

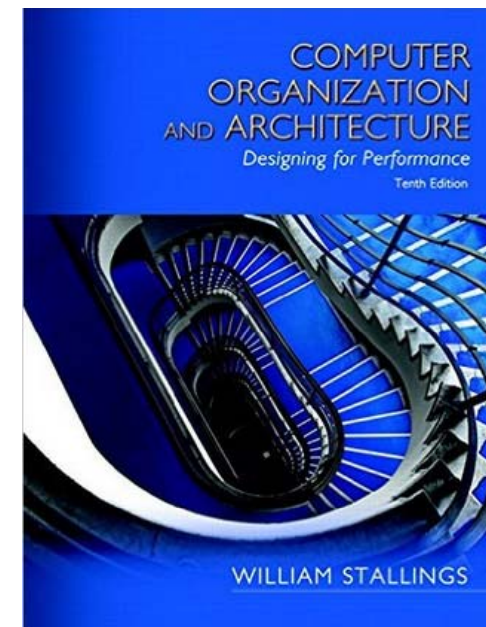
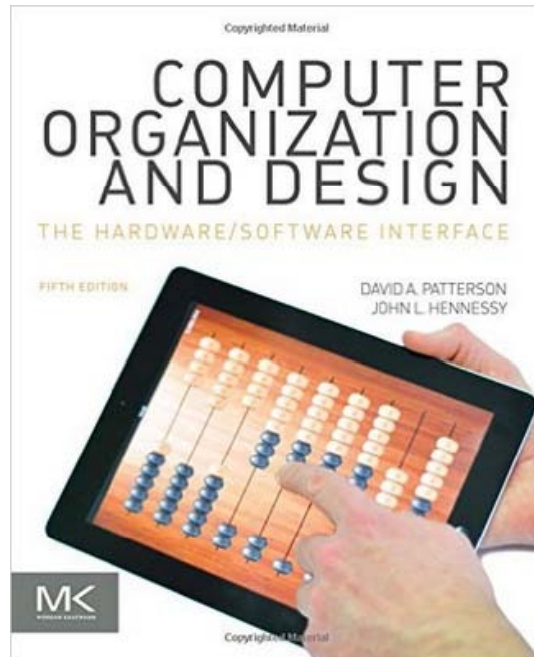
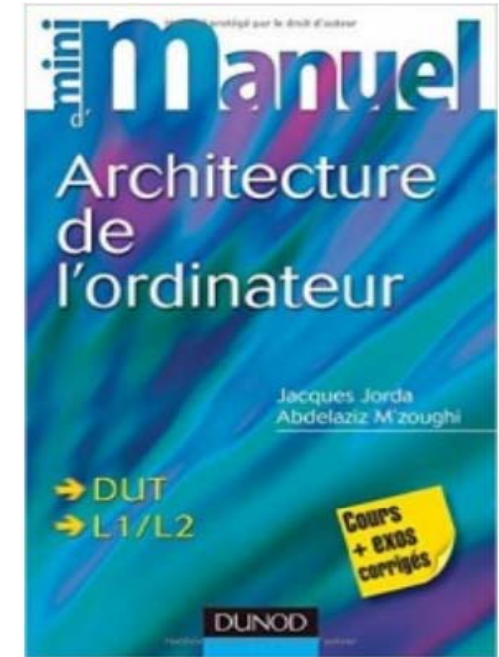
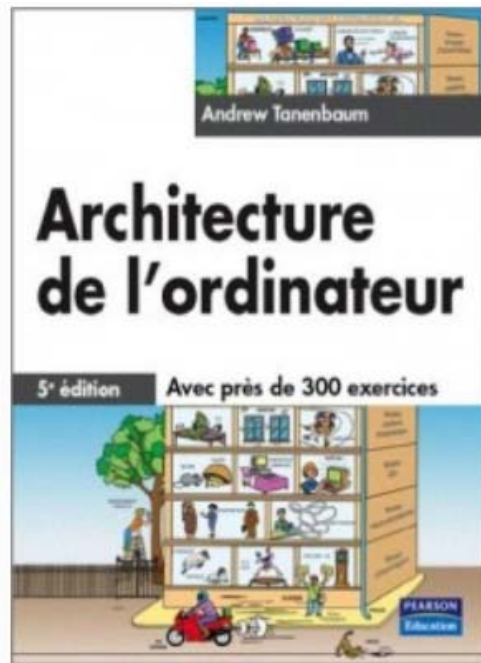
Programme "Architecture des ordinateurs UN".

1. Éléments de base (*Numération, Algèbre de Boole*).
2. Logique Combinatoire et Séquentielle (*Circuits Combinatoires et Séquentiels, Circuits Mémoire*).
3. Principaux éléments d'un ordinateur.
4. Étude d'une machine pédagogique.

Les objectifs du cours, selon le programme officiel.

- ✓ Vulgariser l'ensemble des concepts de base d'un ordinateur ;
- ✓ Connaître la représentation des nombres ;
- ✓ Connaître les principaux composants d'un ordinateur ;
- ✓ Connaître les méthodes de synthèse de systèmes logiques combinatoires et séquentiels ;
- ✓ Acquérir une connaissance de bas niveau de la programmation.

Bibliographie du cours (*Texbooks Requis et recommandés*)



Introduction à la machine à information.

Une machine à information (où l'ordinateur n'est qu'un cas particulier, mais certainement le plus impressionnant) est formée de:

- ✓ **Une partie matérielle.**
- ✓ **Une partie logicielle.**
- ✓ **Une interface utilisateur.**
- ✓ **Et du "bug".**

Autres que l'ordinateur, on peut trouver: le Smartphone, la Tablette, la console de jeux, le démodulateur numérique, les équipements réseaux, les machines à information embarquées à bord d'équipements divers, etc.

La machine à information:

Caractéristiques.

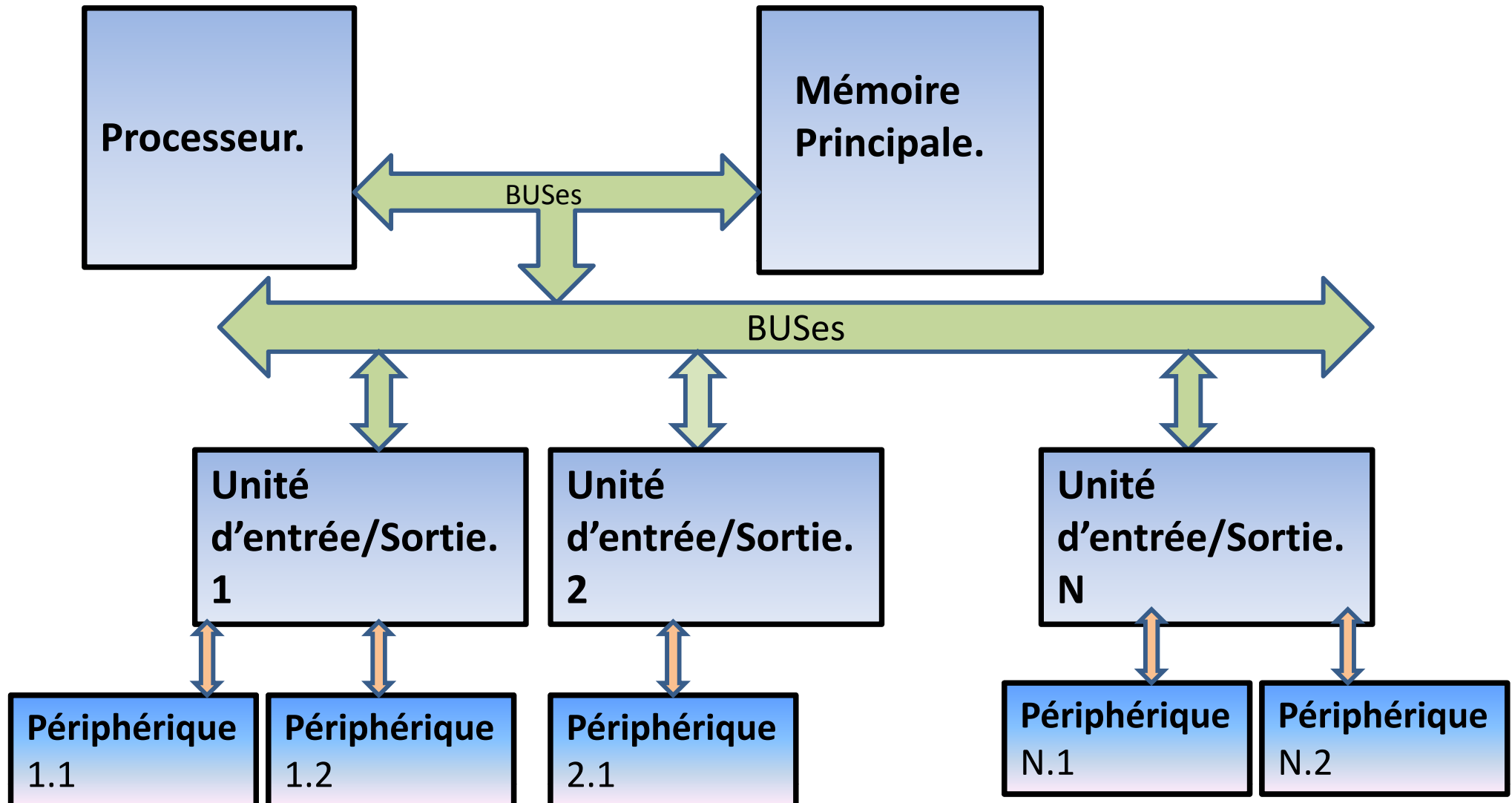
- ✓ **Trois grandes fonctionnalités:**
Traitement de données, Stockage de données, et Communication avec ses paires (en réseau).
- ✓ **Son caractère programmable.**(si c'est un ordinateur alors cela en fait une machine universelle, à multiples usages)
- ✓ **Son caractère extensible** (ajout de matériel, ajout de logiciel.)
- ✓ **Son caractère nomade et ubiquitaire.**
- ✓ **Capacité et vitesse toujours en croissance.**

Les principes de la machine à information selon Von Neumann.

Composants classiques d'un ordinateur:

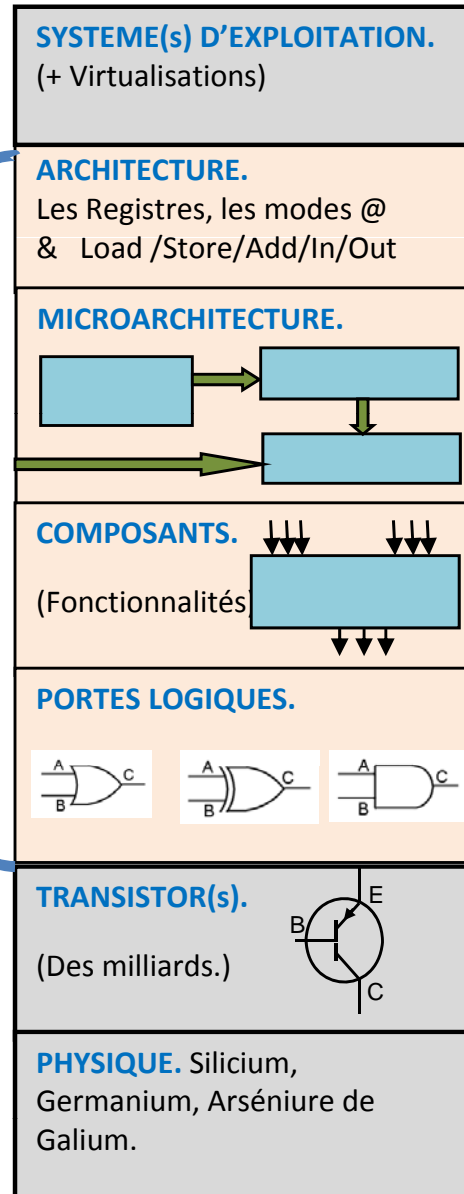
- ✓ **La mémoire principale ou centrale (main memory)** qui héberge les programmes (données et instructions) à exécuter.
- ✓ **Le processeur (CPU ou processor)** qui exécute les programmes chargés en mémoire.
- ✓ **Les unités d'entrée/sortie (input/output units)** qui permettent le lien et l'échange des données ...
- ✓ **avec les périphériques:**
 - de communication Homme-Machine,
 - de stockage de masse et persistant,
 - et de communication Machine-Machine.

Composants classiques d'une Machine à information.



La partie matérielle, objet du cours.

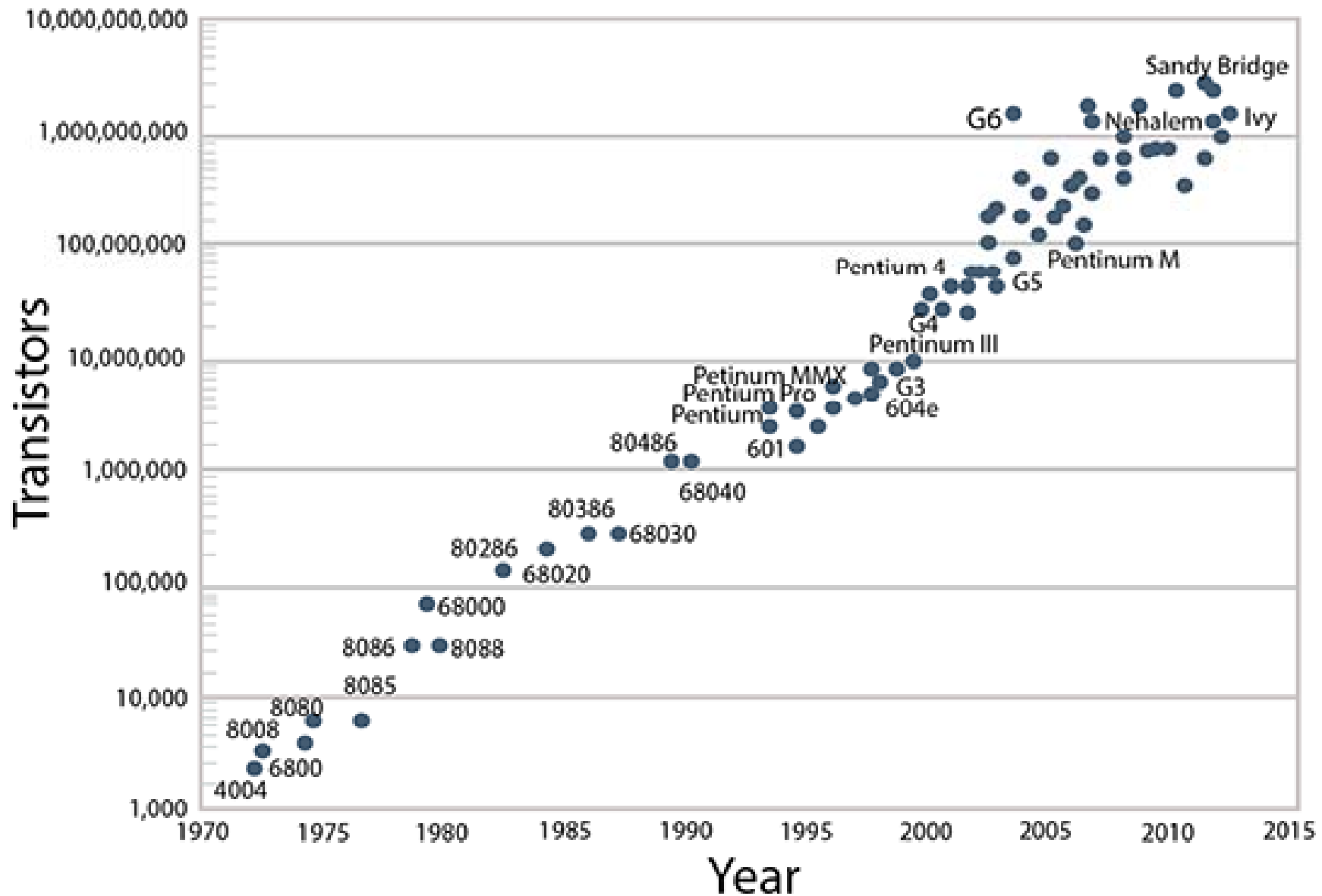
ARCH1
ARCHITECTURE 1 :



début de la
couche Logicielle

Couche
Matérielle

Moore's Law.



Source: University of Wisconsin-Madison

Représentation des données.

✓ Question simple:

Comment sont représentées les données que cette machine à information traite, stocke, envoie et reçoit (dans sa fonction communication)?

✓ Réponse :

Ces données sont toutes formées (codées) à partir d'une donnée atomique appelée "binary digit"

ou bit ou digit binaire à deux états seulement.

Lorsqu'on parle de données, il s'agit de nombres, texte (simple en enrichi), de son, d'image fixe ou animée ou d'une quelconque combinaison de ces types.

Les programmes doivent aussi être représentés.

Toute donnée de notre monde est projetée, dans sa version numérique, sous forme d'une suite de bits selon des règles.

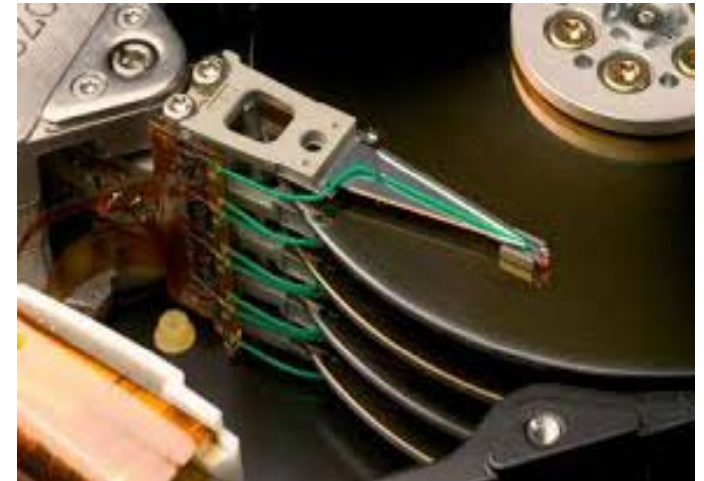
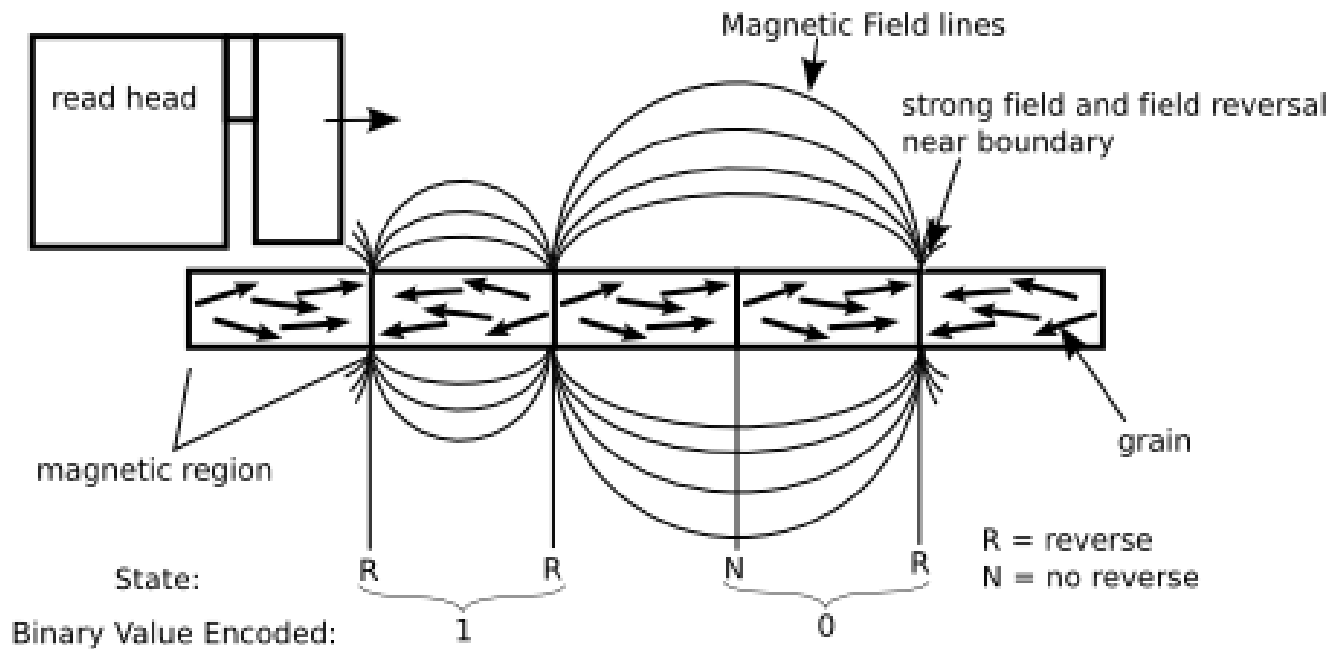
Etat physique/logique du bit.

- ✓ Les deux états du bit sont communément représentés par **1** et **0**, cependant ceci n'est qu'une abstraction de leurs réelles matérialisations.
- ✓ Dans la machine à information, le bit a une concrétisation physique sous forme électrique, magnétique, optique ou déformation géométrique de surface.

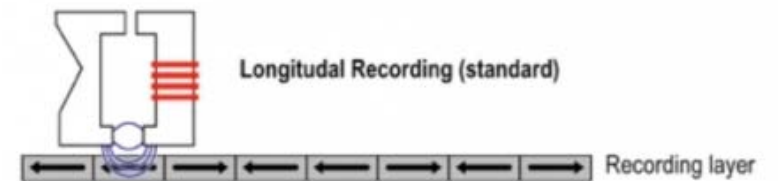
Exemples de concrétisations.

- ✓ **Electrique: deux niveaux de tension électrique:**
exemple
 - 0V à 0.7V Correspond au zéro
 - 4,5V à 5V correspond au un
- ✓ **Magnétique:**
 - deux états de magnétisation et de transitions magnétiques.
- ✓ **Optique:**
 - Deux longueurs d'onde différentes.
- ✓ **Déformations géométrique de surface**
 - Surface trouée ou pas
 - Cuvette ou pas

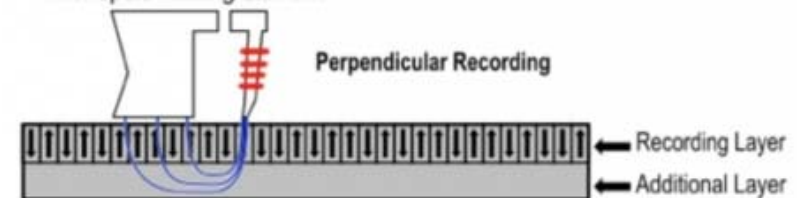
Exemple 1: disque magnétique.



"Ring" writing element

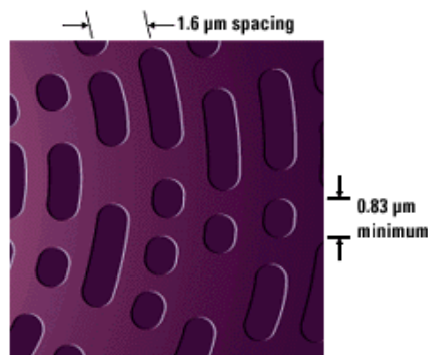


"Monopole" writing element

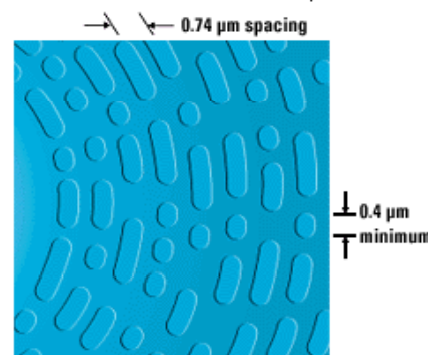


Exemple 2: Déformation géométrique de surface: le CD-ROM & DVD

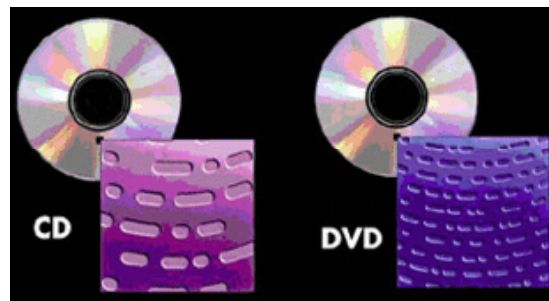
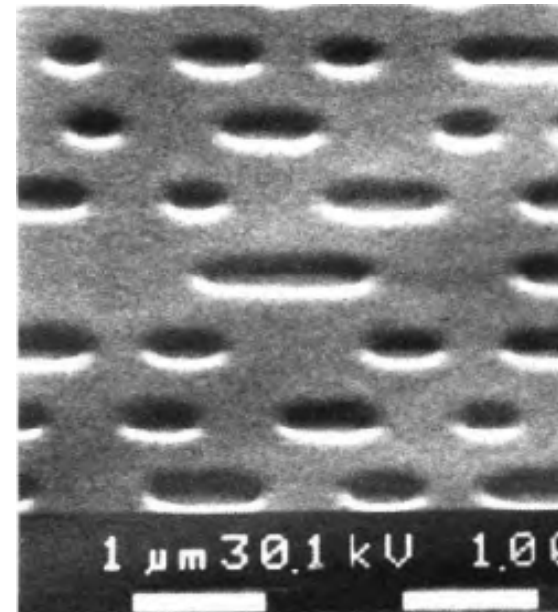
From Computer Desktop Encyclopedia
Reproduced with permission.
© 1998 C-Cube Microsystems



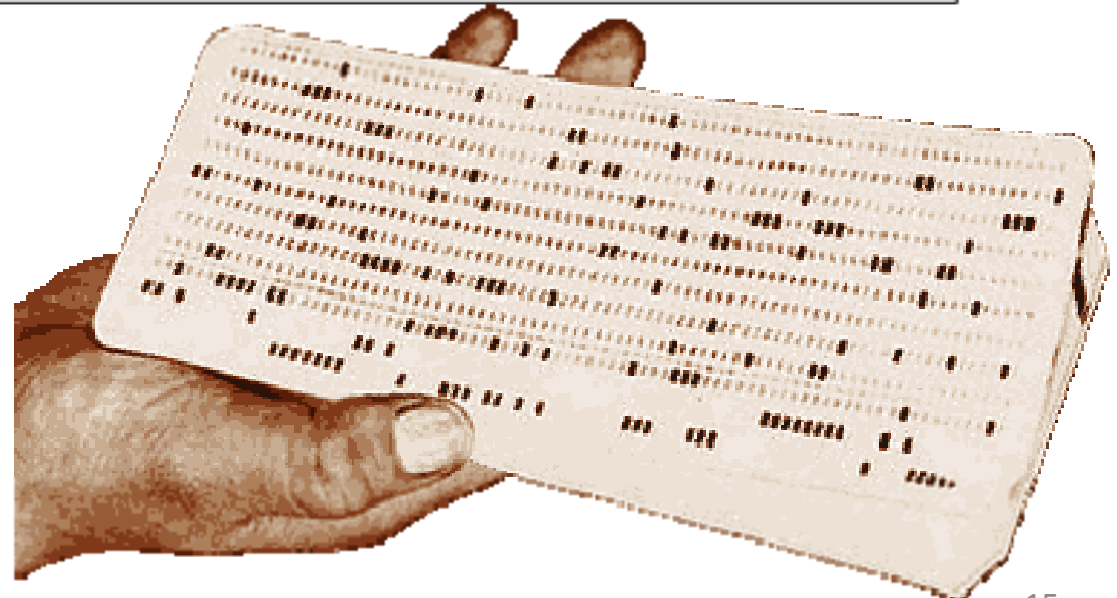
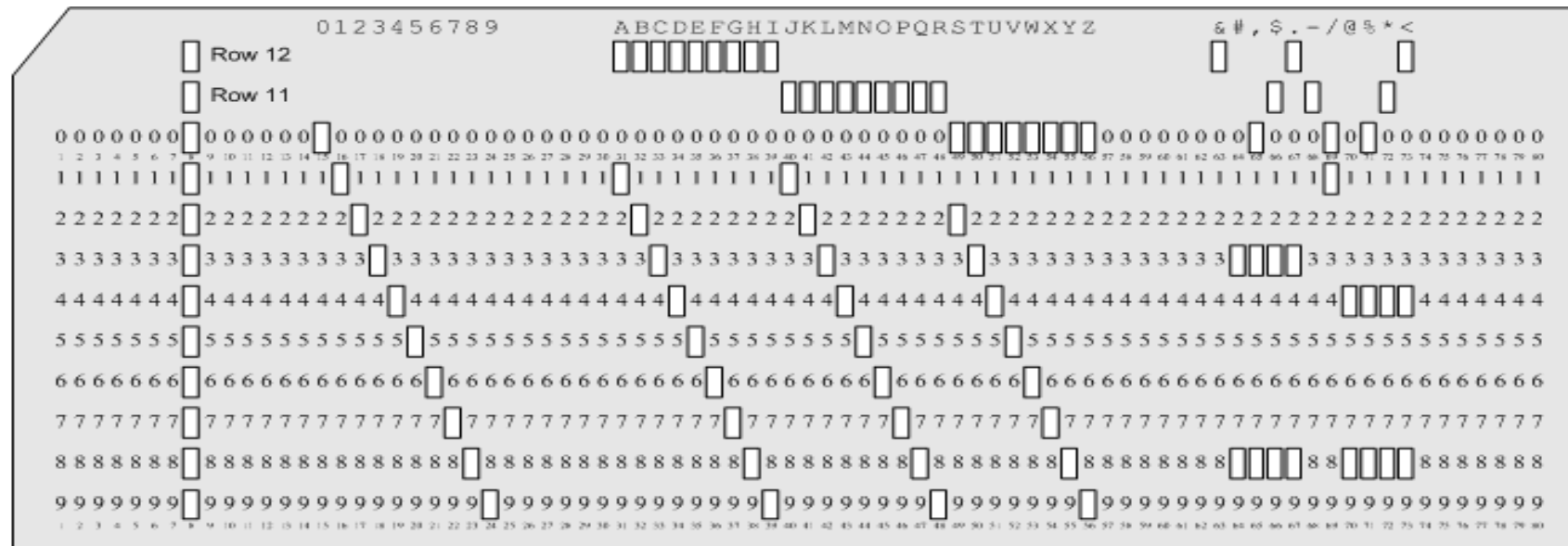
CD-ROM



DVD



Exemple 3: déformation géométrique de surface, la carte perforée (du musée).



Les multiples du bit: l'octet, le mot.

Les multiples de l'octet: Ko, Mo, Go, To, Po

- ✓ Huit bits font un octet (**BYTE**)(de quoi coder un caractère dans son format simple)
- ✓ Le mot (**WORD**) est de longueur variable, non normalisée. On peut trouver les vocables: **half word** et **double word**.
- ✓ Les multiples de l'octet: En tant que professionnel, vous devez maîtriser ces multiples, comme vous avez assimilé le gramme, le litre, le mètre et leurs multiples.

Le multiple, son abréviation	Son équivalent en octet	Approximativement égal à
Le Kilo octet: Ko ou KB (Byte)	2^{10} octets $\simeq 10^3$ octets	Une page de texte
Le Méga octet: Mo ou MB	2^{20} octets $\simeq 10^6$ octets	1000 pages, un livre
Le Giga octet: Go ou GB	2^{30} octets $\simeq 10^9$ octets	1000 livres, Une petite Librairie
Le Téra octet: To ou TB	2^{40} octets $\simeq 10^{12}$ octets	1 million de livres, grande Biblio.
Le Peta octet: Po ou PB	2^{50} octets $\simeq 10^{15}$ octets	1000 grandes Bibliothèques.
L' Exa octet: Eo ou EB	2^{60} octets $\simeq 10^{18}$ octets	1 million de grandes bibliothèques
Le Zeta et le Yotta etc.		

Représentation des données:

Codage et Format:

***Le codage** est la fonction établissant une correspondance entre la représentation externe de la donnée (e.g., A, 36) et sa représentation interne, sous forme d'une suite de bits (e.g., 01000001, 00100100), et tout cela conformément à une norme, standard, règles pour les besoins d'interopérabilité et d'échange.*

*On parle de **format** lorsque la donnée est plutôt complexe:*

Format de données textuelles riche: rtf, doc, docx, pdf ...

Format de données sonores: wav, midi, mp3 ...

Format de données images: bmp, png, gif, ...

Format de données audio/vidéo: avi, mov, mp4 ...

Représentation des données.

Système de numération: les bases de numération 10, 2 et 16

Le système de numération décimal (dit également en base dix) utilise dix chiffres (0 à 9) pour représenter les nombres.

La règle d'écriture des nombres est dite positionnelle. C'est à dire que chaque chiffre est multiplié par dix (la base) élevée à une puissance qui correspond à sa position dans la transcription du nombre. Les positions sont numérotées en commençant par zéro, à partir de la gauche.

Le nombre 3567 signifie 3 milliers, 5 centaines, 6 dizaines et 7 unités ou sous forme polynomiale:

$$(3567)_{10} = (3 \cdot 10^3 + 5 \cdot 10^2 + 6 \cdot 10^1 + 7 \cdot 10^0)_{10}$$

La même règle est valide lorsque le nombre comporte une partie fractionnaire. Cependant les puissances de dix sont négatives.

$$(48,92)_{10} = (4 \cdot 10^1 + 8 \cdot 10^0 + 9 \cdot 10^{-1} + 2 \cdot 10^{-2})_{10}$$

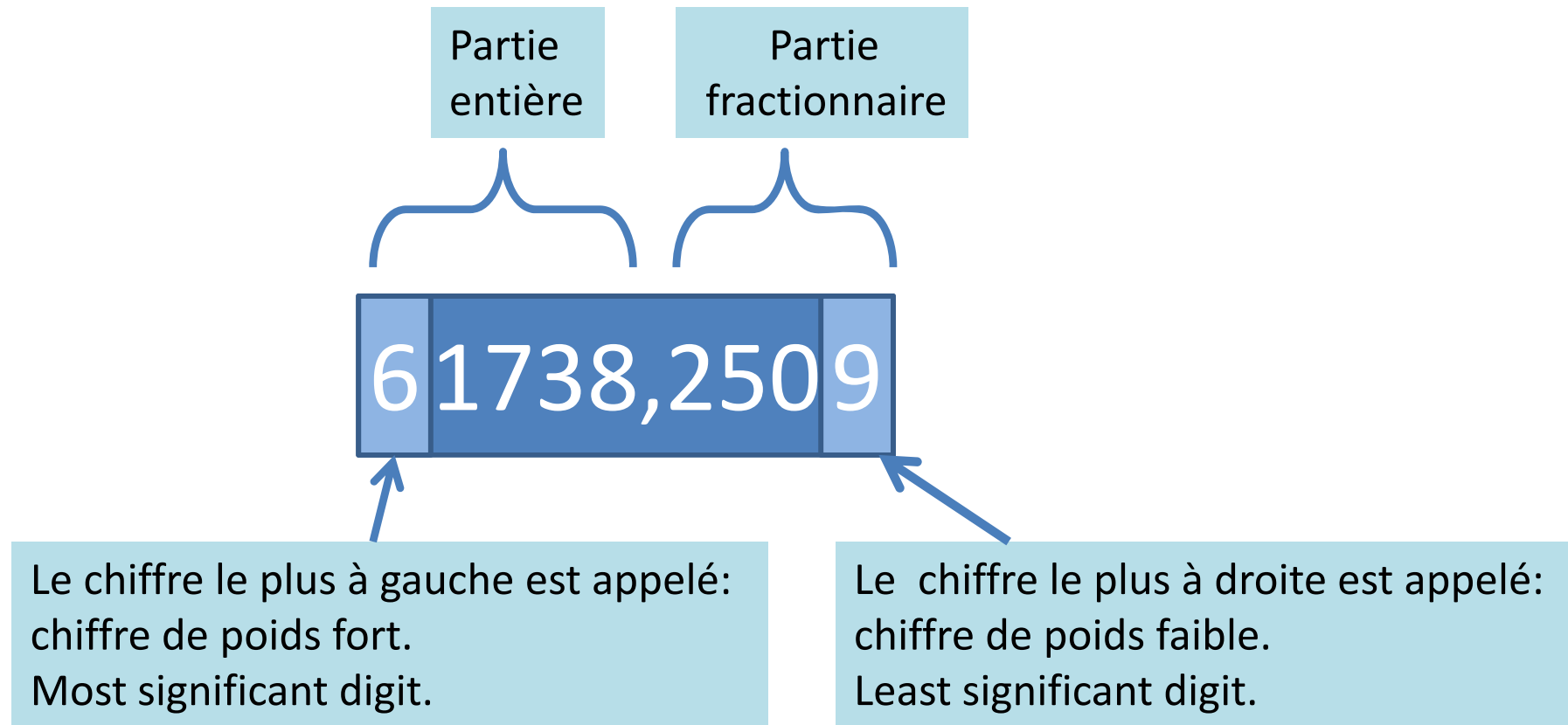
En règle générale, si la transcription d'un nombre N en base dix est:

$(\dots d_2 d_1 d_0, d_{-1} d_{-2} d_{-3} \dots)_{10}$ alors sa valeur est:

$$N = \sum_i (d_i \cdot 10^i)$$

Représentation des données.

Terminologie.



Représentation des données.

Système binaire ou base deux.

D'une manière générale un nombre N qui s'écrit en base b sur $(n+p)$ chiffres:

$(a_{n-1}a_{n-2} \dots a_2a_1a_0, a_{-1}a_{-2}a_{-3} \dots a_{-p+1}a_{-p})_b$ aura pour valeur:

$$N = \sum_{k=-p}^{n-1} a_k b^k$$

La valeur est sous entendue en base humaine.

Si la base est deux, donc deux chiffres sont utilisés: 0 et 1.

Les chiffres 0 et 1 ont la même signification qu'en base dix:

$(0)_2 = (0)_{10}$ et $(1)_2 = (1)_{10}$.

Au delà, chaque chiffre binaire a une valeur qui dépend de sa position.

Les chiffres 0 et 1 sont aussi appelés **bits**.

Représentation des données.

Système binaire ou base deux.

Exemples:

$$(10)_2 = (1 * 2^1) + (0 * 2^0) = (2)_{10}$$

$$(11)_2 = (1 * 2^1) + (1 * 2^0) = (3)_{10}$$

$$(100)_2 = (1 * 2^2) + (0 * 2^1) + (0 * 2^0) = (4)_{10}$$

Désormais, et pour éviter toute confusion, les nombres sont mis entre parenthèses et la base est mise en indice.

Les valeurs fractionnaires sont représentées avec des puissances négatives de la base:

$$(1001,11)_2 = (1 * 2^3) + (0 * 2^2) + (0 * 2^1) + (1 * 2^0) + (1 * 2^{-1}) + (1 * 2^{-2}) = (9,75)_{10}$$

En règle générale, si la transcription d'un nombre N en base deux est:

$(... b_2 b_1 b_0, b_{-1} b_{-2} b_{-3} ...)_2$ alors sa valeur est:

$$N = \sum_i (b_i * 2^i)$$

Représentation des données.

Compter en base deux.

Application:

Exercez vous à compter en base deux de zéro à quinze.

Remarquez comment les bits changent.

✓ Les bits de poids faible changent une fois sur deux.

✓ Les bits de poids fort changent tout les huit fois.

DECIMAL NUMBER	BINARY NUMBER			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Représentation des données.

Conversions entre bases dix et deux et inversement.

- ✓ La conversion de la base deux vers la base dix est le calcul de la valeur de la chaîne (configuration) binaire tel que suggéré précédemment .

$$N = \sum_i (b_i * 2^i)$$

- ✓ La conversion de la base dix vers la base deux se fait comme suit:

Pour la partie entière, on procède par divisions successives.

Si cette partie, disons N s'écrit en base 2 $(b_{m-1}b_{m-2}...b_2b_1b_0)_2$ avec $b_i = 0$ ou 1.

Alors sa valeur est: $N = (b_{m-1} * 2^{m-1}) + (b_{m-2} * 2^{m-2}) + \dots + (b_1 * 2^1) + b_0$

On peut écrire $N = 2 * ((b_{m-1} * 2^{m-2}) + (b_{m-2} * 2^{m-3}) + \dots + (b_2 * 2^1) + b_1) + b_0$

Si on divise N par 2 on obtient:

Un quotient: $(b_{m-1} * 2^{m-2}) + (b_{m-2} * 2^{m-3}) + \dots + (b_2 * 2^1) + b_1$

Et un reste : b_0 . Donc le bit de poids faible b_0 est le reste de la première division de N par 2.

Représentation des données.

La conversion de la base dix vers la base deux (suite).

On peut de nouveau écrire le quotient comme $2 \cdot N_2 + b_1$ et si on divise cette quantité par 2, on obtient de nouveau un quotient et un reste: b_1 .

La procédure se poursuit jusqu'à ce que le quotient soit nul.

Pour la partie fractionnaire, on procède par multiplication successives.

Cette partie s'écrit $0, b_{-1} b_{-2} b_{-3} \dots$ avec $b_i = 0$ ou 1 .

Sa valeur est $(b_{-1} * 2^{-1}) + (b_{-2} * 2^{-2}) + (b_{-3} * 2^{-3}) \dots$

Si on multiplie cette valeur par 2, on obtient:

Une partie entière: (b_{-1}) .

Et une partie fractionnaire.

On répète la multiplication par 2 de cette dernière, on obtient de nouveau une partie entière et une partie fractionnaire etc. On arrête lorsque la partie fractionnaire devienne nulle ou lorsqu'on a épuisé la précision.

Représentation des données.

La conversion de la base dix vers la base deux (suite), exemple.

$$(41, 6875)_{10} = (?)_2$$

$$41/2 = 20 + 1$$

$$20/2 = 10 + 0$$

$$10/2 = 5 + 0$$

$$5/2 = 2 + 1$$

$$2/2 = 1 + 0$$

$$1/2 = 0 + 1$$

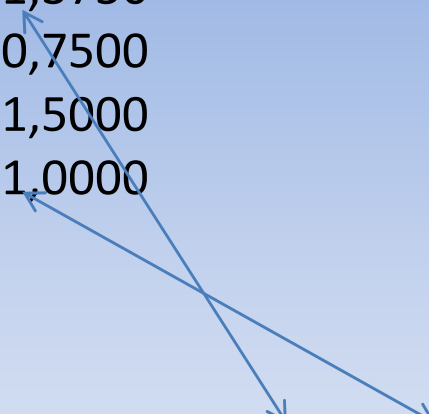
$$(41)_{10} = (101001)_2$$


$$0.6875 \times 2 = 1,3750$$

$$0.3750 \times 2 = 0,7500$$

$$0.7500 \times 2 = 1,5000$$

$$0.5000 \times 2 = 1,0000$$

$$(0,6875)_{10} = (0,1011)_2$$


$$(41, 6875)_{10} = (101001,1011)_2$$

Représentation des données.

Système hexadécimal ou base 16.

- ✓ Lorsqu'on est proche des couches basses de la machine à information et qu'on s'intéresse à ses bits, ses octets et ses mots, on est appelé à manipuler de longues chaînes de bits où le risque de se tromper, dans leurs écritures, leurs recopies ou leurs dictées, est grand. D'où la base hexadécimale ou base 16.
- ✓ Pour écrire un nombre en base 16, il faut 16 chiffres: on recycle les chiffres 0 à 9 auxquels on ajoute A,B,C,D,E et F, dans leurs versions majuscules ou minuscules.
- ✓ Les conversions de la base 2 vers la base 16 et inversement sont triviales.
- ✓ On peut aisément transformer la valeur d'une configuration binaire en une autre valeur où apparaisse la base 16:

$$N = \sum_i (b_i * 2^i) = \sum_j (h_j * 16^j)$$

avec $b_i = 0$ ou 1 et h_j dans $0, 1, \dots, 9, A, B, C, D, E, F$

Représentation des données.

Systeme hexadécimal ou base 16 (suite).

En termes pratiques, il faut d'abord comprendre et retenir le contenu du tableau, simple, de conversion ci contre.

Conversion binaire → hexadec (base 16, soit 2⁴)

Consiste à découper en blocs de 4 chiffres

Exemple

$$(1010011101)_2 \quad \begin{array}{ccc} 0010 & 1001 & 1101 \\ 2 & 9 & D \end{array} = (29D)_{16}$$

Conversion hexadec (base 16, soit 2⁴) → binaire

Consiste à décomposer chaque chiffre hexadécimal en 4 bits

$$\text{Exemple } (A48)_{16} = (1010 \ 0100 \ 1000)_2$$

Dans la littérature on aborde également la base 8, dite octale. Elle joue le même rôle que la base 16. on lui préfère la base hexadécimale en raison du caractère condensé de l'écriture des chaines binaires.

Décimal	Binaire	Hexadécimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

Représentation des données.

Exercices.

✓ **Exercice 1:** Complétez le tableau suivant:

Décimal	Binaire	Hexadécimal
1	000000001	001
27		
	111001010	
		085

✓ **Exercice 2:**

- Quelle est l'écriture générale, en binaire, de $(2^n)_{10}$ et $(2^n - 1)_{10}$? $n > 0$.
- Peut-on convertir $(256)_{10}$ en binaire sur 8 bits seulement?
- Inversement, sur **n bits**, quel est le plus grand entier décimal qu'on peut représenter?