Architecture

Modèle de Von Neumann

Mise en situation

Quand vous lancez un programme Python, que se passe-t-il concrètement dans la machine ? Le code est stocké quelque part, des calculs sont effectués, des résultats sont affichés à l'écran. Pour comprendre ce cheminement, il faut connaître l'architecture interne d'un ordinateur, c'est-à-dire les composants qui le constituent et la façon dont ils communiquent entre eux.

Le modèle de Von Neumann

En 1945, le mathématicien John von Neumann publie un rapport décrivant l'architecture que devrait avoir tout calculateur universel. Cette architecture, connue sous le nom de **modèle de Von Neumann**, reste le fondement de la quasi-totalité des ordinateurs actuels.

L'idée centrale est que le **programme et les données sont stockés ensemble dans la mémoire**. Avant Von Neumann, les programmes étaient câblés physiquement (cartes perforées, tableaux de connexions). L'innovation décisive est de les enregistrer en mémoire sous forme de nombres, au même titre que les données. Cela permet de changer de programme sans modifier le matériel.

Le modèle décompose l'ordinateur en quatre parties : l'unité arithmétique et logique, l'unité de commande, la mémoire et les entrées-sorties. Ces parties sont reliées par des **bus** (canaux de communication).

Les quatre composants

Le processeur (CPU) regroupe les deux premières parties :

— l'**unité arithmétique et logique** (UAL) effectue les calculs (additions, comparaisons, opérations logiques ET, OU, NON) ;

— l'**unité de commande** lit les instructions en mémoire, les décode et coordonne les autres composants pour les exécuter.

La mémoire stocke les instructions du programme et les données qu'il manipule. On distingue la mémoire vive (rapide, volatile) et la mémoire de masse (lente, persistante).

Les entrées-sorties permettent à l'ordinateur de communiquer avec l'extérieur : clavier, souris, écran, imprimante, réseau.

Les bus ne forment pas une cinquième partie : ils relient ces composants entre eux. Il en existe trois types : le bus de données (transporte les valeurs), le bus d'adresse (indique l'emplacement mémoire visé) et le bus de commande (transmet les ordres : lire, écrire, etc.).

Évolutions du modèle

Le modèle original a évolué sur deux points :

— les entrées-sorties sont désormais gérées par des contrôleurs autonomes, ce qui permet au processeur de ne pas attendre la fin d'une opération d'écriture sur disque ;

— les processeurs modernes contiennent plusieurs **cœurs** capables d'exécuter des instructions en parallèle, ce qui augmente la puissance sans augmenter la fréquence.

Le processeur

Le processeur (CPU, *Central Processing Unit*) est le composant qui exécute les instructions. C'est le « cerveau » de l'ordinateur.

Structure interne

Un processeur contient :

— des **registres** : petites mémoires ultrarapides qui stockent temporairement les données en cours de traitement. Parmi les registres importants : le **compteur ordinal** (PC, *Program Counter*) qui contient l'adresse de la prochaine instruction, le **registre d'instruction** (IR) qui contient l'instruction en cours de décodage, et l'**accumulateur** qui stocke les résultats intermédiaires ;

— l'**UAL** qui effectue les opérations arithmétiques et logiques ;

— le **séquenceur** qui coordonne l'exécution des instructions au rythme d'une horloge.

Le cycle fetch-decode-execute

Le processeur exécute chaque instruction en trois étapes qui se répètent indéfiniment :

Étape 1 : Fetch (charger). Le processeur lit en mémoire l'instruction dont l'adresse est contenue dans le compteur ordinal (PC). Cette instruction est copiée dans le registre d'instruction (IR). Le compteur ordinal est incrémenté.

Étape 2 : Decode (décoder). Le séquenceur analyse l'instruction contenue dans IR et détermine l'opération à effectuer et les données concernées.

Étape 3 : Execute (exécuter). L'UAL exécute l'opération. Le résultat est placé dans un registre ou écrit en mémoire.

Exemple :

```
Cycle 1 :
  Fetch   -> PC = 0. IR = "LOAD 100"
  Decode  -> Charger la valeur a l'adresse 100 dans l'accumulateur
  Execute -> Accumulateur = 42

Cycle 2 :
  Fetch   -> PC = 1. IR = "ADD 104"
  Decode  -> Additionner la valeur a l'adresse 104
  Execute -> Accumulateur = 42 + 15 = 57

Cycle 3 :
  Fetch   -> PC = 2. IR = "STORE 108"
  Decode  -> Ecrire l'accumulateur a l'adresse 108
  Execute -> Adresse 108 = 57
```

Ce cycle se répète un très grand nombre de fois par seconde. Un processeur cadencé à 3 GHz reçoit trois milliards d'impulsions d'horloge par seconde ; comme chaque instruction demande plusieurs impulsions, il exécute, dans ce modèle simplifié, de l'ordre d'un milliard d'instructions par seconde.

La fréquence d'horloge

Le processeur est cadencé par une **horloge** interne : un cristal de quartz qui émet des impulsions régulières. La **fréquence** mesure le nombre d'impulsions par seconde, exprimée en hertz. Un processeur à 3 GHz reçoit trois milliards d'impulsions par seconde.

Les catégories d'instructions

Les instructions se regroupent en quatre familles :

Catégorie	Exemples
Transfert de données	Charger une valeur, stocker un résultat
Arithmétique	Addition, soustraction, multiplication, division
Logique	ET, OU, NON, OU exclusif, comparaisons
Contrôle	Saut conditionnel, appel de fonction

La mémoire

La mémoire stocke les programmes et les données. Il existe plusieurs types de mémoire, qui se distinguent par leur vitesse, leur capacité et leur persistance.

La hiérarchie des mémoires

Plus une mémoire est rapide, plus elle est coûteuse et donc petite. Cela conduit à une organisation en couches :

Type	Vitesse	Capacité	Volatile ?	Rôle
Registres	< 1 ns	Quelques dizaines	Oui	Calcul en cours
Cache L1	~ 1 ns	64 Ko	Oui	Données fréquentes
Cache L2	~ 5 ns	256 Ko à 1 Mo	Oui	Tampon L1–RAM
Cache L3	~ 10 ns	4 à 32 Mo	Oui	Partagé entre cœurs
RAM	~ 50 ns	8 à 32 Go	Oui	Programme en cours
SSD	$\sim 100\,\mu$s	256 Go à 2 To	Non	Stockage permanent
Disque dur	~ 10 ms	1 à 4 To	Non	Stockage de masse

Principe. Le processeur cherche d'abord la donnée dans le cache L1. Si elle n'y est pas, il la cherche en L2, puis en L3, puis en RAM. Plus on descend dans la hiérarchie, plus l'accès est lent mais plus la capacité est grande.

Mémoire vive (RAM)

La **RAM** (*Random Access Memory*) est la mémoire de travail de l'ordinateur. Elle stocke les programmes en cours d'exécution et leurs données. Elle est **volatile** : son contenu disparaît dès que l'ordinateur est éteint.

Mémoire de masse

La mémoire de masse est **persistante** : elle conserve les données même sans alimentation électrique. Les principaux supports sont :

— le **disque dur** (HDD), qui utilise des plateaux magnétiques rotatifs ;

— le **SSD** (*Solid State Drive*), qui utilise de la mémoire flash (sans pièce mobile), plus rapide, plus silencieux et plus résistant aux chocs.

La carte mère

La **carte mère** est le circuit imprimé principal de l'ordinateur. Elle accueille le processeur (dans un socket), la RAM (dans des slots), et relie tous les composants entre eux par des bus. Elle intègre aussi un chipset qui coordonne les échanges de données, ainsi que des connecteurs pour les périphériques (USB, réseau, écran).

La carte mère contient enfin une petite mémoire qui conserve son contenu sans alimentation. Elle renferme le premier programme exécuté à la mise sous tension : le **BIOS** (*Basic Input Output System*), remplacé sur les ordinateurs récents par l'**UEFI** (*Unified Extensible Firmware Interface*). Ce programme vérifie et initialise le matériel, puis charge le système d'exploitation en mémoire vive.

La loi de Moore

En 1965, Gordon Moore observe que le nombre de transistors sur une puce double chaque année ; en 1975, il ramène ce rythme à un doublement tous les deux ans. Cette observation empirique, connue sous le nom de **loi de Moore**, s'est vérifiée pendant plus de cinquante ans : le premier microprocesseur (Intel 4004, 1971) contenait 2 300 transistors ; un processeur actuel en contient plusieurs milliards.

Cette croissance exponentielle a permis de fabriquer des ordinateurs toujours plus puissants et moins coûteux. Cependant, la miniaturisation approche des limites physiques (taille des atomes, dissipation thermique), ce qui a conduit les fabricants à multiplier les cœurs plutôt qu'à augmenter la fréquence.

Résumé

Concept	Ce qu'il faut retenir
Modèle de Von Neumann	UAL + unité de commande + mémoire + E/S, reliées par des bus ; programme en m
Processeur (CPU)	UAL + unité de commande + registres
Cycle fetch-decode-execute	Charger, décoder, exécuter ; se répète indéfiniment
Compteur ordinal (PC)	Adresse de la prochaine instruction
RAM	Mémoire rapide, volatile, pour les programmes en cours
Mémoire de masse	Stockage persistant (SSD, disque dur)
Mémoire cache (L1, L2, L3)	Mémoire rapide intégrée au processeur
Bus	Canaux de données, d'adresse et de commande
Loi de Moore	Le nombre de transistors double tous les deux ans