

Architecture des processeurs

Mémoire vive

La **mémoire vive**, ou RAM (Random Access Memory), d'un ordinateur peut être vue comme une série d'emplacements dans lesquels il est possible de stocker des valeurs. En gros, dans chaque emplacement, il est possible de stocker la valeur d'une variable. Pour des objets plus complexes, comme des listes ou des textes, il peut être nécessaire d'utiliser plusieurs emplacements. Chacun de ces emplacements est identifié par une adresse.

Il y a deux opérations possibles sur la mémoire : la lecture ou l'écriture. Pour une lecture, il faut juste indiquer l'adresse et pour une écriture il faut aussi donner la valeur à inscrire. Il est possible de passer de n'importe quelle adresse à n'importe quelle autre, sans passer par toutes celles au milieu. Ce qui explique le nom RAM, ou "mémoire à accès arbitraire".

Il est possible de fabriquer de la mémoire vive à l'aide de condensateurs ou de transistors. Dans le derniers cas, on parle de circuits séquentiels, puisque le résultat dépend de l'état précédent de la mémoire. Dans les deux cas, cette mémoire doit être alimentée électriquement pour perdurer. C'est une mémoire volatile, contrairement aux supports de stockages comme les disques durs ou les clefs usb.

Architecture de von Neumann

Le CPU (Central Processing Unit), plus couramment appelé **processeur**, est au cœur de l'ordinateur. C'est lui qui exécute les instructions des programmes. L'architecture globale des processeurs est appelée **Architecture de von Neumann**, du nom du mathématicien, et physicien, américano-hongrois John von Neumann (1903–1957). Ce n'est pas la seule architecture possible, mais c'est celle qui est encore utilisée par tous les ordinateurs.

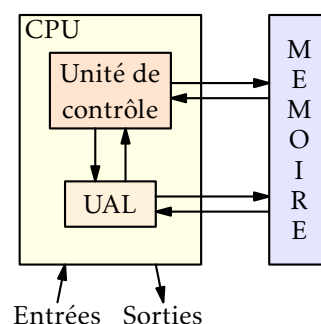
Un processeur s'articule autour de trois parties :

- L'UAL qui réalise les calculs ou les opérations logiques ;
- l'unité de commande qui exécute les instructions des programmes ;
- une petite quantité de mémoire, appelée **registres**, qui permet de stocker les résultats intermédiaires et les valeurs nécessaires pour les opérations en cours.

Le processeur interagit avec la mémoire vive afin de lire et d'écrire les valeurs des variables, leur nombre excédant rapidement le nombre de registres disponibles. Afin de communiquer, les différents composants utilisent un système appelé **bus**. Il y a 3 grands types de bus :

- Le bus d'adresse, par exemple pour envoyer une adresse mémoire.
- Le bus de données, pour faire circuler les données.
- Le bus de contrôle pour spécifier le type d'action, par exemple écrire ou lire une donnée en mémoire.

Une particularité de l'architecture de von Neumann, c'est que les instructions des programmes sont également stockées en mémoire avec les données. Un des registres du processeur contient l'adresse de la prochaine instruction à exécuter. Il est donc tout a fait possible de modifier les instructions en cours d'exécution. C'était nécessaire, initialement, pour réaliser des boucles. De nos jours il y a d'autres méthodes plus simples pour faire des boucles, mais il est toujours possible de modifier les instructions d'un programme en cours d'exécution. C'est notamment ce que font, ou cherchent à faire, les virus informatiques.



Un processeur peut avoir plusieurs **cœurs**, qui sont autant de “petits” processeurs permettant d’exécuter en parallèle plusieurs programmes ou de se partager certains calculs pour obtenir un résultat plus rapidement. Il n’est malheureusement pas possible de le faire à chaque fois et la **programmation parallèle** reste très difficile.

Jeu d’instructions

Chaque processeur est capable d’exécuter un certain nombre d’instructions qui dépendent des circuits qu’il contient. L’ensemble de ces instructions s’appelle le **jeu d’instructions**. Chaque instruction est composée d’une suite de bits. On appelle cela le **langage machine**. Typiquement, un processeur 32 bits acceptera des instructions de 32 bits. Ces instructions sont de plusieurs natures, mais on peut les regrouper en 3 grands types :

- copier la valeur d’un registre en mémoire, à l’adresse donnée ;
- copier le contenu de la mémoire à l’adresse donnée dans un registre ;
- effectuer une opération arithmétique ou logique et mettre le résultat dans un registre.

Afin de déterminer l’action à effectuer, l’instruction est décodée, en général en 2 partie :

- le **code opération** (ou opcode) qui indique le type de traitement à réaliser ;
- les **opérandes** qui indique les données à manipuler.

Il y a 3 types d’opérandes possibles :

- une **valeur immédiate** qui sera utilisée dans l’opération, comme un entier par exemple ;
- un registre du processeur, contenant la valeur à manipuler ou dans lequel il faut stocker le résultat ;
- un adresse en mémoire vive, indiquant la case mémoire dans laquelle écrire ou lire la valeur.

Le type du processeur définit la taille des valeurs que peuvent prendre les valeurs dans les registres et la taille des adresses. Ainsi, pour un processeur 32 bits, les valeurs manipulées sont représentées sur 32 bits. Pour des entiers naturels, cela fait donc des valeurs allant de 0 à 4294967295. Ce qui explique aussi pourquoi un ordinateur 32 bits ne peut pas avoir plus de 4Go de mémoire, puisqu’il n’a pas assez d’adresses pour avoir plus.

Les registres sont généralement notés R0, R1, et ainsi de suite. Il y a néanmoins 2 registres particuliers. Le premier, nommé IP (instruction pointer), ou PC (program counter), contient l’adresse, dans la mémoire vive, de la prochaine instruction à exécuter. Le second, IR (instruction register) contient l’instruction qui va être exécutée immédiatement par le processeur. À chaque nouveau cycle, le processeur effectue les étapes suivantes :

- 1) il copie dans le registre IR le contenu de la mémoire vive à l’adresse pointée par IP ;
- 2) il décode l’instruction contenue dans IR, ce qui active le circuit électrique qui réalise l’opération visée ;
- 3) il exécute l’instruction décodée et met à jour la valeur de IP pour continuer le programme.

Puisqu’une instruction peut modifier directement la valeur de IP, il est possible de réaliser des boucles ou des tests.

Fondamentalement, tous les processeurs sont capables de faire la même chose. Certains seront juste plus rapides ou il sera plus simple de programmer certaines fonctionnalités. S’il est possible d’écrire un programme permettant de réaliser une tâche donnée pour un processeur, il est possible d’écrire un programme faisant la même tâche pour un autre processeur, même si le jeu d’instructions est différent. Par exemple certains processeurs sont capables de réaliser directement des multiplications et des divisions alors que pour des modèles plus anciens, il fallait écrire un bout de programme pour effectuer ces opérations.

Il y a également des processeurs ultra spécialisés pour certaines opérations, comme les processeurs sonores ou ceux des cartes graphiques.