

INTRODUCTION A L'INFORMATIQUE

1- Définition de l'informatique

Le terme « informatique » date de 1962. Il vient de la contraction des mots « information » et « automatique ». L'informatique est la science du traitement automatique de l'information. L'histoire de l'informatique est justement marquée par la volonté des hommes d'automatiser certaines tâches longtemps réalisées à la main, en particulier le calcul

Voici quelques domaines d'application de l'informatique : L'enseignement, Le commerce, La médecine L'agriculture, ...etc

2- Evolution de l'informatique et des ordinateurs

C'est en 1642 que le philosophe et mathématicien Blaise Pascal construit la première machine à calculer (la Pascaline, aussi appelée roue Pascal), capable d'effectuer des additions et des soustractions.

Un peu plus tard, en 1833, l'Anglais Charles Babbage reprend ce principe et construit une machine encore plus élaborée que les machines à calculer de l'époque : la sienne est capable d'exécuter toutes les opérations et de stocker les résultats. C'est à son associée, la mathématicienne Ada Byron, que l'on doit un peu plus tard les principes de base de la programmation.

Selon les langages de programmation et la technologie utilisée, l'histoire de l'informatique est classée par génération :

La première génération (1946)

Construction du premier calculateur électronique ENIAC, utilisant le langage de programmation binaire.

La deuxième génération (1956)

Premier calculateur transistorisé TRADIC, apparition des langages de programmation Assembleur, Algol, Cobol.

La troisième génération (1964)

Premier ordinateur à circuits intégrés (IBM360), langages de programmation Basic, Pascal, C.

La quatrième génération (1976)

Naissance des micro-processeurs et des micro-ordinateurs de bureau, logiciels de bureautique.

□ **La cinquième génération (1984)**

Naissance des micro-ordinateurs familiaux et apparition des systèmes d'exploitations (MS_DOS, MacOS).

□ **La sixième génération (1992)**

Amélioration des fréquences des processeurs 33 MHZ, RAM 4Mo. Diffusion de Windows à grande échelle et exploration d'Internet par le grand public.

□ **Génération actuelle**

Apparition des ordinateurs portables puissants, réseaux sans fil,...

3- Les systèmes de codage des informations

Quelle que soit la nature de l'information traitée par un ordinateur (image, son, texte, vidéo), elle l'est toujours sous la forme d'un ensemble de nombres écrits en base 2, par exemple 01001011. Le terme bit (b minuscule dans les notations) signifie « binary digit », c'est-à-dire 0 ou 1 en numérotation binaire. Il s'agit de la plus petite unité d'information manipulable par une machine numérique. Il est possible de représenter physiquement cette information binaire par un signal électrique ou magnétique, qui, au-delà d'un certain seuil, correspond à la valeur 1.

L'octet (en anglais byte ou B majuscule dans les notations) est une unité d'information composée de 8 bits. Il permet par exemple de stocker un caractère comme une lettre ou un chiffre.

Une unité d'information composée de 16 bits est généralement appelée mot (en anglais word). Une unité d'information de 32 bits de longueur est appelée mot double (en anglais double word, d'où l'appellation dword). Beaucoup d'informaticiens ont appris que 1 kilooctet valait 1024 octets.

. Voici les unités standardisées :

- Un kilooctet (ko) = 2^{10} octets = 1024 octets
- Un mégaoctet (Mo) = 2^{20} octets = 1024 koctets
- Un gigaoctet (Go) = 2^{30} octets = 1024 Moctets
- Un téraoctet (To) = 2^{40} octets = 1024 Goctets
- Un pétaoctet (Po) = 2^{50} octets = 1024 Toctets

Pour qu'une information numérique soit traitée par un circuit, elle doit être mise sous forme adaptée à celui-ci. Pour cela Il faut choisir un système de numération, de nombreux systèmes de numération sont utilisés en technologie numérique ; tels que les systèmes : Décimal (base 10), Binaire (base 2), Tétral (base 4), Octal (base 8) et Hexadécimal (base 16).

Dans cette partie de cours on va étudier les systèmes décimal, binaire et hexadécimal ainsi que les conversions entre ces différentes bases de codage.

3-1 Le système décimal

Les nombres que nous utilisons habituellement sont ceux de la base 10 (système décimal). Nous disposons de dix chiffres différents de 0 à 9 pour écrire tous les nombres. D'une manière générale, **toute base N est composée de N chiffre de 0 à N-1.**

Soit un nombre décimal $N = 2348$. Ce nombre est la somme de 8 unités, 4 dizaines, 3 centaines et 2 milliers. Nous pouvons écrire

$$N = (2 \times 1000) + (3 \times 100) + (4 \times 10) + (8 \times 1)$$

$$\mathbf{2348 = (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (8 \times 10^0)}$$

10 représente la base et les puissances de 0 à 3 le rang de chaque chiffre. Quelque soit la base, le chiffre de droite est celui des unités. Celui de gauche est celui qui a le poids le plus élevé. Cette écriture s'appelle **forme polynomiale.**

3-2 Le système binaire

Dans les domaines de l'automatisme, de l'électronique et de l'informatique, nous utilisons la base 2. Tous les nombres s'écrivent avec deux chiffres uniquement (0 et 1). De même que nous utilisons le système décimal parce que nous avons commencé à compter avec nos dix doigts, nous utilisons le binaire car les systèmes technologiques ont souvent deux états stables.

Un interrupteur est ouvert ou fermé, une diode est allumée ou éteinte, une tension est présente ou absente, une surface est réfléchissante ou pas (CD), un champ magnétique est orienté Nord-Sud ou Sud-Nord (disque dur). A chaque état du système technologique, on associe un état logique binaire.

La présence d'une tension sera par exemple notée 1 et l'absence 0. Le chiffre binaire qui peut prendre ces deux états est nommé "Bit" (Binary digit).

Avec un bit nous pouvons coder deux états

| |
|---|
| 0 |
| 1 |

Avec deux bits nous pouvons coder quatre états

| | |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

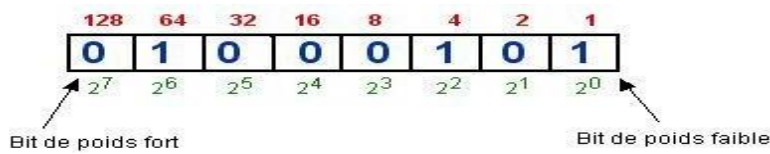
Avec trois bits nous pouvons coder huit états

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

A chaque nouveau bit, le nombre de combinaisons possibles est doublé. Ce nombre est égal à **2 puissance N** (N étant le nombre de bits).

Un groupe de bits est appelé un mot, un mot de huit bits est nommé un octet (byte). Avec un octet, nous pouvons écrire 2 puissance 8 = **256 nombres binaires de 0 à 255**

Description d'un octet.



Un 1 dans une case représente la valeur décimale qui est au dessus.

Correspondance entre binaire et décimal.

Conversion d'un nombre binaire en décimal.

Il suffit de faire la somme des poids de chaque bit à 1, c'est-à-dire écrire le nombre sous forme polynomiale. Le nombre ci dessus est égal à

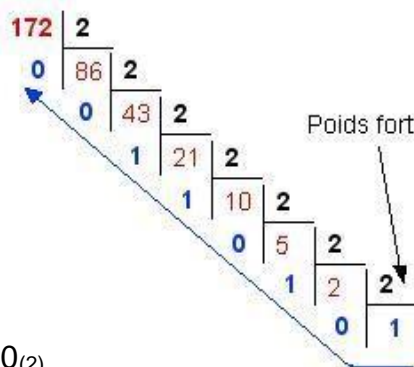
$$(01000101)_2 = (1 \times 2^0 + 1 \times 2^2 + 1 \times 2^6)_{10} = (69)_{10}$$

$$(111011)_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (61)_{10}$$

$$(10011.1101)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = (19.75)_{10}$$

Conversion d'un nombre décimal en binaire (exemple : N = 172).

Méthode par divisions



$$172_{(10)} = 10101100_{(2)}$$

Conversion d'un nombre décimal à virgule

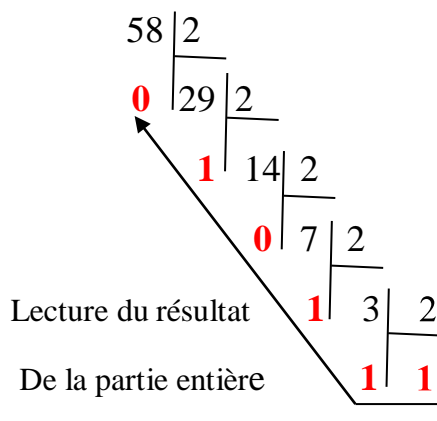
Pour convertir un nombre décimal à virgule dans une base B quelconque, il faut :

1-Convertir la partie entière en effectuant des divisions successives par **2** (comme nous l'avons vu précédemment).

2-Convertir la partie fractionnaire en effectuant des multiplications successives par **2** et en conservant à chaque fois le chiffre devenant entier.

Exemple : Conversion du nombre (58,625) en base 2

Conversion de la partie entière



Conversion de la partie fractionnaire

$$0.625 \times 2 = \mathbf{1} .25$$

$$0.25 \times 2 = \mathbf{0} .5$$

$$0.5 \times 2 = \mathbf{1} .0$$

lecture du résultat
 de la partie
 fractionnaire

$$(58.625)_{10} = (111010.101)_2$$

Parfois en multipliant la partie fractionnaire par la base 2 ou n'importe quelle base B on n'arrive pas à convertir toute la partie fractionnaire. Ceci est dû essentiellement au fait que le nombre à convertir n'a pas un équivalent exacte dans la base B et sa partie fractionnaire est cyclique ou infinie.

Exemple : $(0.15)_{10} = (?)_2$

$$0.15 \times 2 = 0.3$$

$$0.3 \times 2 = 0.6$$

$$0.6 \times 2 = 1.2$$

$$0.2 \times 2 = 0.4$$

$$0.4 \times 2 = 0.8$$

$$0.8 \times 2 = 1.6$$

$$0.6 \times 2 = 1.2$$

$$0.2 \times 2 = 0.4$$

$$0.4 \times 2 = 0.8$$

$$0.8 \times 2 = 1.6$$

$$(0.15)_{10} = (0.00\underline{1001\ 1001})_2$$

On dit que le nombre $(0.15)_{10}$ est cyclique dans la base 2 de période **1001**.

3-3 Le système quaternaire (base 4) :

Ce système appelé aussi base 4 comprend quatre chiffres possibles $\{0, 1, 2, 3\}$.

Un nombre quaternaire peut s'écrire sous la forme polynomiale comme le montre les exemples suivant :

$$(2331)_4 = 2 \times 4^3 + 3 \times 4^2 + 3 \times 4^1 + 1 \times 4^0 = (189)_{10}$$

$$(130.21)_4 = 1 \times 4^2 + 3 \times 4^1 + 1 \times 4^0 + 2 \times 4^{-1} + 1 \times 4^{-2} = (29.5625)_{10}$$

Conversion d'un nombre décimal en base 4

$$(105.125)_{10} = (?)_4$$

| Partie entière | partie fractionnaire | |
|---|--|-----------------------------|
| $ \begin{array}{r} 105 \quad \quad 4 \\ \hline (1) \quad 26 \quad \quad 4 \\ \hline (2) \quad 6 \quad \quad 4 \\ \hline (2) \quad (1) \end{array} $ | $ \begin{array}{l} 0.125 \times 4 = (0) .5 \\ 0.5 \times 4 = (2) .0 \end{array} $ | lecture du résultat ↓ |
| lecture du résultat | $(105.125)_{10} = (1221.02)_4$ | |

3-4 Le système octal (base 8) :

Le système octal ou base 8 comprend huit chiffres qui sont $\{0, 1, 2, 3, 4, 5, 6, 7\}$. Les chiffres 8 et 9 n'existent pas dans cette base. Ecrivons à titre d'exemple, les nombres 45278 et 1274.6328 :

$$(4527)_8 = 4 \times 8^3 + 5 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 = (2391)_{10}$$

$$(1274.632)_8 = 1 \times 8^3 + 2 \times 8^2 + 7 \times 8^1 + 4 \times 8^0 + 6 \times 8^{-1} + 3 \times 8^{-2} + 2 \times 8^{-3} = (700.8007813)_{10}$$

Conversion d'un nombre décimal en base 4

$$(110.625)_{10} = (?)_8$$

| | |
|--|--|
| <p>Partie entière</p> $\begin{array}{r l} 110 & 8 \\ \hline 6 & 13 \\ \hline & 8 \\ & 5 & 1 \end{array}$ | <p>partie fractionnaire</p> $0.625 \times 8 = 5.0$ |
|--|--|

$(110.625)_{10} = (156.5)_8$

3-5 Le système hexadécimal

La manipulation des nombres écrits en binaire est difficile pour l'être humain et la conversion en décimal n'est pas simple. On utilise aussi très souvent *le système hexadécimal* (base 16) du fait de sa simplicité d'utilisation et de représentation pour les mots machines (il est bien plus simple d'utilisation que le binaire).

Pour écrire les nombres en base 16 nous devons disposer de 16 chiffres, pour les dix premiers, nous utilisons les chiffres de la base 10, pour les suivants nous utiliserons des lettres de l'alphabet.

| | | | | | | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Décimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| Hexadécimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

Les règles sont ici aussi les mêmes que pour le décimal.

$$A3F_{(16)} = (A \times 16^2) + (3 \times 16^1) + (F \times 16^0)$$

$$A3F_{(16)} = (10 \times 256) + (3 \times 16) + (15 \times 1)$$

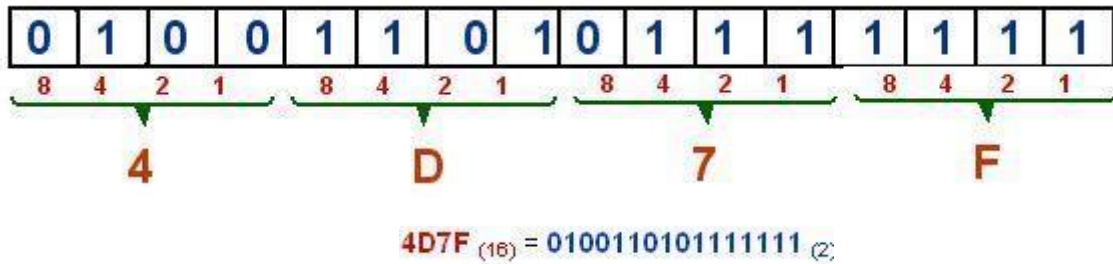
$$A3F_{(16)} = 2560 + 48 + 15 = 2623_{(10)}$$

Correspondance entre binaire et hexadécimal.

La conversion du binaire en hexadécimal est très simple, c'est d'ailleurs la raison pour laquelle nous utilisons cette base.

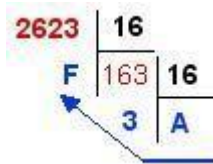
Il suffit de faire correspondre un mot de quatre bits (quartet) à chaque chiffre hexadécimal.

Conversion d'un mot de 16 bits entre binaire et hexadécimal



Correspondance entre décimal et hexadécimal.

La méthodes par divisions s'applique comme en binaire (exemple : $N = 2623$).



$$2623_{(10)} = A3F_{(16)}$$

En général pour faire La conversion d'un nombre d'une base quelconque B_1 vers une autre base B_2 il faut passer par la base 10. Mais si la base B_1 et B_2 s'écrivent respectivement sous la forme d'une puissance de 2 on peut passer par la base 2 (binaire) :

Base quaternaire (base 4) : $4=2^2$ chaque chiffre quaternaire se convertit tout seul sur 2 bits.

Base octale (base 8) : $8=2^3$ chaque chiffre octal se convertit tout seul sur 3 bits.

Base hexadécimale (base 16) : $16=2^4$ chaque chiffre hexadécimal se convertit tout seul sur 4 bits.

Exemples :

$$(1\ 0\ 2\ 2\ 3)_4 = (01\ 00\ 10\ 10\ 11)_2$$

$$(6\ 5\ 3\ 0.7)_8 = (110\ 101\ 011\ 000.111)_2$$

$$(101\ 010\ 100\ 111\ 000)_2 = (5\ 2\ 4\ 7\ 0)_8$$

$$(1101\ 1000\ 1011\ 0110.011)_2 = (D\ 8\ B\ 6.6)_{16}$$

$$(11\ 10\ 01\ 00\ 10.101)_2 = (3\ 2\ 1\ 0\ 2.22)_4$$

3-6 Les nombres signés

Nous avons jusqu'à présent parlé de nombres entiers naturels. Ils ne peuvent par nature qu'être positifs ou nuls. Envisageons maintenant les nombres entiers relatifs ou autrement dit, munis d'un signe '+' ou '-'. En décimal,

+1, +2, +3 etc. sont des nombres positifs. Ils sont supérieurs à 0 ($n > 0$)

-1, -2, -3 etc. sont des nombres négatifs. Ils sont inférieurs à 0 ($n < 0$)

De même en binaire,

+1, +10, +11, +100, +101 etc. sont des nombres binaires positifs,

-1, -10, -11, -100, -101 etc. sont des nombres binaires négatifs.

Le problème est que les circuits électroniques digitaux ne peuvent enregistrer que des 0 ou des 1 mais pas de signes + ou -. Le seul moyen est alors de convenir que si un nombre est susceptible d'être négatif on lui réserve un bit pour indiquer le signe.

Plusieurs méthodes sont utilisées pour représenter les nombres négatifs dans un ordinateur, parmi lesquelles nous citons : la représentation en signe et valeur absolue (SVA), le complément à 1 (CP1) et le complément à 2 (CP2). On cherche une représentation binaire des entiers négatifs pour que l'addition de deux nombres entiers relatifs fonctionne, c'est la représentation complément à 2 (CP2).

Comment calculer les codes des nombres négatifs en CP2 sur 8 bits:

Le calcul se fait en trois étapes :

1° calcul du code binaire du nombre sur 8 bits

2° Calcul du complément à 1 = Remplacer tous les 0 par des 1 et tous les 1 par des 0.

3° Calcul du complément à 2 = Ajouter 1 au complément à 1

Exemples : 1-comment écrire -4 en CP2 ?

$$+4 = 0000\ 0100_{(2)}$$

Le complément à 1 de ce code est CP1(+4)=1111 1011₍₂₎

Ajoutons 1 à ce code pour obtenir son complément à 2

$$1111\ 1011_{(2)} + 1 = 1111\ 1100_{(2)} = (-4)_{CP2}$$

2-comment écrire -24 en CP2 ?

Le nombre $+24 = (11000)_2 = (0001\ 1000)_{CP2}$

Le complément à 1 de ce code est $CP1(+24) = 1110\ 0111$

Ajoutons 1 à ce code pour obtenir son complément à 2

$1110\ 0111_{(2)} + 1 = 1110\ 1000_{(2)} = (-24)_{CP2}$

ainsi $(-24)_{10} = (1110\ 1000)_{CP2}$

Le bit le plus à gauche du code CP2 est celui qui va représenter le signe. Signe négatif si ce bit vaut 1, signe positif quand ce bit vaut 0.

Le plus grand nombre signé sur 8 bits est **+127 (01111111)**

Le plus petit nombre signé sur 8 bits est **-128 (10000000)**

-128 à +127 => 256 combinaisons (2 puissance 8)

Ce qui s'applique sur 8 bits s'applique aussi sur 4, 16, ... bits.

Notifications : L'addition binaire :

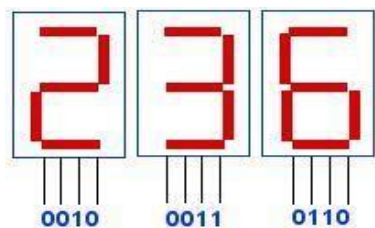
L'addition de deux bits binaire se réalise selon la spécification suivante :

| bit1 | bit2 | résultat | retenue |
|------|------|----------|---------|
| 0 | 0 | = 0 | 0 |
| 0 | 1 | = 1 | 0 |
| 1 | 0 | = 1 | 0 |
| 1 | 1 | = 0 | 1 |

L'addition binaire de deux nombres s'effectue bit à bit de droite à gauche, en reportant les retenues

Le décimal codé binaire (code DCB)

Ce codage est destiné à l'affichage de valeurs décimales, chaque digit doit être codé en binaire sur 4 bits (unités, dizaines, centaines ...).



Ce codage ne permet aucun calcul, il est uniquement destiné à la saisie et à l'affichage de données

Le codage ASCII

Le binaire permet de coder les nombres que les systèmes informatiques peuvent manipuler. Cependant, l'ordinateur doit aussi utiliser des caractères alphanumériques pour mémoriser et transmettre des textes. Pour coder ces caractères, on associe à chacun d'entre eux un code binaire, c'est le codage ASCII (American Standard Code for Information Interchange).

4-Principe de fonctionnement d'un ordinateur

Lorsque l'ordinateur démarre, il commence par lancer un logiciel spécial que l'on appelle l'OS (par exemple Windows, Linux ou MacOS, etc... Sans cet OS, vous ne pouvez pas utiliser l'ordinateur et l'ordinateur ne serait qu'une machine qui ne servirait pas à grand chose. C'est l'OS qui va interpréter vos commandes et vous afficher les résultats.

Les OS modernes affichent généralement un écran d'accueil (appelé bureau) contenant toutes vos icônes. Ces icônes vont permettre de lancer les logiciels en cliquant dessus.

Voici comment ça marche quand vous lancez un logiciel étape par étape.

1. Vous cliquez (ou double cliquez) sur l'icône.
2. L'ordinateur va alors chercher dans le disque dur vers quel logiciel pointe cette icône.
3. Il va lire le logiciel à partir du disque dur car celui-ci est enregistré dessus.
4. Tout ce qui sera lu sera chargé dans la mémoire vive (RAM).
5. A partir de la RAM, le processeur va exécuter le logiciel instruction après instruction.
6. Une fois que vous fermerez le programme, celui-ci sera effacé de la RAM.
7. Mais bien évidemment, il reste présent sur le disque dur pour une prochaine utilisation.

La RAM, le processeur et le disque dur sont des composants de l'unité centrale de l'ordinateur.

5-Partie matériel d'un ordinateur

On peut citer parmi les composants d'un ordinateur :

- La carte mère
- Le processeur
- La mémoire vive
- Le disque dur
- Le lecteur et/ou graveur de CD-Rom et/ou DVD-Rom

carte mère

Carte principale sur laquelle on retrouve tous les composants nécessaires au fonctionnement d'un ordinateur, dont le processeur, la mémoire centrale, les bus et les connecteurs d'extension destinés à recevoir des cartes d'extension et gère les différentes interfaces avec les ports USB pour les périphériques externes.

Le processeur

Souvent qualifié de cerveau de l'ordinateur, le processeur permet de manipuler et de traiter les données qui lui sont fournies. Sa puissance a une influence sur la vitesse d'exécution de vos logiciels et des opérations que vous effectuez.

La mémoire vive

Partenaire de travail du processeur, la mémoire vive (appelée aussi **RAM**) stocke temporairement les données à traiter par le processeur. Ainsi, plus il y a de mémoire de disponible, plus il est possible d'y conserver des données temporairement (ce qui évite d'accéder au disque dur qui est plus lent).

La mémoire vive est vidée à chaque arrêt ou redémarrage de votre ordinateur.

Le disque dur

Contrairement à la mémoire vive, les données stockées sur un disque dur sont permanentes et ne sont pas effacées à l'arrêt de votre ordinateur. C'est donc sur un disque dur que votre système d'exploitation (Linux, MacOS ou Windows), vos logiciels et vos documents sont conservés. C'est donc un espace de stockage permanent, où vous pouvez copier et supprimer des données à volonté.

Le lecteur et/ou graveur de CD-Rom et/ou DVD-Rom

Définition lecteur :

Appareil permettant la lecture, à l'aide d'un rayon laser, des informations enregistrées sur un CD-ROM ou un DVD-Rom

Les lecteurs de CD-ROM et DVD-Rom peuvent être internes (c'est-à-dire intégrés dans l'ordinateur) ou externes (se présentant sous la forme d'un boîtier autonome).

Définition graveur :

Appareil permettant l'enregistrement de disques compacts, à l'unité ou en petite série, par gravure au laser sur un CD ou DVD

Il y a d'autres composants tels que la carte graphique pour l'affichage sur l'écran, la carte son pour gérer les entrées (microphones) et les sorties (hauts parleurs..), la carte réseau qui permet le transfert des données, entre un ordinateur personnel et un réseau ou entre un serveur et un réseau. Permet également un accès à internet dans un réseau domestique ou d'entreprise.

6- Partie système

Nous avons présenté l'ordinateur comme un ensemble de circuits électroniques interconnectés. Pour établir la communication entre ces éléments matériels et l'homme, considéré comme l'utilisateur, il est nécessaire d'avoir une interface appropriée qui rende possible la communication homme-machine, d'où le concept de système d'exploitation.

Le système d'exploitation (*SE*, en anglais *Operating System* ou *OS*) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications de l'utilisateur (traitement de texte, jeu vidéo, ...).

Il existe sur le marché des dizaines de systèmes d'exploitation différents, souvent livrés avec l'appareil informatique. C'est le cas de Windows, Mac OS, Irix, Symbian OS, Linux (pour lequel il existe de nombreuses distributions) ou Android.

INTRODUCTION TO COMPUTER SCIENCE

1-Definition of computing

The term “computer science” dates from 1962. It comes from the contraction of the words “information » And “ automatic “. Computing East there science of treatment automatic of the information. The story of computing East exactly marked bythere will of the men to automate some tasks a long time carried out has there hand,in particular THE calculation

Here is a few areas application of computing : teaching, THEtrade, There medicine agriculture, ...etc

2-Evolution of computing And computers

It was in 1642 that the philosopher and mathematician Blaise Pascal built the first machine has calculate (there Pascaline, Also called wheel Pascal), able to carry out of the additions And of the subtractions.

A little later, in 1833, the Englishman Charles Babbage took up this principle and built a machine Again more elaborate that THE machines has calculate of the time : his is capable of executing all operations and storing the results. It is at her associated, there mathematician Adam Byron, that we must A little more late THEprinciples of base of programming.

Depending on the programming languages and technology used, the history of computing East classified by generation :

☐ There first generation (1946)

Construction of the first ENIAC electronic computer, using the language of binary programming.

☐ There second generation (1956)

First calculator transistorized TRADIC, appearance of the languages of programming Assembler, Algol, Cobol.

☐ There third generation (1964)

First integrated circuit computer (IBM360), programming languages Basic, Pascal, vs.

☐ There fourth generation (1976)

Birth of microprocessors and desktop microcomputers, software of office automation.

□ **There fifth generation (1984)**

Birth of the microcomputers family And appearance of the systems of exploitations (MS_DOS, MacOS).

□ **There sixth generation (1992)**

Improved processor frequencies 33 MHZ, RAM 4MB. Broadcast of Windows to big ladder And exploration from the Internet by great audience.

□ **Generation current**

Appearance computers powerful laptops, wireless networks thread,...

3- THE systems of coding of the information

Whatever the nature of the information processed by a computer (image, sound, text, video), it is always in the form of a set of numbers written in base 2, for example 01001011. The term bit (lowercase b in notations) means " binary digit ", that's to say 0 Or 1 in numbering binary. He is of there more smallunit of information manipulable by a machine digital. He East possible of represent physically this information binary by A signal electric Or magnetic, Who, beyond of a certain threshold, corresponds has there value 1.

The byte (in English byte Or B capital letter In THE ratings) East a unit of information composed of 8 bits. It allows for example to store a character as a letter Or A figure.

A unit of information consisting of 16 bits is generally called a word (in Englishword). A unit of information 32 bits long is called a double word (in English double word, hence the name dword). Many computer scientists have learned that 1 kilobyte was worth 1024 bytes.

. Here are the standardized units :

- A kilobyte (kb) = 2^{10} bytes =1024 bytes
- A megabyte (MB) = 2^{20} bytes =1024 kbytes
- A gigabyte (Go) = 2^{30} bytes = 1024 Mbytes
- A terabyte (To) = 2^{40} bytes=1024 GBytes
- A petabyte (Po) = 2^{50} bytes=1024 Tbytes

For digital information to be processed by a circuit, it must be placed under shape adapted has this one. For that He must choose A system of numeration, of Many number systems are used in digital technology; such as the systems: Decimal (base 10), Binary (base 2), Tetral (base 4), Octal (base 8) and Hexadecimal (base 16).

In this part of the course we will study the decimal, binary and hexadecimal systems. as well as THE conversions between these different basics of coding.

3-1 THE decimal system

THE numbers that We let's use usually are those of there base 10 (decimal system). We have ten different numbers from 0 to 9 to write all Numbers. Generally speaking, **any base N is composed of N digits from 0 to N-1.**

Consider a decimal number $N = 2348$. This number is the sum of 8 units, 4 tens, 3 hundreds and 2 thousands. We can write

$$N = (2 \times 1000) + (3 \times 100) + (4 \times 10) + (8 \times 1)$$

$$\mathbf{2348 = (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (8 \times 10^0)}$$

10 represents the base and the powers of 0 to 3 the rank of each number. Regardless of the base, the digit on the right is that of the units. The one on the left is the one with the weight The highest. This writing is called **shape polynomial.**

3-2 THE system binary

In the fields of automation, electronics and IT, welet's use base 2. All numbers are written with only two digits (0 and 1). Just as we use the decimal system because we started to counting with our ten fingers, we use binary because technological systems often have two states stable.

A switch is open or closed, a diode is on or off, a voltage is present or absent, a surface is reflective or not (CD), a field magnetic is oriented North-South or South-North (hard disk). At each state of the system technological, we partner A state logic binary.

The presence of a voltage will for example be noted as 1 and the absence as 0. The binary number who can take these two states is named "Bit" (Binary digit).

With A bit we can code two states

| |
|---|
| 0 |
| 1 |

With two bits we can code four states

| | |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

With three bits We can code eight states

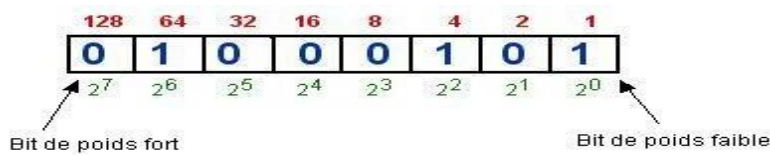
| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

With each new bit, the number of possible combinations is doubled. This number is equal to **2 to the power of N** (N being the number of bits).

A group of bits is called a word, a word of eight bits is called a byte.

With one byte we can write 2 to the power of 8 = **256 binary numbers from 0 to 255**

Description of a byte.



A 1 in a box represented there decimal value Who East At above.

Correspondence between binary And decimal.

Conversion of a number binary in decimal .

Simply add the weights of each bit to 1, i.e. write the number in the form polynomial. THE number this above East equal has

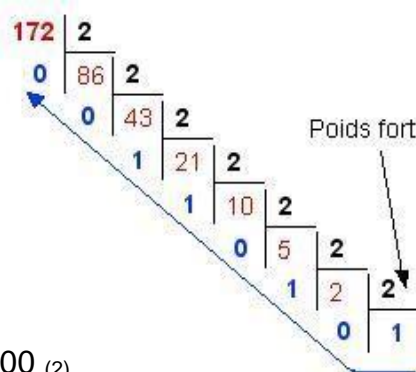
$$(01000101)_2 = (1 \times 2^0 + 1 \times 2^2 + 1 \times 2^6)_{10} = (69)_{10}$$

$$(111011)_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (61)_{10}$$

$$(10011.1101)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = (19.75)_{10}$$

Conversion of a decimal number in binary (example : NOT = 172).

Method by divisions



$$172_{(10)} = 10101100_{(2)}$$

$$(0.15)_{10} = (0.00 \underline{1001 1001})_2$$

We said that the number $(0.15)_{10}$ is cyclic in this basis 2 of period **1001**.

3-3 THE quaternary system (base 4) :

This system called Also basis 4 includes four possible numbers {0, 1, 2, 3}. A quaternary number can be written in polynomial form as shown in following examples :

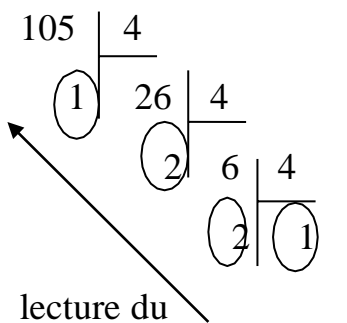
$$(2331)_4 = 2 \times 4^3 + 3 \times 4^2 + 3 \times 4^1 + 1 \times 4^0 = (189)_{10}$$

$$(130.21)_4 = 1 \times 4^2 + 3 \times 4^1 + 1 \times 4^0 + 2 \times 4^{-1} + 1 \times 4^{-2} = (29.5625)_{10}$$

Conversion of a number decimal in base 4

$$(105.125)_{10} = (?)_4$$

Whole part



lecture du
result

part fractional

$$0.125 \times 4 = 0.5$$

$$0.5 \times 4 = 2.0$$

reading of
result

$$(105.125)_{10} = (1221.02)_4$$

3-4 THE octal system (base 8) :

The octal or base 8 system consists of eight digits which are {0, 1, 2, 3, 4, 5, 6, 7}. Numbers 8 and 9 do not exist not in this database. Let's write as an example, the numbers 45278 And 1274.6328 :

$$(4527)_8 = 4 \times 8^3 + 5 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 = (2391)_{10}$$

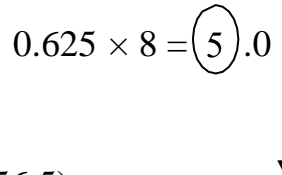
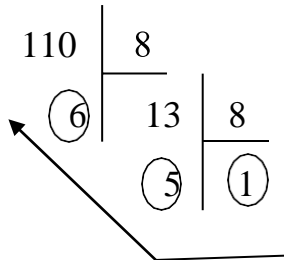
$$(1274.632)_8 = 1 \times 8^3 + 2 \times 8^2 + 7 \times 8^1 + 4 \times 8^0 + 6 \times 8^{-1} + 3 \times 8^{-2} + 2 \times 8^{-3} = (700.8007813)_{10}$$

Conversion of a number decimal in base 4

$(110.625)_{10} = (?)_8$

Whole part

part fractional



$(110.625)_{10} = (156.5)_8$

3-5 THE hexadecimal system

Manipulating numbers written in binary is difficult for humans and Converting to decimal is not simple. We also very often use *the hexadecimal system* (base 16) due to its simplicity of use and representation for machine words (it East GOOD more simple of use that THE binary).

To write numbers in base 16 we must have 16 digits, for the first ten, we use base 10 digits for the following we will use letters of the alphabet.

| | | | | | | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Décimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| Hexadécimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

THE rules are here Also the same that For THE decimal.

$$A3F_{(16)} = (A \times 16^2) + (3 \times 16^1) + (F \times 16^0)$$

$$A3F_{(16)} = (10 \times 256) + (3 \times 16) + (15 \times 1)$$

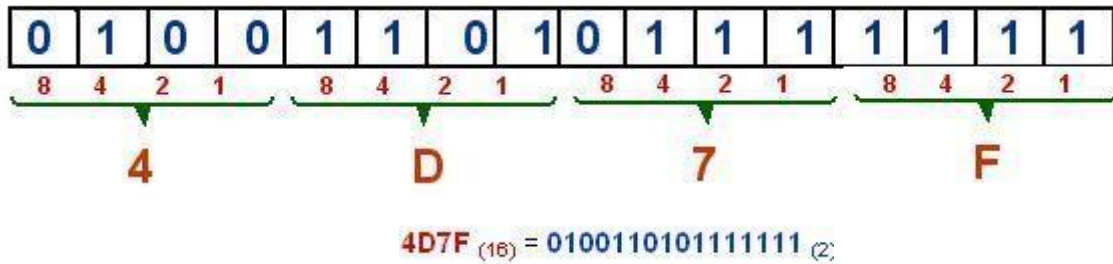
$$A3F_{(16)} = 2560 + 48 + 15 = 2623_{(10)}$$

Correspondence between binary And hexadecimal.

The conversion from binary to hexadecimal is very simple, which is also the reason for which we use this base.

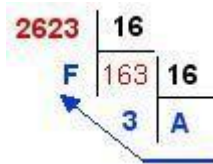
He enough of TO DO match A word of four bits (quartet) has each figure hexadecimal.

Conversion of a word of 16 bits between binary And hexadecimal



Correspondence between decimal And hexadecimal.

There methods by divisions applies as in binary (example : NOT = 2623).



$$2623_{(10)} = A3F_{(16)}$$

In general to convert a number from any base B1 to another base B2 he must pass by there base 10. But if there base B1 And B2 are written respectively below there shape of a power of 2 we can pass by there base 2 (binary) :

Base quaternary (base 4) : $4=2^2$ each figure quaternary se converts All alone on 2 bits. Base

octal (base 8) : $8=2^3$ each number octal converts All alone on 3 bits.

Hexadecimal base (base 16): $16=2^4$ each hexadecimal digit converts by itself to 4 bits.

Examples :

$$(10223)_4 = (0100101011)_2$$

$$(6530.7)_8 = (110101011000.111)_2$$

$$(101,010100111,000)_2 = (52470)_8$$

$$(1101100010110110.011)_2 = (D8B6.6)_{16}$$

$$(1110010010.101)_2 = (3210.22)_4$$

3-6 THE signed numbers

So far we have talked about natural numbers. They cannot by nature whether to be positive or zero. Let us now consider relative integers or in other words, equipped of a sign '+' Or '-'. In decimal,

+1, +2, +3 etc. are of the numbers positive. They are superiors has 0 (not >0)

-1, -2, -3 etc. are negative numbers. They are less than 0 ($n < 0$)Of

even in binary,

+1, +10, +11, +100, +101 etc. are of the numbers binary positive,

-1, -10, -11, -100, -101 etc. are of the binary numbers negative.

The problem is that digital electronic circuits can only record 0s or 1 but no + or - signs. The only way is then to agree that if a number is likely to be negative we him to book A bit For indicate the sign.

Several methods are used to represent negative numbers inA computer, among which We let's quote : there representation in sign and value absolute (**SVA**), 1's complement (**CP1**) and 2's complement (**CP2**) . We are looking for a binary representation of negative integers so that the addition of two relative integers works, It is the representation complement has 2 (CP2).

How calculate THE codes of the numbers negative in CP2 on 8 bits:

THE calculation se do in three steps :

1st calculation of coded binary of the number out of 8 bits

2° Calculation of the complement to 1 = Replace all 0s with 1s and all 1s with 0s.3°

Calculation of complement has 2 = Add 1 At complement to 1

Examples : 1-how write – 4 in CP2 ?

$$+ 4 = 0000\ 0100_{(2)}$$

The complement to 1 of this code is CP1(+4)=1111 1011

(2)Let's add 1 to this code to get its 2's complement 1111

$$1011_{(2)} + 1 = \mathbf{1,111} _ 1100_{(2)} = (-4)_{CP2}$$

2-how to write – 24 in CP2?

THE number $+24 = (11000)_2 = (0001\ 1000)_{CP2}$

The complement to 1 of this code is $CP1(+24) = 1110$

0111 Let's add 1 to this code to get its 2's complement

$1110\ 0111_{(2)} + 1 = 1,110\ 1000_{(2)} = (-24)_{CP2}$

Thus $(-24)_{10} = (1110\ 1000)_{CP2}$

THE bit THE more has LEFT of coded CP2 East the one Who go represent the sign. Sign negative if Thisbit worth 1, positive sign When This bit worth 0.

THE bigger number sign on 8 bits East $+127 (01111111)$

THE smaller number sign on 8 bits is $-128 (10000000)$

-128 has +127 => 256 combinations (2 power 8)

This Who applies on 8 bits applies Also on 4, 16, ... bits.

Notifications : The addition binary :

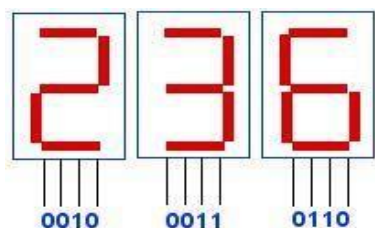
The bill of two binary bits se accomplished according to there specification next :

| bit1 | bit2 | result | detenti on |
|------|------|--------|---------------|
| 0 | + | 0 | 0 |
| 0 | + | 1 | 0 |
| 1 | + | 1 | 0 |
| 1 | + | 0 | 1 |

The bill binary of two numbers is carried out bit has bit of RIGHT has left, in reporting the restraints

THE decimal binary encoded (DCB code)

This coding East destined has the display of values decimal places, each digit must be coded inbinary on 4 bits (units, tens, hundreds...).



This coding does not allow any calculation, it is only intended for the entry and display of data

THE coding ASCII

THE binary allow of code THE numbers that THE systems IT canmanipulate. However, the computer must also use alphanumeric characters to memorize and transmit texts. To encode these characters, we associate with each of them them A coded binary, It is THE coding ASCII (American Standard Coded for Information Interchange).

4-Principle operation of a computer

When the computer starts up, it first launches special software that is calls the OS (for example Windows, Linux or MacOS, etc... Without this OS, you cannot not to use the computer And the computer born would be than a machine Who born would serve not has bigthing. It is the bone Who go interpret your orders And YOU display THE results.

Modern operating systems generally display a home screen (called desktop) containing all your icons. These icons are going allow of throw THE software in clicking above.

Here is how That walk When YOU launch A software stage by step.

1. YOU click (Or double click) on the icon.
2. The computer will SO search in THE disk hard towards what software point this icon.
3. He go read THE software has leave of disk hard because this one East checked in above.
4. All This Who will be read will be charge in the memory long live (RAM).
5. HAS leave of there RAM, THE processor go execute THE software instruction After instruction.
6. A times that YOU will close THE program, this one will be erased of there RAM.
7. But GOOD obviously, he stay here on THE disk hard For a next use.

There RAM, THE processor And THE disk hard are of the components of the unit central ofthe computer.

5-Hardware part of a computer

We can cite among THE components of a computer :

- There map mother
- THE processor
- There memory long live
- THE disk hard
- The reader and/or burner of CD-ROM and or DVD-Rom

motherboard

Main board on which we find all the components necessary for operation of a computer, including the processor, central memory, buses and connectors extension intended has receive of the cards extension And manages THE different interfaces with USB ports for peripherals external.

THE processor

Often qualified of brain of the computer, the processor allow of manipulate And of to treat the data which are provided. Her power has an influence on the speed of execution of your software and operations that YOU perform.

There memory long live

Partner of work of processor, there memory long live (called Also **RAM**) temporarily stores data to be processed by the processor. Thus, the more there are more memory available, the more data can be stored temporarily (This Who avoid access At Hard disk who is more slow).

There memory long live East emptied has each stop Or reboot of your computer.

THE disk hard

Unlike RAM, data stored on a hard drive is permanent and are not erased when you shut down your computer. It is therefore on a disk hard that your system operating (Linux, MacOs Or Windows), your software And your documents are preserved. It is SO A space of storage permanent, where you can to copy and delete of the data has will.

THE drive and or engraver of CD-ROM and or DVD-Rom

Definition drive :

Device allowing there reading, has ugly of a Ray laser, of the information recorded on A CD-ROM or one DVD-Rom

THE readers of CD-ROM And DVD-Rom can be internal (that's to say integratedIn computer) or external (appearing as there box shape autonomous).

Definition engraver :

Device allowing the recording of compact discs, individually or in smallseries, by engraving At laser on A CD or DVD

There are other components such as the graphics card for display on the screen, there map her For manage THE entries (microphones) And THE exits (Tops speakers..), the network card which allows the transfer of data between a computer personal and a network or between a server and a network. Also allows access to Internet In A network domestic Or business.

6-Part system

We have present the computer as A together of circuits interconnected electronics. For establish the communication between these elements materials and the man, considered as the user, he East necessary having a interface appropriate which makes human-machine communication possible, hence the concept of system operating.

THE system operating (*SE* , in English *Operating System* Or *OS*) East A set of programs responsible for linking hardware resources of a computer and THE apps of the user (treatment of text, game video, ...).

There are dozens of different operating systems on the market, often delivered with the computing device. This is the case for Windows , Mac OS , Irix ,Symbian BONE, Linux (For which he exist of many distributions) Or Android .