

ARCHITECTURE D'UN ORDINATEUR

1) Architecture de « Von Neumann »

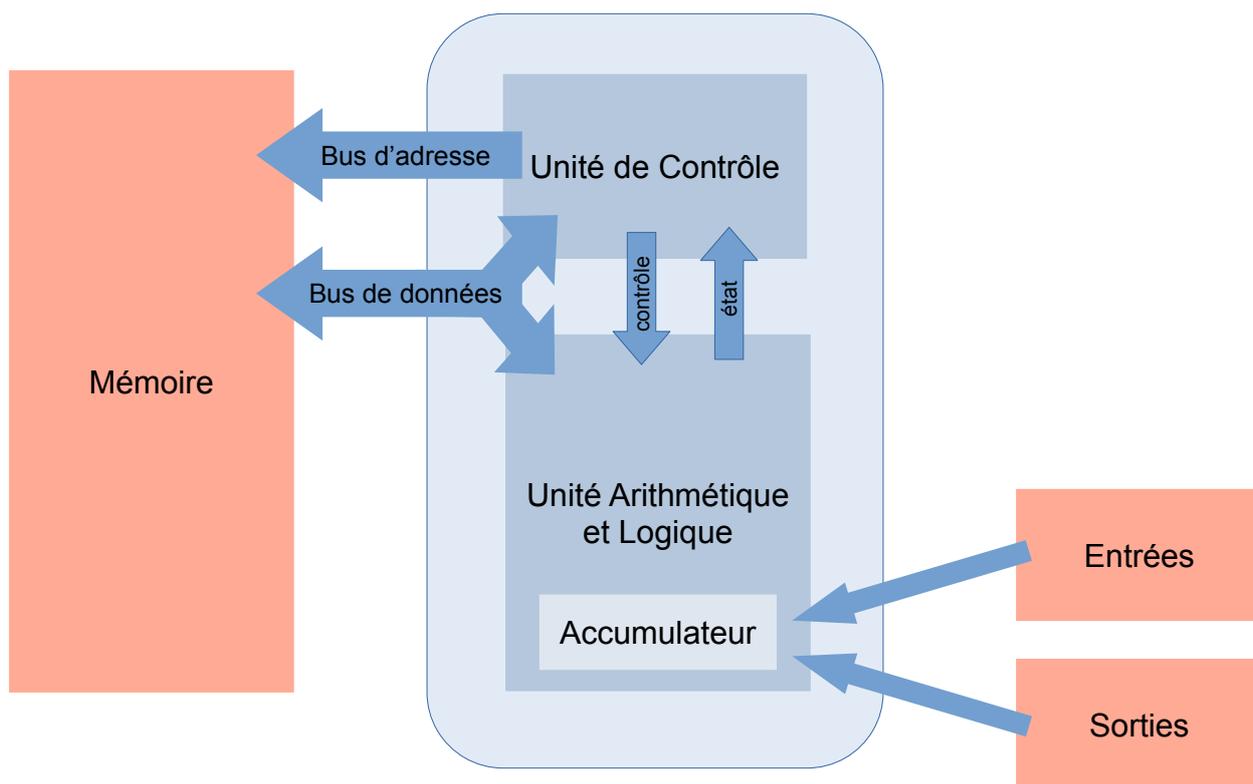
Dans les années 1940, apparaissent les premiers calculateurs électroniques. Ces calculateurs n'ont ni clavier, ni souris, ni écran et il n'y a pas encore de véritables langages de programmation.

Pour exécuter un nouveau programme, il faut manipuler de très nombreux interrupteurs ou modifier le branchement d'un grand nombre de câbles et cela prend plusieurs heures à toute une équipe.

En 1945, des ingénieurs travaillant sur un nouveau projet d'ordinateur appelé EDVAC publient une note proposant deux innovations majeures qui feront date dans l'histoire de l'informatique et qui seront reprises dans la plupart des projets qui suivront :

- Le programme est désormais enregistré dans la mémoire et il n'y aura donc plus besoin de modifier le câblage de la machine. La mémoire contient donc à la fois le programme et les données.
- À l'**Unité Arithmétique et Logique (UAL)** qui fait les calculs, on ajoute une **Unité de Contrôle (UC)**, qui lit les instructions du programme en mémoire et pilote l'UAL.

C'est cette nouvelle architecture que l'on appelle aujourd'hui encore « architecture de Von Neumann ».



En savoir plus sur l'aspect historique :

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

II) Langage machine et assembleur.

Un processeur exécute des instructions qui sont des suites de 0 et de 1.

Exemple, avec le [simulateur AQA](#) pour faire l'addition : Registre1+Registre2 vers Registre3

Instruction langage machine	1110 0000 1000	0001	0011	0000 0000	0010
	Addition de registre à registre	N° registre opérande 1	N° registre résultat	Inutilisé	N° registre opérande 1
Assembleur	add R3, R1, R2				

Les instructions acceptées par un processeur sont en nombre en général assez réduit et elles sont directement câblées dans les différents circuits électroniques du processeur.

On y trouve :

- Des instructions de transfert de données entre la mémoire et les registres
- Des opérations arithmétiques (les 4 opérations) ou logique (opérateurs booléens)
- Des instructions de contrôle (branchement, branchement conditionnel,...)

III) L'Unité de Contrôle

L'unité de contrôle est le chef d'orchestre du processeur. Elle exécute en permanence une boucle sans fin :

- 1) **Fetch** : L'unité de contrôle récupère l'adresse mémoire de la prochaine instruction à exécuter. Cette adresse est stockée dans le registre PC (Programme Counter). Puis elle lit en mémoire la valeur de cette instruction et la stocke dans le registre CIR (Current Instruction Register). Ensuite, elle incrémente le registre PC.
- 2) **Decode** : L'unité de contrôle identifie de quel type d'instruction il s'agit et récupère les données sur lesquelles cette instruction doit opérer (registre, valeurs en mémoire...)
- 3) **Execute** : L'unité de contrôle exécute l'instruction. Par exemple, s'il s'agit d'une multiplication, elle la fait calculer par l'UAL, s'il s'agit d'un branchement, elle met à jour la valeur de PC...

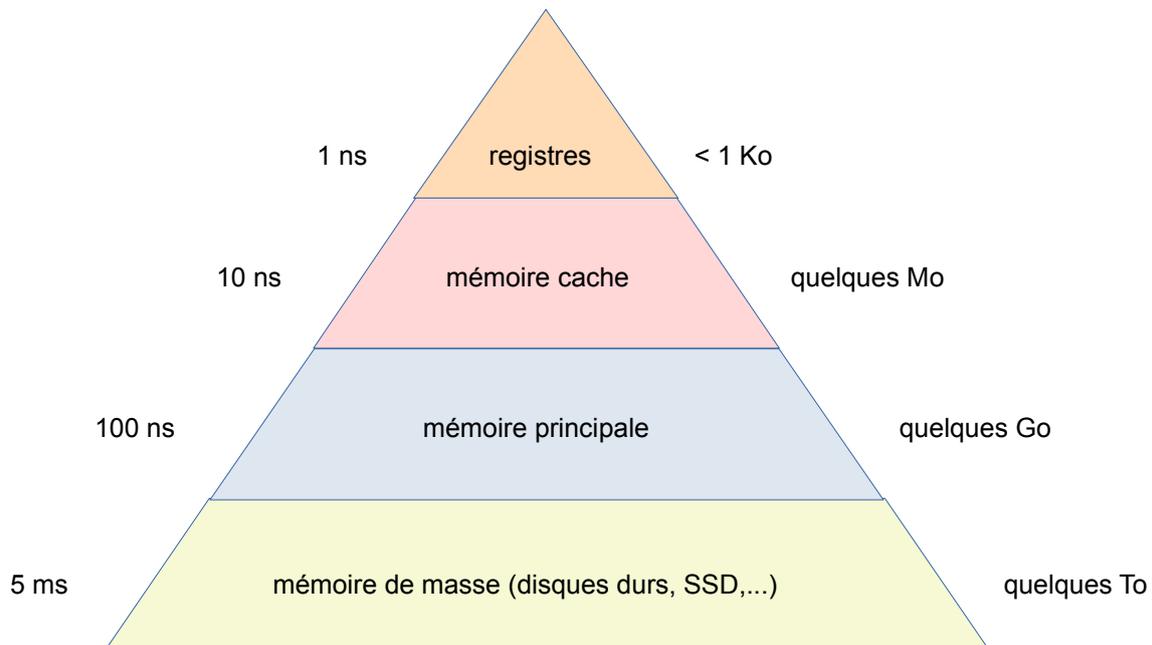
IV) L'Unité Arithmétique et Logique

L'Unité Arithmétique et Logique s'occupe des calculs et contient des circuits spécifiques pour :

- Exécuter les quatre opérations sur les entiers
- Exécuter les quatre opérations sur les nombres en virgule flottante
- Faire des comparaisons
- Exécuter les différentes fonctions booléennes (et, ou, non, ou exclusif...)
- Faire des opérations sur les bits (décalage à droite ou à gauche...)

V) La mémoire

La mémoire principale est beaucoup plus lente que le processeur. Pour éviter que ce dernier ne soit trop ralenti par les accès mémoire, on intercale une mémoire rapide (mais chère!) appelée « mémoire cache » où un contrôleur de mémoire copie les instructions et données qui ont de fortes chances d'être utilisées prochainement par le processeur.



On distingue également :

La mémoire **vive** ou **volatile** appelée également **RAM** (Random Access Memory) dont le contenu est perdu dès que l'ordinateur est éteint.

La mémoire **morte** ou **non volatile** appelée également **ROM** (Read Only Memory) dont le contenu ne s'efface pas lorsque l'ordinateur est éteint. Par exemple, le programme qui se lance au démarrage de l'ordinateur (BIOS ou UEFI) est stocké en mémoire morte et il a notamment pour rôle d'aller chercher sur le disque dur les fichiers de démarrage du système d'exploitation de l'ordinateur pour les copier dans la mémoire vive.

VI) L'horloge

Le processeur est cadencé par une horloge. A chaque impulsion de cette horloge, il exécute une instruction ou une partie d'instruction. La durée entre deux impulsions est choisie pour laisser le temps aux signaux électriques de se propager dans tout le processeur.

Si l'on augmente la fréquence de l'horloge (overclocking), le processeur calcule plus vite mais risque de faire des erreurs car les signaux n'auront pas toujours le temps d'arriver au bon endroit. De plus, les transistors utilisés dans les circuits intégrés consomment surtout au moment où ils « commutent » c'est à dire passent du niveau électrique 0 à 1 ou 1 à 0. Plus un processeur va vite, plus ses transistors commutent souvent, plus le processeur chauffe et peut même être détruit !

VII) Quelques ordres de grandeur

Exemple avec un processeur Intel® Core™ i5 :

- Jeu d'instructions : x86-64 (environ 200 instructions)
- Nombre de transistors : 1,3 milliards
- Registres : Une vingtaine de registres 64 bits
- Fréquence d'horloge : 4 Ghz
- Mémoire cache : 24 Mo
- Nombre de cœurs : 10
- Précision de la gravure : 10 nm
- Prix : 200 €

En savoir plus :

- Sur les circuits intégrés en général : <https://www.youtube.com/watch?v=ee-LhNZPZ1U>
- Sur la fabrication des processeurs : <https://www.youtube.com/watch?v=NFr-WyytNfo>