NSI · PREMIÈRE / COURS

L'architecture d'un ordinateur

Astrolab Cours · astrolabcours.fr

01 Le problème

Un ordinateur exécute des milliards d'instructions par seconde, alors que ses composants ne savent faire que des choses très simples : laisser passer du courant ou non. Comment s'organise une machine capable de tout calculer ?

01 L'idée : le cuisinier et sa recette

Un cuisinier lit la recette ligne par ligne : il lit l'instruction, comprend ce qu'elle demande, l'exécute, puis passe à la suivante. La recette et les ingrédients sont rangés dans le même placard. Le processeur, c'est le cuisinier ; la mémoire, c'est le placard.

01 Le modèle de von Neumann

Depuis 1945, presque tous les ordinateurs suivent l'**architecture de von Neumann** :

- le **processeur** (CPU), formé de l'**unité de commande** (qui lit et décode les instructions) et de l'**unité arithmétique et logique** (UAL, qui calcule) ;
- la **mémoire**, qui contient à la fois **le programme et les données** ;
- les **entrées-sorties** (clavier, écran, disque, réseau) ;
- des **bus**, qui transportent les informations entre eux.

01 Le cycle d'exécution

Le processeur répète sans fin :

1. **charger** l'instruction dont l'adresse est dans le compteur de programme ;
2. la **décoder** ;
3. l'**exécuter**, puis passer à l'instruction suivante.

Il garde quelques valeurs dans des **registres**, de toutes petites mémoires internes très rapides. Un processeur à 3 GHz fait 3 milliards de cycles d'horloge par seconde.

Le langage machine

Les instructions sont des nombres binaires. L'**assembleur** les écrit avec des mots courts :

```
LOAD R1, 10     ; R1 reçoit le contenu de l'adresse 10
ADD R1, R1, 5   ; R1 = R1 + 5
```

© Astrolab Cours · astrolabcours.fr/nsi/premiere/architecture-d-un-ordinateur/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

```
STORE R1, 11    ; on range R1 à l'adresse 11
```

Un programme Python est traduit, au bout du compte, en instructions de ce genre.

Les portes logiques

Tout calcul est construit avec des **portes logiques**, réalisées par des transistors :

a	b	ET	OU	NON a
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

En combinant des portes, on fabrique un additionneur, puis toute l'UAL. Un demi-additionneur de deux bits donne la somme `a XOR b` et la retenue `a ET b`.

Les pièges

- Mémoire vive (RAM, effacée à l'extinction) et stockage (disque, permanent) ne sont pas la même chose.
- Les registres sont dans le processeur, pas dans la RAM.
- Une fréquence plus élevée ne suffit pas à dire qu'un processeur est plus rapide (nombre de cœurs, mémoire cache…).

© Astrolab Cours · astrolabcours.fr/nsi/premiere/architecture-d-un-ordinateur/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits