

Chapitre 1

Architecture des ordinateurs

1.0.1 Ordinateur

Un ordinateur est un appareil électronique qui fonctionne à partir d'une alimentation électrique. Ce peut être ce peut être le secteur ou une batterie.

Avec un composant électronique il est facile de représenter deux états : chargé/déchargé, circuit ouvert/fermé, qu'on peut assimiler à deux valeurs, 0 et 1.

Aussi dans un ordinateur, toutes données, quelle qu'elle soit est représentée par une suite de 0 et de 1 qu'on appelle des **bits** (en anglais, contraction de **BI**nary **di**gi**TS**, ou **chiffres binaires**).

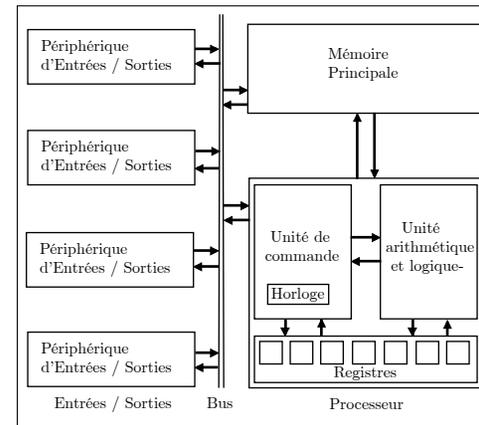
Dans un ordinateur toutes les données sont stockées sous forme d'une suite finie de **bits**. Chaque bit peut prendre 2 valeurs, 0 ou 1.

1.1 Architecture de Von Neumann

1.1.1 Architecture de Von Neumann

Tous les ordinateurs depuis 1948, sont modélés sur le même principe de fonctionnement, qui bien qu'il ait été complexifié depuis, repose sur le schéma proposé par John Von Neumann : l'architecture de Von Neumann :

Tous les ordinateurs sont conçus sur la même architecture, dite *architecture de Von Neumann*, qui a été proposée par le mathématicien John Von Neumann en 1948.



Dans cette architecture l'ordinateur est composé du **processeur**, de la **mémoire principale**, des **périphériques d'entrées/sorties**, et du **bus** qui permet leur communication.

Dans l'architecture de Von Neumann :

- La **mémoire principale** contient les données et les programmes. Le processeur y accède directement pour l'exécution de tous les programmes.
- Le **processeur** (ou micro-processeur) contient :
 - une **horloge**, qui cadence toutes les opérations,
 - l'**unité de commande** qui séquence l'exécution des instructions contenues dans les programmes en mémoire,

- l'**unité arithmétique et logique** qui effectue tous les calculs et les opérations logiques, et
- les **registres**, unités de mémoires d'accès ultra rapide.

Le processeur communique avec les autres composants par le biais du bus.

1.1.2 Mémoire principale

- La **mémoire principale**, contient toutes les données et les programmes au cours de leur exécution. Elle est constituée de :
 - la mémoire dite "morte", perpétuelle, qui ne peut être que lue, et qui contient les données et programmes nécessaires au démarrage de l'ordinateur. C'est la **ROM** (Read-Only Memory).
 - La **mémoire vive**, ou **RAM** (Random Access Memory), volatile, qui se charge à l'exécution des programmes et au chargement de données.

Le processeur a un accès direct à la mémoire principale. On parle de mémoire à accès direct.

1.1.3 Le bus

- Le **bus**, permet d'échanger les données entre les différentes unités. Il est constitué :
 - d'un **bus d'adresse**, qui spécifie où se situe la donnée.
 - d'un **bus de donnée**, qui permet le transfert, lecture, ou écriture de la donnée à son adresse.
 - L'**unité d'entrée / sortie** permet la connection des différents **périphériques d'entrées / sorties**. Chaque périphérique d'entrée/sortie est connecté sur un port. L'unité

d'entrée/sortie fait le pont entre le bus et les différents périphériques.

1.2 La mémoire

1.2.1 La mémoire - unité de stockage

Toutes les données en mémoire sont stockées sous forme de suites de 0 et de 1, des *bits*. Afin d'optimiser les temps d'accès, l'ordinateur ne peut pas accéder à des bits isolés, mais seulement à des regroupement de plusieurs bits, traditionnellement 8 bits. On les appelle des *octets*.

Un *octet* est un groupement contigu de 8 bits. On anglais un octet se dit *byte*.

Au fur et à mesure de l'augmentation des performances des ordinateurs, l'unité de stockage a évolué de 8 bits, à 16 bits (années 80), 32 bits (années 90), 64 bits (années 2010). C'est l'**unité de stockage**. Cela signifie que dans votre ordinateur 64bits, tous les composants de l'ordinateur, ne stockent et ne transfèrent que des groupement de 64 bits, soit 8 octets. Par exemple le bus d'adresse sera constitué de 64 fils électriques. Cela permet de représenter $2^{64} \approx 18 \times 10^{18}$ adresses, soit près de 20 milliards de milliards.

Dans un ordinateur 64 bits, qui est la norme actuelle des ordinateurs personnels, les données sont regroupées par unités de stockage de 64 bits, soit 8 octets.

1.2.2 La mémoire : les registres

Toutes les données seront stockées dans différents composants de mémoire. Elles peuvent être lues, et écrites, par le processeur, mais aussi parfois par des périphériques d'entrée/sortie. Ces composants sont de différents types :

- Les **registres**, situés au sein du microprocesseur. Volatiles (leur contenu est perdu à l'arrêt de l'ordinateur), ils peuvent être accédés en lecture et en écriture. En nombre restreint (de plusieurs dizaines à plusieurs centaines d'unités de stockage), ils ne sont utilisés pour les stockages temporaires durant les calculs et opérations du processeur. Leur temps d'accès est le plus rapide, de l'ordre d'une nano-seconde (1 milliardième de seconde) par unité de stockage.

Lorsqu'on effectue un addition, le processeur stocke les deux opérands dans deux registres, effectue le calcul au sein de l'unité arithmético-logique, qui stocke le résultat dans une registre privilégié, l'accumulateur. Le processeur peut ensuite communiquer ce résultat sur le périphérique de sortie souhaité, ou l'enregistrer dans la mémoire principale, selon les instructions de l'utilisateur, ou plutôt du programme utilisé.

1.2.3 La mémoire : la mémoire vive

- La **mémoire vive** (RAM) est située dans la mémoire principale. Volatile, accessible en lecture et en écriture, c'est le lieu d'exécution de tous les programmes. Elle contiendra à la fois les programmes, lors de leur exécution, et les données qu'ils manipuleront. Le processeur y accède directement. Le temps d'accès y est très rapide, de l'ordre d'une dizaine de nano-secondes par unité de stockage. Sa quantité est en général limitée à quelques giga-octets (milliards d'octets).

À l'exécution d'un programme, ce dernier se charge d'abord dans la mémoire vive sous forme d'instructions en langage machine, uniquement comprises par l'unité de commande. C'est l'unité de commande au sein du processeur qui séquence son exécution, en exécutant la suite de ses instructions. Les données qu'il manipule seront aussi stockées

dans la mémoire vive. A l'arrêt de l'ordinateur toutes les données de la mémoire vive seront perdues.

1.2.4 La mémoire : la mémoire morte

- La **mémoire morte** (ROM) située dans la mémoire principale et d'accès direct rapide. Perpétuelle (non volatile), elle n'est accessible qu'en lecture. Elle sert à stocker le programme qui se charge au démarrage de l'ordinateur (le BIOS sous PC), et qui sert à vérifier la connexion des périphériques, à charger leurs pilotes, et à charger dans la RAM le programme d'exécution du système d'exploitation (shell).

De nos jours il en fait possible d'écrire dans la ROM, mais seulement pour des mises à jours réservées aux spécialistes et au système d'exploitation. C'est la ROM EPROM.

1.2.5 La mémoire : la mémoire de stockage

- Le **disque dur**. C'est un périphérique d'entrée et sortie destiné au stockage à long terme du système d'exploitation et des programmes et données. Non volatile, son contenu demeure à l'extinction de l'ordinateur, il est accessible en lecture et en écriture. Le transfert des données est lent, de l'ordre de quelques milli-secondes par unité de stockage (1000 fois plus lent que la mémoire principale).

Au démarrage, et à l'ouverture des programme, l'ordinateur charge les données utilisées du disque dur vers la RAM, pour l'exécution ou le traitement des données. On distingue de nos jours essentiellement deux types de disque dur, HDD fragile et lent équipant la plupart des ordinateurs, SSD robustes et plus rapides, mais aussi plus chers, ils occupent les ordinateurs plus hauts de gamme.

- Les **périphériques de stockage externes**. Mémoires non-volatiles ; on peut citer les clés USB, cartes mémoires, disques durs externes, stockage sur réseau (cloud), disques CD, DVD, Blue-ray gravables, accessibles en lecture et écriture, ou les CD, DVD, Blu-ray non gravables, accessibles uniquement en lecture.

1.3 Le processeur

C'est le coeur de l'ordinateur. Il est constitué de :

- **L'horloge**. Elle détermine la rapidité du processeur, et par voie de conséquence de l'ordinateur. Sa rapidité est donnée en Hz, MHz, GHz. Ainsi un processeur cadencé à 3 GHz effectue 3 milliards d'opérations à la seconde. Toutes les opérations de l'ordinateur sont cadencés par l'horloge.
- **L'unité de commande** ; elle exécute les instructions des programmes. Ces instructions sont lues séquentiellement dans la mémoire principale ; elles sont codées en langage machine. Les instructions du langage machine dépendent du microprocesseur et sont en nombre très restreints : elles consistent en quelques opérations élémentaires ; lire le contenu d'une unité de stockage située à une certaine adresse (dans la mémoire principale ou via le bus), écrire une donnée à une certaine adresse (idem), commander un calcul ou une opération logique à l'unité arithmétique et logique, passer à l'instruction suivante en mémoire, ou sauter à une instruction situées à une adresse donnée en mémoire principale à condition que la valeur d'un registre donné soit non-nul. Une opération est exécutée à chaque temps d'horloge.

Tout ce qui s'exécute dans l'ordinateur, est commandé par l'unité de commande. C'est le donneur d'ordre. Tout pro-

gramme, aussi complexe soit-il, s'obtient par l'exécution de ces quelques opérations élémentaires.

- **L'unité arithmétique et logique**. Commandée par l'unité de commande, son rôle est d'exécuter tous les calculs élémentaires, addition, soustraction, multiplication, division, et surtout toutes les opérations logiques, et, ou, non, xor, décalage, dont toutes les opérations mathématiques sont déduites. Les opérandes et résultats sont stockés dans les registres. L'unité arithmétique et logique dispose aussi d'un accès direct à la mémoire principale, et d'une connexion avec le bus.

Un ordinateur "multi-core" dispose de plusieurs processeurs qui peuvent travailler simultanément.

Une carte graphique est un périphérique de sortie qui de nos jours contient son propre processeur et sa propre mémoire principale (pour fluidifier l'affichage d'images et d'animations).

1.4 Périphériques d'entrées-sorties

Ils sont indispensables pour communiquer avec l'ordinateur. Ils étaient initialement réduits à des entrées et sorties sous forme de cartes perforées. Les périphériques d'entrée-sorties peuvent être, par exemple :

- Périphériques d'entrée : clavier, souris, trackpad, joystick, scanner, webcam, écran tactile...
- Périphériques de sortie : écran, vidéo-projecteur, haut-parleurs, imprimante, bras articulé, etc...
- Périphériques de stockage ; ce sont des périphériques d'entrées et sorties : disques externes, lecteur/graveur CD, DVD, Blu-ray, clé usb, etc...
- Périphériques de communication ; ce sont des périphériques d'entrées et de sorties : carte ethernet, carte wifi, bluetooth, clés 3G, etc...

1.5 Langage de programmation

1.5.1 Le langage machine

Un processeur ne comprend qu'un seul langage, le langage machine. C'est un langage séquentiel constitué d'une jeu d'instruction très réduit permettant d'accéder, lire et modifier les données stockées dans la mémoire principale ou via le bus dans les périphériques, d'effectuer des calculs et opération logiques basiques dans l'unité arithmétique et logique, et de poursuivre l'exécution du programme en sautant à une instruction située à une certaines adresses. C'est un langage de bas niveau. Voilà à quoi peuvent ressembler quelques instructions en langage machine sur un ordinateur 8 bits :

Adresse	Contenu (binaire)	Contenu (hexadécimal)
00000000	01010101	55 1ère instruction
00000001	11111111	FF 2ème instruction
00000010	00000000	00
00000011	11001000	C8
00000100	00110001	31 3ème instruction

Y figurent, l'adresse en mémoire, l'instruction sur un ou plusieurs octets. On a écrit le contenu en binaire, mais aussi en hexadécimal (base 16) pour faciliter sa lecture.

1.5.2 L'assembleur

L'assembleur a été le premier langage informatique lisible par un humain. Ce n'est qu'une retranscription du langage machine utilisant certaines commandes pour faciliter sa lecture et son écriture : LD (pour Load, charger un registre avec une donnée), MOV (pour Move, déplacer une donnée entre deux adresses ou registres), ADD (pour ajouter au contenu d'un registre le contenu situé à une adresse), etc... La conversion langage machine/assembleur est très simple, c'est une traduction mot à mot, et devra être exécuté par un programme appelé assembleur. Voici

un exemple d'instructions en assembleur et leur conversion en langage machine sur un ordinateur 16 bits ; les adresses sont données en hexadécimal :

Adresse	Language Machine	Assembleur	Description
0000	A1 0A FF	MOV AX, [0AFF]	charger le registre AX avec la donnée à l'adresse 0AFF
0003	03 06 0B 02	ADD AX, [0B02]	ajouter le contenu de 0B02 à AX
0007	A3 0B 04	MOV [0B04], AX	ranger le résultat en 0B04

L'assembleur est un langage de bas niveau : son jeu d'instruction ne consiste qu'en des opérations élémentaires du microprocesseur.

1.5.3 Les langages évolués

Pour faciliter l'écriture de programmes, des langages de plus haut niveau ont été conçus : citons Fortran, Cobol, Langage C, Basic, Pascal, etc... ils permettent l'écriture de programme avec un jeu d'instructions bien plus étendu, permettant par exemple l'écriture de texte à l'écran, le tracé de graphiques, l'ouverture/lecture d'un fichier sur le disque, ou des instructions permettant de répéter des jeux d'instructions (boucles). Ce sont en général des langages dits structurés, par opposition aux langages séquentiels que constituent le langage machine et l'assembleur.

L'écriture d'un programme dans un langage évolué est considérablement plus simple (car l'utilisateur n'a pas à connaître les adresses des différents périphériques), et concis (car le jeu d'instruction est considérablement plus large et réalise des opérations plus complexes). De nos jours, à part pour la programmation de pilotes de périphériques ou des systèmes d'exploitation, la programmation ne se fait plus que dans des langages évolués.

1.5.4 Langages compilés/interprétés

Un programme écrit dans un langage évolué doit être ensuite compilé par un programme appelé compilateur qui le traduit en langage machine. C'est ce code traduit qui constitue le programme exécutable qui sera séquencé par l'unité de commande (les choses sont en fait un rien plus complexes).

Depuis l'arrivée de l'internet, sont apparus des langages qui ne sont plus compilés, mais interprétés. Dans ce cas les instructions du programme sont traduites en langage machine et exécutés simultanément. L'exécution des programmes en est plus lente car l'interpréteur doit gérer la mémoire au fur et à mesure de l'exécution, et ne peut pas optimiser le code, mais cela permet d'exécuter un programme sous forme de scripts situés sur des sites web distants. Citons notamment PHP, Java script, et `python` qui est le langage de programmation évolué que nous utiliserons.

1.5.5 Installation de python

On installera une distribution `python`, avec un EDI (Environnement de développement intégré). Plusieurs versions du langage sont disponibles, il faudra prendre soin de télécharger la dernière version `python 3` (à ce jour la dernière version est la 3.5.1 ; n'importe quelle version 3.x.x fera l'affaire). Attention à ne pas télécharger une version antérieure, `python 2`, dont le jeu d'instructions varie significativement.

Plusieurs possibilités sont offertes, toutes gratuites, et incluant le langage `python` sous sa dernière version, notamment :

- La distribution Pyzo : Python to the people. Simple, fonctionnelle, portable (peut être située sur une clé usb et utilisé sans installation préalable), elle est disponible à l'adresse :

<http://www.pyzo.org>

Elle utilise pour E.D.I. le logiciel IEP.

Nous encourageons son usage.