

# Architecture :

## Introduction au modèle de Von Neumann

### Objectif :

- **Savoir si un composant informatique est un périphérique d'entrée, de sortie ou fait partie du traitement de l'information.**
- **Différencier les types de mémoires par leurs rôles et leurs caractéristiques techniques**
- **Comprendre le rôle de l'ALU**
- **Comprendre que la mémoire contient les données et les instructions**

## I Fonctionnement général d'un ordinateur

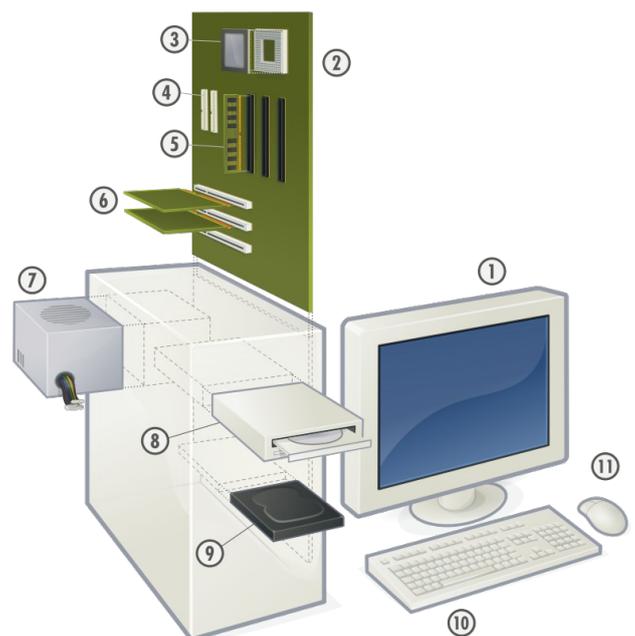
Les ordinateurs ont été conçus à la base pour traiter automatiquement des données (INFORMATION autoMATIQUE). Leur organisation matérielle se décompose en trois parties :

Les périphériques d'entrée // Les périphériques de sortie // L'unité de traitement de l'information.

On retrouve cette organisation matérielle dans quasiment tous ce qui contient un processeur.

Dans un ordinateur moderne, des périphériques d'entrée sont par exemple le clavier, la souris et le disque dur, des périphériques de sortie sont par exemple l'écran, le graveur de DVD et le disque dur.

- 1 Écran
- 2 Carte mère
- 3 Processeur
- 4 Parallèle ATA
- 5 Mémoire vive (RAM)
- 6 Connecteurs d'extensions : Carte Graphique, Carte Son, Carte réseau, etc.
- 7 Alimentation électrique
- 8 Lecteur de disque optique
- 9 Disque dur, disque électronique
- 10 Clavier
- 11 Souris



## II La mémoire

Dans un ordinateur la mémoire stocke des variables, des données (textes, vidéos, etc.) et des programmes. Il y a différents types de mémoire mais dans tous les cas on peut la voir comme une suite d'emplacements qui peuvent stocker une quantité fixée de bits par exemple 64 bits. Chaque emplacement est associée à une unique adresse qui est un nombre. Pour lire le contenu d'un emplacement il suffit donc de connaître son adresse. Si on veut écrire à cet emplacement, il faut l'adresse et la donnée à écrire. Pour des données volumineuses, il faut plusieurs emplacements pour les stocker.

Il y a deux grands types de mémoire :

- **La mémoire centrale** (ou interne).  
C'est la mémoire qui sert à stocker *temporairement* les données et programmes en court d'utilisation. C'est ce qu'on appelle **la mémoire vive**
- **La mémoire de masse** (on externe ou encore physique)  
C'est la mémoire qui sert à stocker les informations à long terme et qui sont conservées lorsque l'ordinateur est éteint. C'est par exemple le cas des disques durs, des supports comme les DVD-ROM ou les CD-ROM.

### Caractéristiques techniques principales :

- **Capacité** : représente la quantité d'information que peut stocker une mémoire exprimée en nombre de bits.
- **Temps d'accès** : L'intervalle de temps entre la demande d'information ou l'accès en écriture et la disponibilité de la donnée.
- **Débit** : Le volume d'information échangé par unité de temps.
- **Non volatilité** : la mémoire est dite non volatile si l'information est conservée lorsque qu'elle n'est plus alimentée électriquement.

Voici une liste de différentes mémoires d'un ordinateur :

Mémoire	Temps d'accès	Débit	Capacité
Registres	1 ns		Quelques Kio
Mémoire cache	2-3 ns		Quelques Mio
Mémoire vive	5-60ns	1-20 Gio/s	Quelques Gio
Disques durs	3-20 ms	10- 320 Mio/s	Quelques Tio
DVD	140 ms	2-22 Mio/s	4,6-17 Gio

Les mémoires les plus rapides et de capacité plus faibles sont situées à proximité du processeur et les mémoires de masse, moins rapide, servent à stocker les informations de manière permanente.

**La RAM (Random Access Memory mémoire à accès direct) :** C'est ce qu'on appelle aussi la *mémoire vive*. C'est une mémoire volatile qui sert à stocker de façon temporaire les données et les programmes lors de l'exécution de ses derniers.

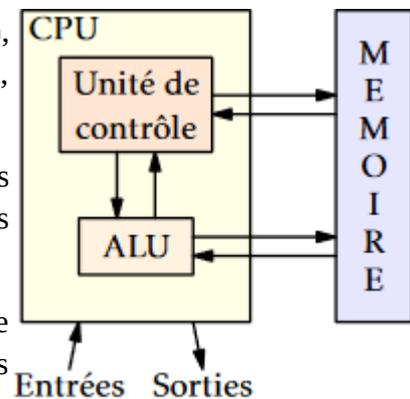
**La ROM (Read Only Memory mémoire en lecture seule) :** On l'appelle aussi *mémoire morte*. C'est une mémoire non volatile sur laquelle on ne peut écrire de données. C'est ce type de mémoire qui sert au démarrage d'un ordinateur, d'un smartphone mais aussi qui contient les microprogrammes d'un baladeur, du programmeur d'une machine à laver.

### III Architecture de Von Neumann

Les grands principes de fonctionnement des ordinateurs actuels reposent toujours sur les travaux effectués dans les années 40 par une équipe de chercheurs de l'université de Pennsylvanie dont Von Neumann. La grande idée a été de stocker au même endroit les données et les programmes qui devaient les manipuler. On appelle cette architecture le **modèle de Von Neumann** en hommage au mathématicien John Von Neumann qui publia en 1945 un rapport sur la conception de l'EDVAC, un nouvel ordinateur conçu sur ce modèle.

L'architecture de Von Neumann décompose un ordinateur en quatre parties distinctes :

- **L'unité arithmétique et logique (UAL ou ALU en anglais)**, son rôle est d'effectuer les traitements de base (addition, soustraction, ou, non, comparaison, etc).
- **L'unité de contrôle** qui s'occupe de « séquencer » les opérations à effectuer. Elle s'occupe d'effectuer les instructions dans l'ordre attendu.
- **La mémoire** qui contient à la fois les données et le programme qui indique à l'unité de contrôle quelles opérations faire sur les données et dans quel ordre. Les échanges entre le CPU et la mémoire se fait par l'intermédiaire d'un médium appelé **bus**. Il y a aussi des **registres** qui sont des mémoires très rapides et indépendantes de la mémoire vive par exemple l'*accumulateur* qui stocke les résultats intermédiaires et le compteur ordinal qui contient l'adresse de la dernière instruction exécutée (ou celle à venir selon les architectures).
- **Les entrées et sorties** qui permettent de communiquer avec l'extérieur.



Le processeur ou CPU (Central Processing Unit en anglais) contient donc l'unité de contrôle, l'ALU et des registres.

Une particularité de l'architecture de Von Neumann, est que les instructions des programmes sont stockées en mémoire avec les données. Initialement, pour pouvoir faire des boucles il était indispensable de pouvoir modifier des instructions pendant l'exécution d'un programme. Depuis on procède autrement et c'est même une bonne pratique, mais cela reste possible. C'est ce que tente de faire une partie des virus informatique. Ils se présentent sous la forme de données qui quand elles sont lues tentent de modifier le programme et par exemple de lui faire exécuter des instructions qui sont inscrites dans ces données.

Bien que ce modèle reste le modèle de référence, il y a eu depuis quelques évolutions.

- Les processeurs sont maintenant souvent multicœurs ce qui permet de faire des calculs en parallèles sans augmenter la fréquence des processeurs ce qui limite les problèmes d'évacuation de la chaleur (plus la fréquence est rapide plus la quantité de chaleur dégagée est importante). Optimiser le parallélisme reste un problème difficile car souvent il est très délicat de séparer des opérations dont les résultats sont dépendants les uns des autres.
- Les entrées-sorties sont gérées par des processeurs indépendants ce qui associé à la multiprogrammation (partage de la mémoire en plusieurs programmes) permet le développement de systèmes en temps partagés (multi-tâches et multi-utilisateurs).

## IV Jeu d'instruction d'un processeur

Chaque processeur est capable d'effectuer un certain nombre d'instructions qui dépendent des circuits qu'il contient. L'ensemble des instructions s'appelle le *jeu d'instructions*.

Chaque instruction est une suite de bits, comme le sont les données, compréhensible par le CPU. Ces instructions forment **le langage machine**. Les instructions sont lues les unes après les autres à un rythme imposé par une horloge. Le *cycle d'exécution d'une instruction* est composé de trois étapes :

- **Chargement** : À l'adresse mémoire indiquée par le registre PC, l'unité de contrôle va récupérer le mot binaire qui contient la prochaine instruction à coder et le stocker dans son registre IR
- **Décodage** : La suite de bits est décodée ce qui permet de savoir quelle instruction exécuter et quelles données sont éventuellement nécessaires. Si des données sont nécessaires, on dit aussi *opérandes*, elles sont chargées en mémoire.
- **Exécution** : L'instruction est exécutée soit dans l'ALU (s'il s'agit d'une opération arithmétique ou logique) soit dans l'unité de contrôle s'il s'agit d'un branchement qui va modifier le contenu du registre PC.

Une instruction machine se présente sous la forme suivante :

Champ code opération	Champ opérandes
----------------------	-----------------

La programmation en binaire étant loin d'être évidente pour un humain, on a inventé le langage **assembleur** qui permet d'écrire les instructions de manière plus compréhensible.

Prenons l'exemple de l'assembleur lié au simulateur de processeur disponible ici :

<https://csreith.com/cpu.php>

ADD	R2,R1,R0
-----	----------

Permet d'ajouter le contenu du registre R0 à celui du registre R1 et de stocker le résultat dans le registre R2.

CMP	R1,R0
-----	-------

Compare le contenu des registres R0 et R1 et renvoi un statut

BGT	label
-----	-------

Si le code statut indique que R1 est plus grand que R0 alors l'adresse suivante est celle du label.

Sinon l'instruction suivante est lue.

LDR	R0, 8
-----	-------

Met le contenu de l'adresse 8 dans le registre R0

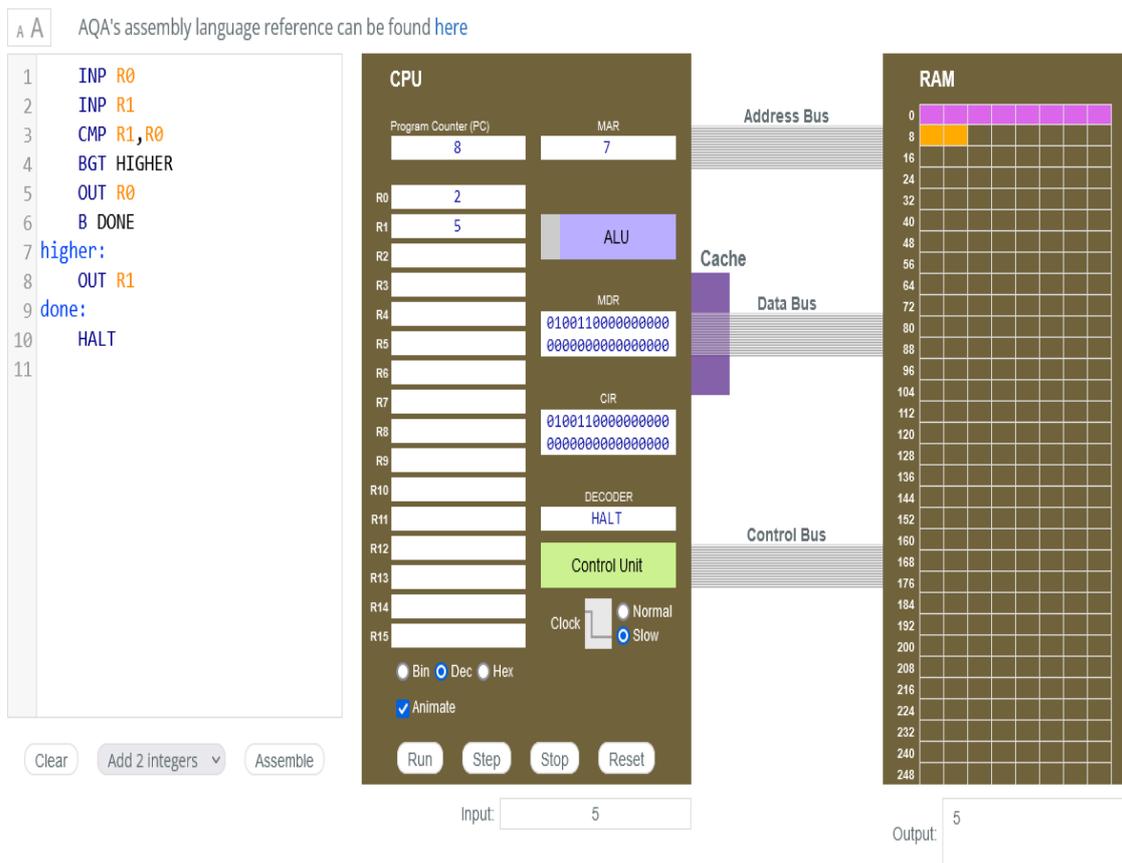
Des instructions ont été ajoutées pour les entrées-sorties.

INP	R0
-----	----

Copie le nombre entré par l'utilisateur dans le registre R0

OUT	R2
-----	----

Affiche le contenu du registre R2

**Exemple :**

Pour voir ce modèle simplifié de cette architecture en action :

<https://csreith.com/cpu.php>

**Sources :**

<https://fr.wikipedia.org/wiki/Ordinateur>

<http://nymphomath.ch/info/comment/comment.pdf>

<https://interstices.info/le-modele-darchitecture-de-von-neumann/>