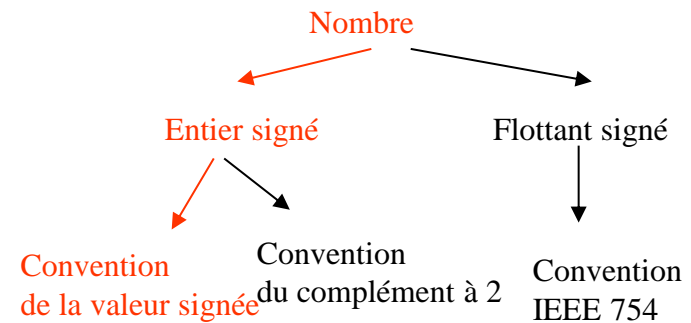
$$(235,625)_{10} = (11101011,101)_2$$

La représentation des informations sur la machine physique



La valeur d'un nombre signé est représentée par une **chaîne binaire de taille fixe** (n bits).
en utilisant une **convention de représentation**

La représentation des informations sur la machine physique



Le bit de poids fort sert de bit de signe et vaut 1 si le nombre est négatif, 0 sinon. Les autres bits codent la valeur absolue du nombre en binaire (base 2)

Exemple : représenter + 124 sur 8 bits en valeur signée

$$124 = 64 + 32 + 16 + 8 + 4 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 01111100$$

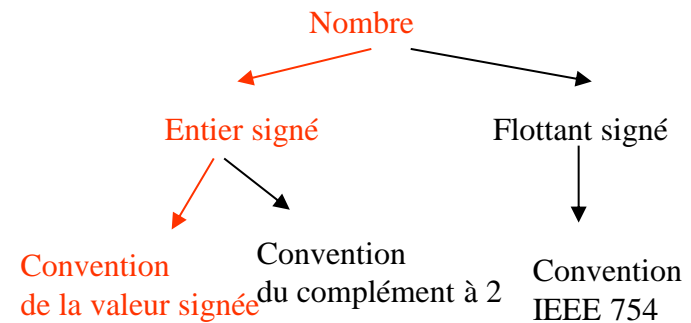
$$+124 = 01111100$$

Exemple : représenter - 124 sur 8 bits en valeur signée

$$124 = 64 + 32 + 16 + 8 + 4 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 01111100$$

$$- 124 = 11111100$$

La représentation des informations sur la machine physique

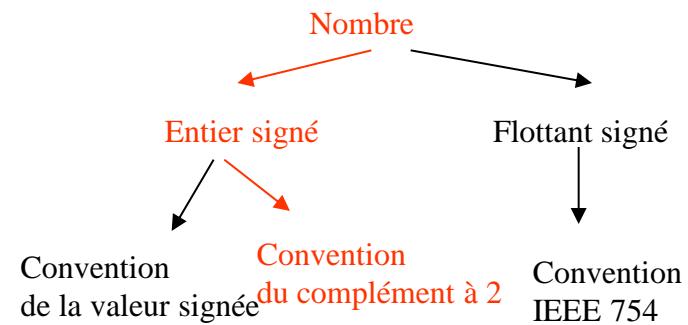


Toute chaîne binaire commençant par un 1 est un nombre négatif : sur 8 bits ce sont les chaînes 11111111 (-127) à 10000000 (- 0)

Toute chaîne binaire commençant par un 0 est un nombre positif : sur 8 bits ce sont les chaînes 01111111 (+127) à 00000000 (+ 0)

! L'arithmétique des ordinateurs est différente de l'arithmétique des humains :
l'ensemble des nombres représentables est contraint.
Ici il y a deux représentations du 0 !

La représentation des informations sur la machine physique



Un nombre positif est représenté par son équivalent binaire sur n bits.
Un nombre négatif est représenté en prenant le complément à 2 de son équivalent positif.

Exemple : représenter + 124 sur 8 bits en complément à 2

$$124 = 64 + 32 + 16 + 8 + 4 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 01111100$$

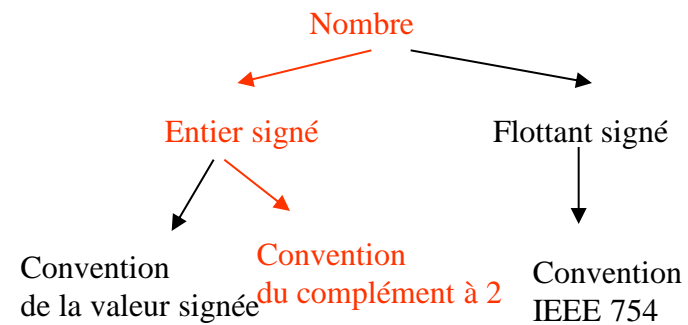
Exemple : représenter - 124 sur 8 bits en complément à 2

$$124 = 64 + 32 + 16 + 8 + 4 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 01111100$$

01111100	
10000011	inversion des bits ajout de 1
+ 1	

10000100 = - 124	

La représentation des informations sur la machine physique



Toute chaîne binaire commençant par un 1 est un nombre négatif : sur 8 bits ce sont les chaînes 11111111 (-1) à 10000000 (- 128)

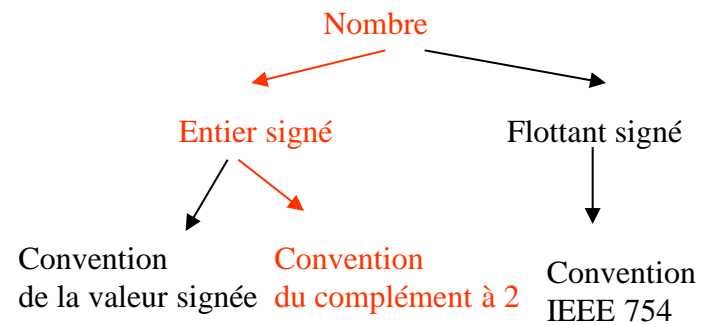
Toute chaîne binaire commençant par un 0 est un nombre positif : sur 8 bits ce sont les chaînes 01111111 (+127) à 00000000 (+ 0)

$$\begin{array}{r} 11111111 \\ 00000000 \\ + \quad \quad 1 \\ \hline 00000001 \end{array}$$

$$\begin{array}{r} 11111111 \\ - \quad \quad 1 \\ \hline 11111110 \\ 00000001 \end{array}$$

$$\begin{array}{r} 10000000 \\ 01111111 \\ + \quad \quad 1 \\ \hline 10000000 \end{array}$$

La représentation des informations sur la machine physique



Exemple : additionner + 127 et + 2 avec une représentation en complément à 2 sur 8 bits

127	01111111
+ 2	+ 00000010

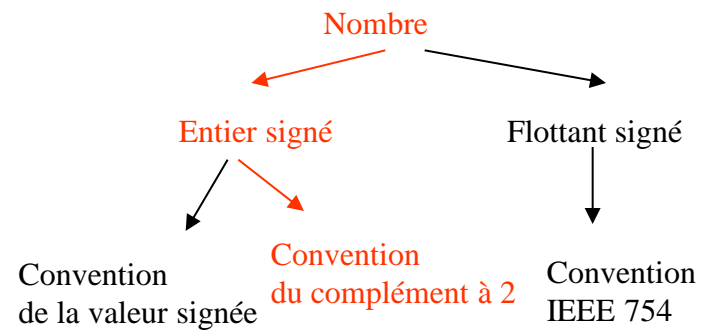
129	10000001
-----	----------

→ ce résultat est un nombre négatif (-127) et non la valeur 129
le résultat est trop grand pour la machine : il y a **OVERFLOW**

Le résultat est faux !

Éventail des nombres représentables [- 128, 127]

La représentation des informations sur la machine physique



Exemple : additionner + 127 et - 2 avec une représentation en complément à 2 sur 8 bits

127	01111111
- 2	+ 11111110

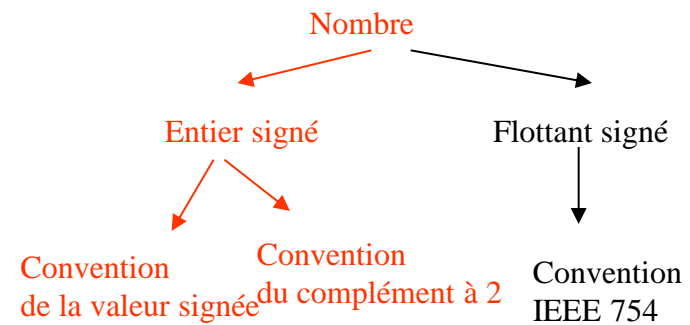
125	101111101

---> le résultat ne tient pas sur 8 bits : il y a **CARRY**

Le résultat est sur 9 bits !

Éventail des nombres représentables [- 128, 127]

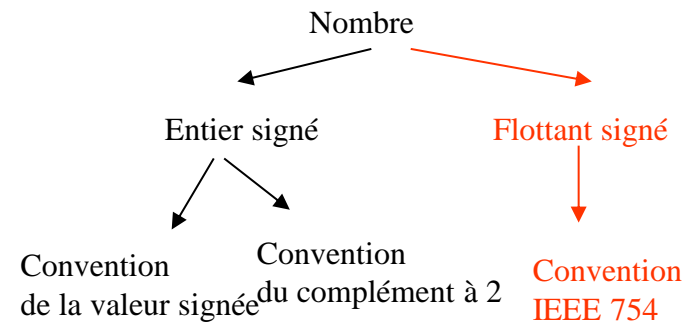
La représentation des informations sur la machine physique



- Nombres entiers signés sur n bits

	8 bits	p bits
Valeur signée	[-127, + 127]	$[- 2^{p-1} - 1, + 2^{p-1} - 1]$
Complément à 2	[-128, + 127]	$[- 2^{p-1}, + 2^{p-1} - 1]$

La représentation des informations sur la machine physique



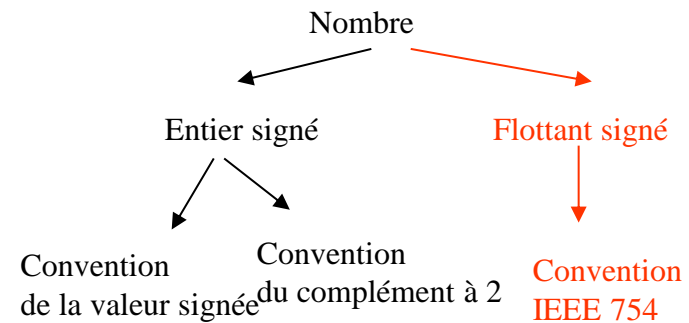
- Représentation en virgule flottante
- Un nombre est représenté en virgule flottante dans la base B s'il est sous la forme

$$\pm M1, M2 \times B^{\pm c}$$

M1, M2 est la **mantisse**; B est la base; c est l'exposant.

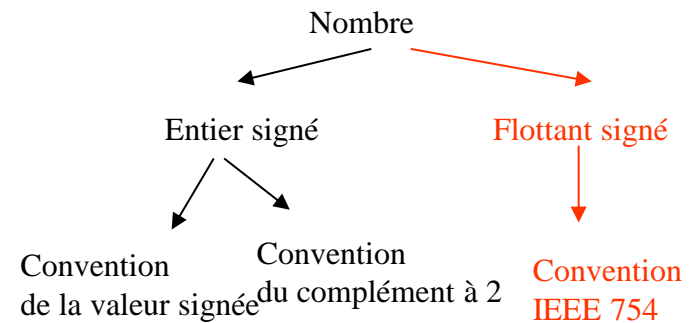
- *Exemple* : $+59,4151 * 10^{-5}$
- Il s'agit de représenter la mantisse et son signe, ainsi que l'exposant et son signe.

La représentation des informations sur la machine physique

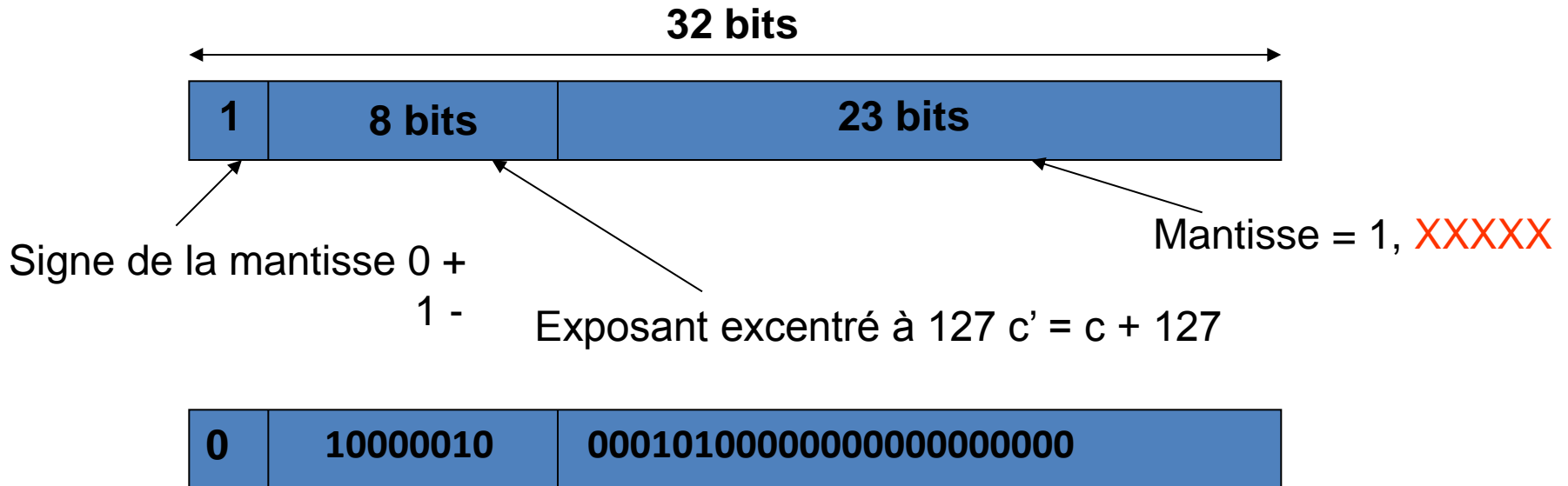


- Représentation IEEE 754 : $\pm 1,XXXX \times 2^c$
 - Deux formats de représentation : simple précision sur 32 bits; double précision sur 64 bits
 - La mantisse est normalisée sous la forme 1,XXXX (pseudo-mantisse) et le 1 n'est pas codé en machine (bit caché). Le signe est codé sur un bit valant 0 si la mantisse est positive, et 1 si elle est négative.
 - L'exposant est codé avec un excédant à 127 (SP) et 1023 (DP) de manière à coder une valeur positive. L'exposant c' codé est alors égal à $c + 127$ ou 1023.

La représentation des informations sur la machine physique



• Représentation IEEE 754 Simple Précision :



Représenter $(8,625)_{10}$

$$(8,625)_{10} = (1000 + 0,101) = 1,000101 * 2^3$$

$$\text{exposant } c' = 3 + 127 = 130 = (10000010)_2$$

Représenter $(8,625)_{10}$

$$(8)_{10} = (1000)_2$$

$$(0,625)_{10} = (0,101)_2$$

La représentation des caractères sur la machine physique

- La valeur d'un caractère est représentée par une **chaîne binaire de taille fixe**.
- La valeur binaire de chaque caractère est stockée dans une table.
- Il existe trois grands codages pour les caractères :
 - le code ASCII (*American Standard Code for Information Interchange*) sur 7 bits
 - le code EBCDIC (*Extended Binary-Coded Decimal Interchange Code*), **développé par IBM** sur 8 bits
 - le code UNICODE **sur 16 bits mis au point en 1991. Il est compatible avec le code ASCII.**

La représentation des caractères sur la machine physique

	0	1	2	3	4	5	6	7	Poids fort
0	NUL	DLE	sp.	0		P		p	0
1	SOH	DC1	!	1	A	Q	a	q	1
2	STX	DC2		2	B	R	b	r	2
3	ETX	DC3		3	C	S	c	s	3
4	EOT	DC4		4	D	T	d	t	4
5	ENQ	NAK	%	5	E	U	e	u	5
6	ACK	SYN	&	6	F	V	f	v	6
7	BEL	ETB		7	G	W	g	w	7
8	BS	CAN	(8	H	X	h	x	8
9	HT	EM)	9	I	Y	i	y	9
A	LF	SUB	*	:	J	Z	j	z	A
B	VT	ESC	+	;	K		k		B
C	FF	FS	,	<	L		l		C
D	CR	GS	-	=	M		m		D
E	SO	RS	.	>	N		n		E
F	SI	US	/	?	O	-	o	DEL	F
Poids faible	0	1	2	3	4	5	6	7	

■ caractère non défini dans la norme (choix nationaux)

G est codé 47 hexa 0100 0111

Figure II-6 – Table des codes A.S.C.I.I.

- Code ASCII
- Les codes 0 à 31 sont *caractères de contrôle*. Ils permettent de faire des actions telles que :
 - retour à la ligne (CR)
 - Bip sonore (BEL)
- Les codes 65 à 90 représentent les majuscules
- Les codes 97 à 122 représentent les minuscules