

ARCHITECTURES MULTIPROCESSEURS

On utilise aujourd'hui des architectures multiprocesseurs afin d'améliorer la rapidité d'exécution sans augmenter la fréquence d'horloge. Chaque processeur a alors accès à la même mémoire.

Cœur : le cœur physique est une unité de traitement indépendante exécutant des instructions indépendamment. Plusieurs cœurs peuvent exécuter des instructions simultanément, ce qui est utile lorsque les applications sont parallélisées. C'est à dire lorsqu'elles sont divisées en plusieurs tâches.

Thread : cœur logique. Unité d'exécution sur un cœur. Ces cœurs logiques sont créés par le CPU (processeur) pour fluidifier le comportement de la machine et accélérer les calculs. Il s'agit de fils d'exécution. Chaque cœur physique en possède 1 ou 2 (en 2024).

L'utilisation efficace des architectures multiprocesseurs nécessite de pouvoir traiter les informations de manière simultanée, pour cela il faut des algorithmes spécialisés: on parle de parallélisme.

LES DIVERS TYPES DE MÉMOIRE

Les registres du processeur: Le processeur possède un très petit nombre de cases mémoires, très rapides, appelées des *registres* dont la capacité dépasse rarement quelques dizaines d'octet pour des raisons de coût et de place. Ces mémoires permettent de mémoriser l'état courant (Program Counter) et les données en cours de traitement. Les registres sont partagés entre l'UC et l'UAL.

Mémoire cache : mémoire intermédiaire placée entre le processeur et la mémoire centrale pour optimiser la capacité de traitement. La mémoire cache L1 est au plus près de l'UC, puis viennent L2 -> L3 -> RAM.

RAM : (random access memory) La mémoire désignée dans l'architecture de Von Neumann est ce que l'on appelle aujourd'hui la **mémoire vive (RAM)**. Celle-ci est *rapide*, mais a besoin d'être alimentée en permanence pour garder les données, ainsi à chaque extinction de l'ordinateur, elle est perdue, on la qualifie de *mémoire volatile*.

Pour résoudre ce problème on recourt à deux types de mémoires non volatiles:

La mémoire de masse: Pour stocker **les données et les programmes**, on ajoute un périphérique appelé mémoire de masse: le disque dur de vos ordinateurs, ou une mémoire flash dans le cas des tablettes et smartphones. Cette mémoire est capable de stocker une grande quantité de données, mais a l'inconvénient d'être beaucoup moins rapides que la mémoire vives (RAM), c'est pour cela que lors du lancement d'un programme les données nécessaire à son exécution sont généralement transférées vers la RAM pour une exécution plus rapide.

La mémoire morte est une mémoire qui ne peut être que lue (ROM Read Only Memory), elle contient en général le *firmware* de l'ordinateur (BIOS ou UEFI) qui est le programme qui se charge à chaque allumage de l'ordinateur. La définition a été élargie aux mémoires dites EPROM (microcontrôleurs), dans *l'architecture de Harvard*. La mémoire EPROM contient les programmes qui seront exécutés par le microcontrôleur. Pour modifier ces programmes, il faut utiliser une autre machine, capable d'éditer et transférer le programme sur la mémoire EPROM du microcontrôleur (via un câble usb).

Bus

Un bus est un ensemble de fils conduisant les signaux d'un composant du système informatique à un autre. Le volume d'information transmises simultanément, sur plusieurs fils en parallèle.

Il y a 3 sortes de Bus : Adresse/ Données / Contrôle.

Pourquoi 3 bus différents ? Imaginez que le processeur ait besoin de charger une valeur à l'adresse **ef193a** de la mémoire RAM :

- il envoie la référence de l'adresse souhaitée à la RAM
- il envoie le signal LECTURE
- il charge la donnée

QUESTIONS :

1. Associer les capacités à chaque type de mémoire (Registre, RAM, disque dur, Cache L1) : 64 octets, 16Mo, 128Mo, 1 To.
2. Vrai ou Faux : Les ordinateurs possèdent tous une mémoire morte.
3. Vrai ou Faux : La mémoire morte n'est réinscriptible que sur les machines à l'architecture de Harvard.
4. Le processeur a besoin de charger une valeur à l'adresse **ef193a** de la mémoire RAM. Quels bus sont sollicités, et pour quel usage ?
5. Vrai ou Faux : un processeur à 8 cœurs va forcément exécuter 2 threads en même temps. Expliquer.
6. Parmi les langages suivants, quels sont ceux de bas niveau/ haut niveau : C, langage assembleur, Python
7. Citer au moins 2 langages compilés.
8. Citer au moins 2 langages interprétés.
9. Vrai ou Faux : Le processeur ne comprend que les codes opération codés en langage machine.
10. Vrai ou Faux : le langage assembleur est une traduction sous forme de mnémonique des codes opérations binaires.
11. Le langage assembleur est-il encore utilisé de nos jours. Et si oui, pourquoi ?