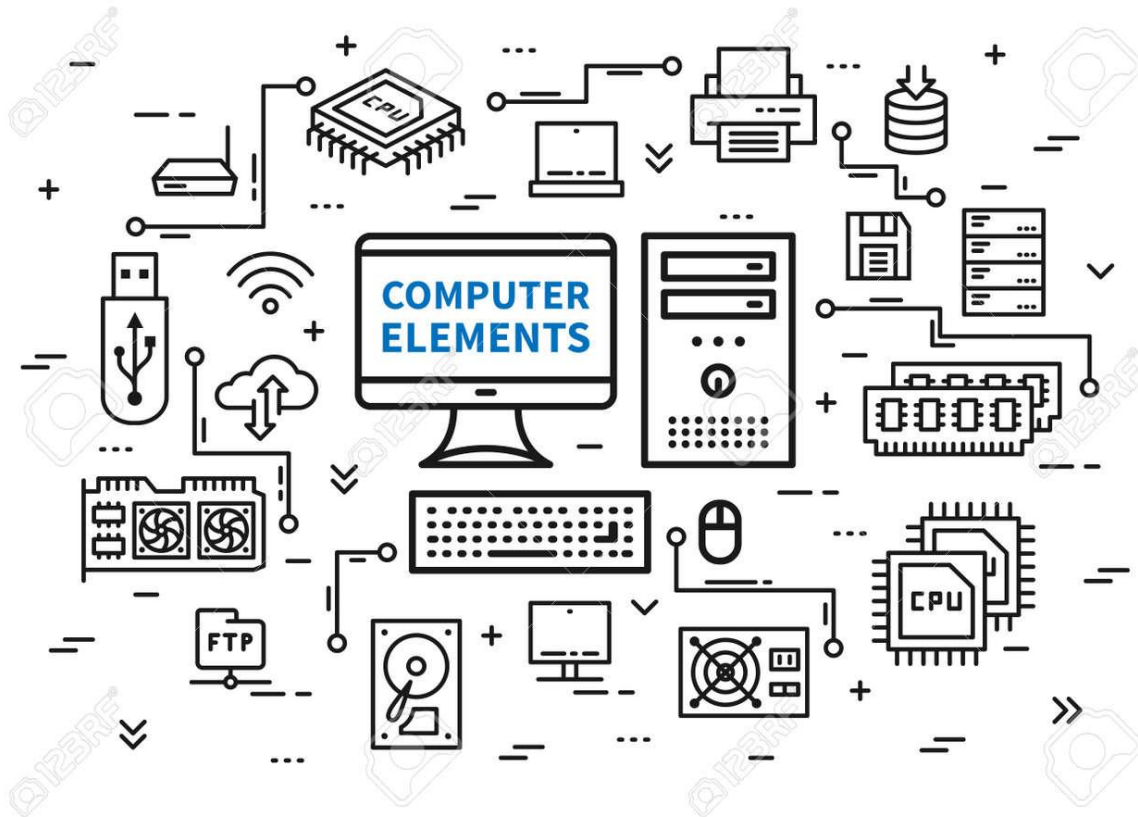


Les composants d'un ordinateur

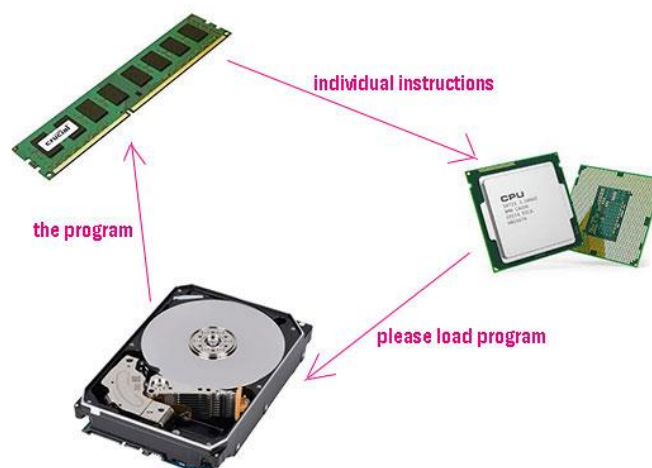


Les composants d'un ordinateur	1
Objectifs du cours.....	3
Définition de l'ordinateur	5
Ordinateurs analogiques : une curiosité historique	5
Introduction à l'information digitale	6
La logique binaire : fondement de l'information digitale	7
Zone d'indétermination et stabilité des systèmes numériques.....	7
L'étendue des données : Data range et importance des bits	8
Conclusion : Ordinateur numérique et ses fondations	8
Les composants d'un ordinateur élémentaire.....	8
La mémoire centrale	9
Les interfaces d'entrée/sortie (I/O)	9
Le processeur (CPU)	10
La carte mère et les bus.....	11
Les différents types de bus	12
La gestion des bus : Éviter les conflits.....	13
Coordination et gestion des composants actifs.....	13
L'organisation des bus dans un système informatique.....	14
Architecture des bus : Von Neumann vs Harvard	14
Différences entre les architectures Von Neumann et Harvard	15
L'architecture de Von Neumann.....	15
L'architecture Harvard	16
Comparaison et application des deux architectures	16
Le processus de démarrage de l'ordinateur.....	17
Étape 1 : Mise sous tension et RESET	17
Étape 2 : Lecture de l'adresse initiale en mémoire ROM	17
Étape 3 : Exécution des premières instructions	18
Étape 4 : Initialisation du matériel	18
Étape 5 : Chargement du système d'exploitation	18
Différences d'accès entre la mémoire centrale et la mémoire de masse	19
Les registres internes du CPU	19

Objectifs du cours

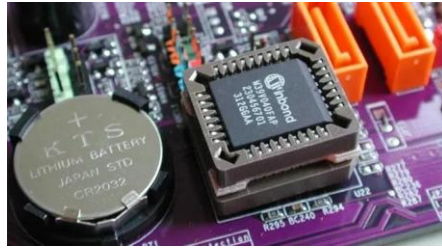
Ce cours d'**architecture des ordinateurs** vise à fournir une compréhension détaillée des composants élémentaires d'un système informatique. L'objectif principal est d'identifier et de décrire ces composants, tout en expliquant leur rôle spécifique et leur fonctionnement au sein d'un ordinateur simple. Cela comprend une introduction aux concepts de base des systèmes informatiques, ainsi qu'une analyse approfondie de leurs différents éléments matériels.

Dans un premier temps, le cours introduira les principaux composants d'un ordinateur, tels que le processeur (CPU), la mémoire (RAM et ROM), les bus de données, et les périphériques d'entrée/sortie. Chaque composant sera ensuite étudié dans ses caractéristiques propres : son rôle fonctionnel, son mode de communication avec les autres éléments, ainsi que son impact sur les performances globales du système.



Une attention particulière sera portée à la manière dont ces composants interagissent entre eux pour assurer un fonctionnement cohérent. L'importance des bus, des registres et de l'architecture mémoire sera discutée afin de permettre une compréhension claire des flux d'informations et des processus d'adressage. En particulier, le cours abordera la manière dont les bus de données et d'adresses permettent le transfert des informations entre le processeur et les différents périphériques.

Enfin, le processus de démarrage d'un ordinateur sera détaillé, avec une étude du **boot sequence** et des premières instructions exécutées à partir de la mémoire centrale. Les mécanismes de démarrage, tels que le rôle du BIOS et la lecture des premières instructions en mémoire ROM, seront couverts pour expliquer comment un système passe de l'état hors tension à un état opérationnel.



Les concepts abordés dans ce cours reposent sur des bases théoriques solides, en s'appuyant sur des ouvrages de référence en architecture des ordinateurs, tels que *Computer Organization and Design* de Patterson et Hennessy.

Définition de l'ordinateur

Un ordinateur est une machine électronique capable de traiter des informations en suivant des instructions préalablement programmées. Il manipule des données en effectuant des opérations d'écriture, de réécriture, de suppression ou de modification, et ce, en accord avec un ensemble de règles définies. Ces opérations se poursuivent jusqu'à ce que des conditions d'arrêt, comme la mise hors tension, soient atteintes.



En d'autres termes, un ordinateur peut être vu comme un dispositif qui déplace, stocke et traite des données à travers différents composants électroniques, utilisant principalement des courants électriques pour fonctionner. À travers ces courants, l'ordinateur est capable de réaliser des calculs complexes à grande vitesse, en exécutant des tâches allant du traitement de texte à l'analyse de données massives.

Ordinateurs analogiques : une curiosité historique

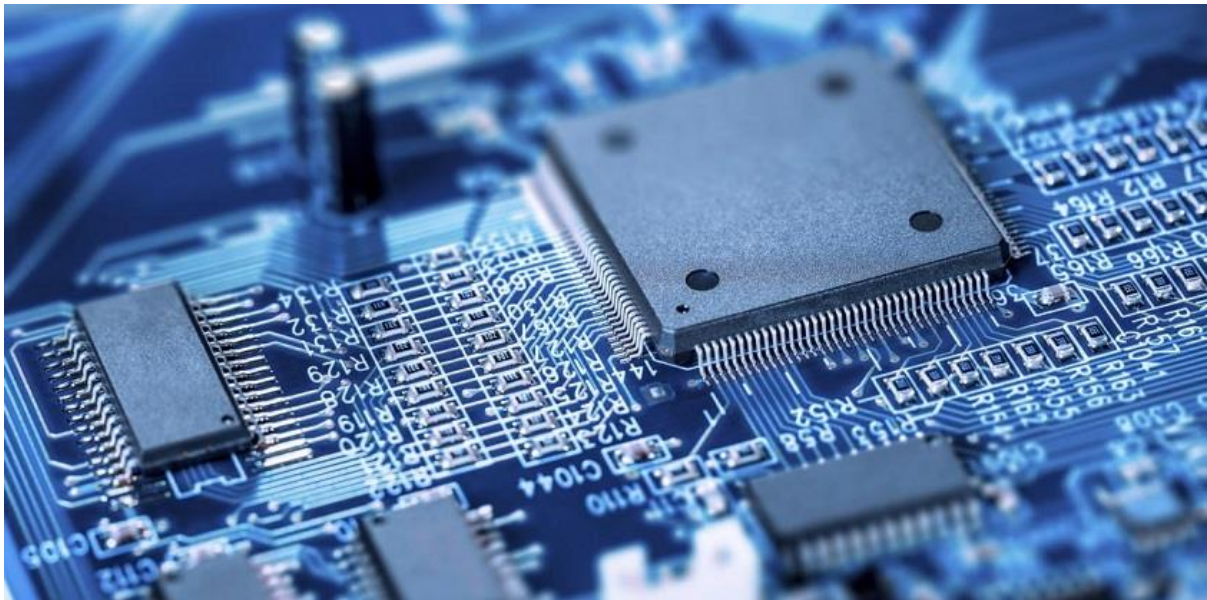


Au début de l'ère informatique, les premiers ordinateurs étaient **analogiques**. Ces machines traitaient des données continues, c'est-à-dire des valeurs qui pouvaient prendre n'importe quelle mesure dans un ensemble infini de réels. Les ordinateurs analogiques étaient particulièrement utilisés pour les calculs scientifiques complexes, comme la modélisation des systèmes physiques. Le **différenciateur de Mealy**, une des premières machines analogiques, en est un bon exemple. Ces systèmes utilisaient des tensions électriques continues pour représenter des quantités telles que la température ou la vitesse, et les manipulaient par des moyens mécaniques ou électroniques.

L'avènement des ordinateurs numériques a toutefois rapidement supplanté les machines analogiques. Les ordinateurs numériques, contrairement aux analogiques, manipulent des données discrètes, codées en binaire, ce qui permet une plus grande précision, une meilleure fiabilité et une scalabilité accrue des systèmes informatiques.

Introduction à l'information digitale

Les ordinateurs modernes sont des machines **numériques**, également appelées **digitaux**. Dans un système numérique, l'information est représentée par des **signaux électriques** qui prennent des valeurs discrètes, souvent codées sous forme binaire (0 ou 1). Ce système de codage permet une simplification des opérations arithmétiques et logiques, ainsi qu'une communication plus fiable des informations à travers les différents circuits.



Un signal binaire repose sur des niveaux de tension pour représenter ces deux états. Par exemple, dans un système fonctionnant sous une tension de 5 volts, une tension proche de 0 volt représente un état logique "0", tandis qu'une tension proche de 5 volts

représente un état logique "1". Ces niveaux de tension varient en fonction des standards électroniques utilisés, avec des seuils différents pour des technologies plus modernes comme le **CMOS** (Complementary Metal-Oxide Semiconductor) .

La logique binaire : fondement de l'information digitale

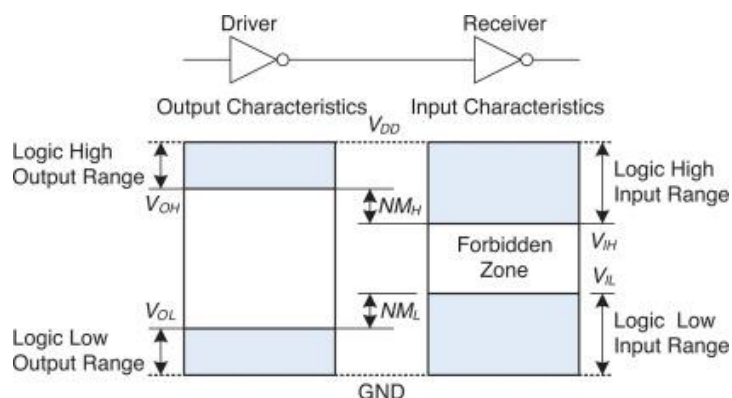
Le fonctionnement de l'ordinateur repose sur la manipulation de l'information sous forme **binaire**. Chaque unité d'information binaire est appelée un **bit** et peut prendre deux états : 0 ou 1. La **résolution** d'un système numérique est déterminée par le nombre de bits qu'il peut traiter simultanément. Par exemple, un **octet** (composé de 8 bits) peut représenter 256 valeurs différentes (de 0 à 255), car il y a $2^8 = 256$ combinaisons possibles de bits.

L'unité de base du binaire est insuffisante pour traiter des données complexes, telles que des images ou des vidéos, c'est pourquoi les ordinateurs combinent les bits pour former des mots plus longs, appelés **mots**, **longs mots**, ou encore **entiers** (integers). Cela permet de coder des valeurs beaucoup plus élevées, ou même des valeurs signées, c'est-à-dire qui incluent des nombres négatifs.

En pratique, pour simplifier la représentation des nombres binaires, on utilise souvent des systèmes de numération plus familiers tels que le **décimal** (base 10) ou **hexadécimal** (base 16). L'hexadécimal est particulièrement utile dans les systèmes numériques pour représenter de longues séquences de bits, car il permet de condenser 4 bits en un seul chiffre hexadécimal.

Zone d'indétermination et stabilité des systèmes numériques

Un aspect essentiel à prendre en compte dans les systèmes numériques est la présence d'une **zone d'indétermination** entre les états logiques "0" et "1". Cette zone correspond à une plage de tensions qui n'est ni assez basse pour être considérée comme un "0", ni assez haute pour être interprétée comme un "1". Par exemple, dans un système à 5 volts, une tension comprise entre 0,05 volt et 2,5 volts serait interprétée comme une zone indéterminée.



La présence d'une zone d'indétermination peut provoquer des erreurs de traitement ou des comportements imprévisibles dans un système numérique. Pour éviter ces problèmes, la conception des circuits électroniques prend soin de définir des seuils de tension clairs et des **marges de tolérance** pour que les transitions entre les états soient nettes et bien distinctes.

L'étendue des données : Data range et importance des bits

L'étendue des données, ou **data range**, fait référence à l'intervalle de valeurs que peut coder un système numérique donné. Par exemple, avec un ensemble de 8 bits non signés, un ordinateur peut représenter des nombres allant de 0 à 255. Cependant, si l'on utilise des bits signés (ce qui permet de représenter des valeurs négatives), l'intervalle des valeurs devient -128 à +127, car un bit est réservé au signe.

Cette capacité à coder des valeurs variées est essentielle pour permettre aux ordinateurs de traiter une grande diversité d'informations, allant des nombres simples aux données complexes comme les fichiers multimédias. Les systèmes numériques modernes utilisent souvent des architectures 32 bits ou 64 bits pour manipuler de plus grandes quantités de données, offrant une plus grande précision et des performances accrues.

Conclusion : Ordinateur numérique et ses fondations

En résumé, un ordinateur numérique est une machine électronique qui manipule des données discrètes, principalement sous forme binaire, à l'aide de signaux électriques. Si les ordinateurs analogiques ont historiquement joué un rôle, c'est grâce à l'introduction des systèmes numériques que les ordinateurs modernes ont pu atteindre des niveaux de précision, de fiabilité et de puissance inégalés. La compréhension de la logique binaire, des bits, et des plages de données est essentielle pour saisir le fonctionnement de ces systèmes.

Le passage d'une technologie analogique à numérique a révolutionné l'informatique, permettant une plus grande flexibilité dans le traitement de l'information et ouvrant la voie à l'ère numérique que nous connaissons aujourd'hui.

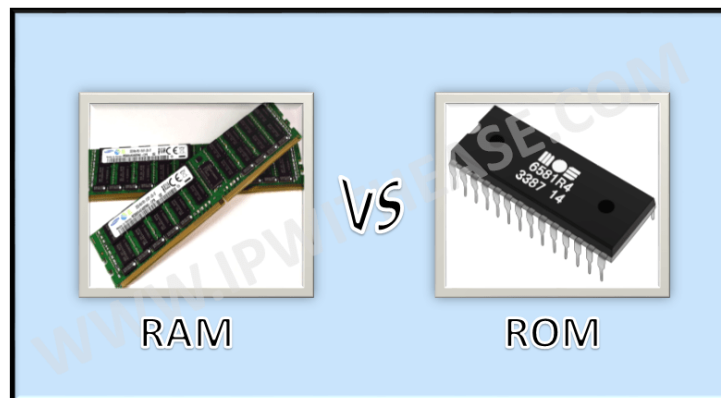
Les composants d'un ordinateur élémentaire

Un ordinateur est composé de plusieurs éléments essentiels, chacun ayant un rôle spécifique dans le traitement, le stockage et la communication des informations. Pour comprendre comment un ordinateur fonctionne, il est important de connaître ces composants de base, qui forment le noyau de tout système informatique moderne.

La mémoire centrale

La **mémoire centrale**, également appelée **mémoire principale**, est l'un des composants essentiels d'un ordinateur. Son rôle est de stocker les données et les instructions nécessaires à l'exécution des programmes. Il existe plusieurs types de mémoires, chacune ayant des caractéristiques et des fonctions différentes.

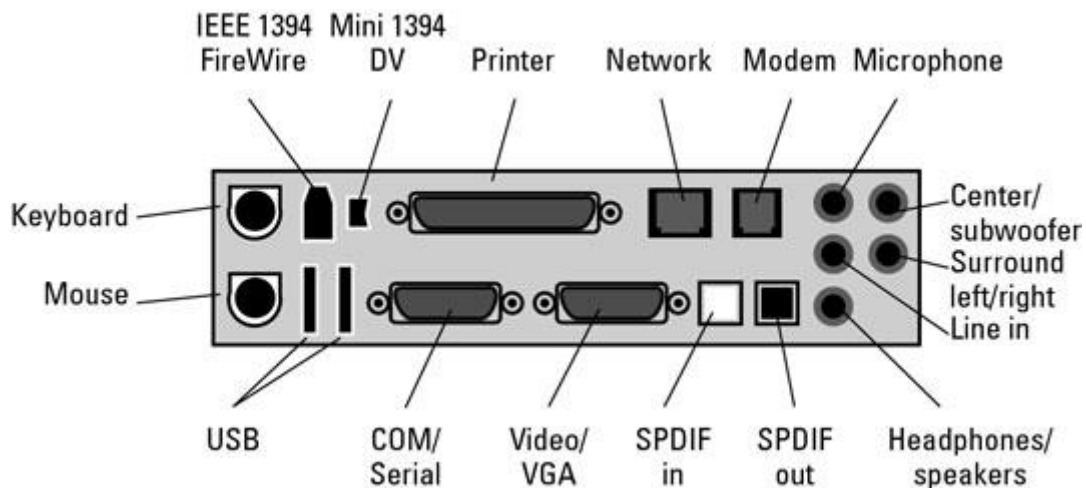
La première catégorie est la **mémoire non volatile**, qui conserve les données même lorsque l'ordinateur est éteint. L'exemple typique de cette mémoire est la **ROM** (Read-Only Memory). La ROM est utilisée pour stocker des informations permanentes, telles que le **BIOS** (Basic Input/Output System), qui est essentiel au démarrage de l'ordinateur. Le BIOS contient les instructions nécessaires pour initialiser les composants matériels et lancer le système d'exploitation.



En revanche, la **mémoire vive** ou **RAM** (Random Access Memory) est une mémoire **volatile**, ce qui signifie que son contenu est perdu dès que l'ordinateur est éteint. La RAM est utilisée pour stocker temporairement les données et les programmes en cours d'exécution, offrant ainsi un accès rapide et direct aux informations nécessaires pour le processeur. Contrairement à la ROM, la RAM est modifiable, ce qui permet de lire et d'écrire des données en temps réel. Il est important de noter que ce qui est "aléatoire" dans la RAM n'est pas le contenu, mais la manière d'y accéder : l'ordinateur peut atteindre n'importe quelle donnée de manière instantanée, indépendamment de sa position physique dans la mémoire.

Les interfaces d'entrée/sortie (I/O)

Les **interfaces d'entrée/sortie** (I/O) permettent à l'ordinateur de communiquer avec le monde extérieur. Ces interfaces incluent des dispositifs tels que le clavier, la souris, le moniteur, ainsi que des périphériques de stockage externes comme les **disques durs**, les **clés USB** et les **lecteurs de bande**. Les interfaces I/O jouent un rôle crucial dans l'interaction entre l'utilisateur et l'ordinateur, ainsi que dans la communication avec d'autres systèmes et périphériques.



Les **ports USB** (Universal Serial Bus) sont un exemple courant d'interface I/O. Ils permettent de connecter différents périphériques à l'ordinateur, tels que des imprimantes, des disques externes ou des clés USB. D'autres interfaces spécialisées, comme les **contrôleurs de disque dur** ou les **contrôleurs de carte réseau**, facilitent des fonctions spécifiques, par exemple la gestion des données sur un disque ou l'interconnexion avec un réseau local ou Internet.

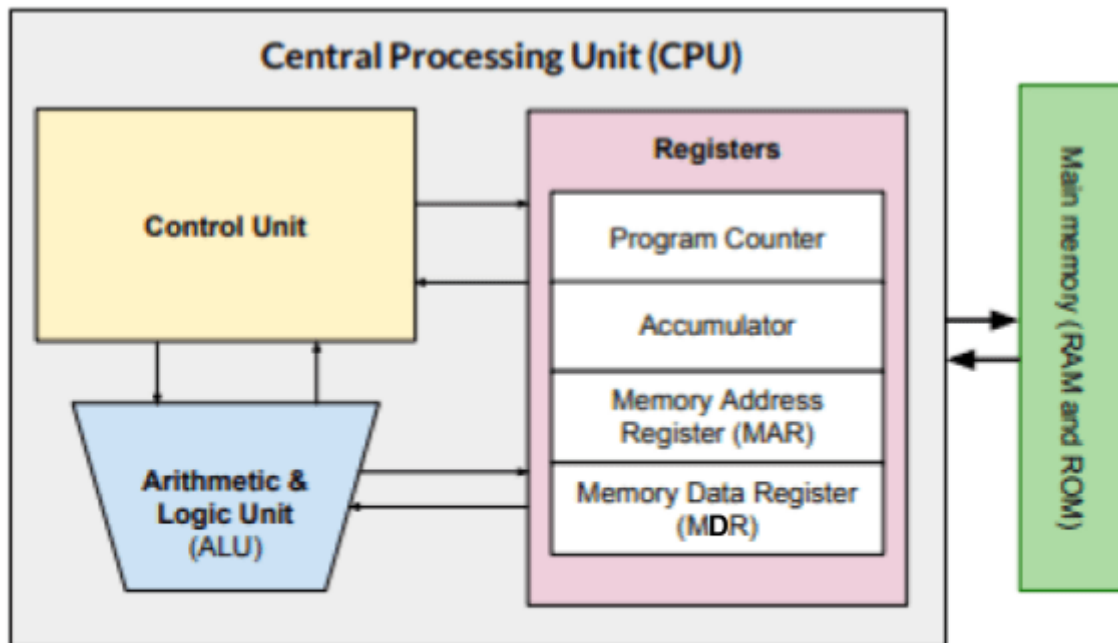
Ces interfaces peuvent être contrôlées et gérées par le processeur, mais aussi par des contrôleurs spécifiques qui optimisent la gestion des entrées et sorties de données. Cela permet d'alléger le travail du processeur et de garantir des performances optimales.

Le processeur (CPU)

Le **processeur**, ou **Central Processing Unit (CPU)**, est souvent décrit comme le **cerveau** de l'ordinateur. C'est le composant qui exécute les instructions des programmes et contrôle l'ensemble des autres composants. Il gère toutes les opérations de calcul et de traitement des données, que ce soit des opérations arithmétiques, logiques ou de contrôle.



Le CPU est constitué de plusieurs sous-unités, dont l'**unité arithmétique et logique** (ALU), qui effectue les calculs mathématiques et logiques, et l'**unité de contrôle**, qui coordonne le flux des instructions à travers l'ordinateur. En plus de cela, le CPU contient également des **registres** qui stockent temporairement les données utilisées lors des opérations.



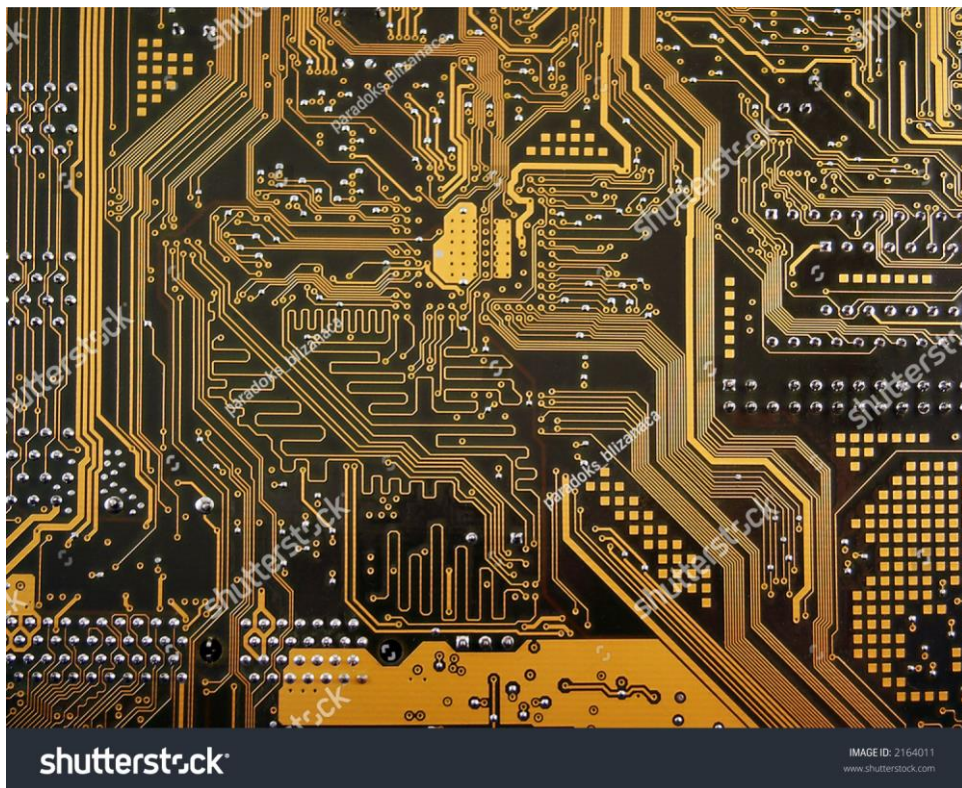
L'une des principales fonctions du processeur est de piloter l'ensemble des composants et périphériques connectés sur la carte mère. Le CPU reçoit des instructions depuis la mémoire centrale, les décode, puis les exécute en accédant à la RAM ou en envoyant des commandes aux périphériques via les bus de données et de contrôle.

Le rôle du processeur est primordial, car il assure non seulement le calcul, mais aussi la synchronisation de tous les éléments de l'ordinateur. Chaque instruction suit un cycle précis – le **cycle machine** – qui inclut la lecture, le décodage et l'exécution des instructions.

La carte mère et les bus

La carte mère est le composant central d'un ordinateur, sur lequel sont connectés tous les éléments essentiels au bon fonctionnement du système. Elle abrite le processeur (CPU), la mémoire vive (RAM), la mémoire morte (ROM) ainsi que les divers périphériques d'entrée et de sortie (I/O). Pour que ces composants puissent interagir

entre eux de manière cohérente et efficace, ils doivent être **interconnectés**. Cette interconnexion est réalisée par des conducteurs, appelés **bus**, qui circulent sur le circuit imprimé de la carte mère sous forme de **pistes**.



Le rôle des bus dans l'architecture d'un ordinateur est fondamental : ils permettent d'échanger des données, de transmettre des adresses mémoire et de gérer les signaux de contrôle qui synchronisent les opérations entre les différents composants.

Les différents types de bus

Un bus est un ensemble de conducteurs électriques, regroupés selon leur fonction, qui assure la transmission des informations. Dans un système informatique, les bus peuvent être classés en trois catégories principales :

1. **Bus de données** : Ce bus est chargé de transporter les données entre le processeur, la mémoire et les périphériques. Chaque bit d'information est véhiculé par une piste distincte sur la carte mère, ce qui signifie que plus le bus a de pistes, plus il peut transférer de données simultanément. Par exemple, un bus de 32 bits peut transférer 32 bits de données en parallèle.
2. **Bus d'adresses** : Ce bus sert à spécifier l'emplacement où les données doivent être lues ou écrites en mémoire. Lorsqu'un processeur veut accéder à une information, il place l'adresse mémoire correspondante sur le bus d'adresses, qui la transmet à la RAM ou à d'autres dispositifs de stockage. La taille du bus

d'adresses détermine la quantité de mémoire que le système peut adresser. Par exemple, un bus d'adresses de 32 bits peut théoriquement adresser jusqu'à 4 Go de mémoire.

3. **Bus de contrôle** : Ce bus transporte les signaux de commande émis par le processeur pour indiquer quel composant doit être activé et quelle opération doit être effectuée (lecture, écriture, etc.). Ces signaux permettent de synchroniser les activités des composants connectés, garantissant ainsi un fonctionnement ordonné du système. Le bus de contrôle contient des lignes dédiées à la gestion de divers signaux, tels que le **signal de lecture/écriture**, le **signal d'horloge**, ou encore le **signal d'interruption**.

La gestion des bus : Éviter les conflits

Il est important de comprendre que plusieurs composants peuvent vouloir utiliser un bus en même temps, ce qui pourrait créer des conflits si les données ou adresses transmises se superposent. Pour éviter cela, des mécanismes de gestion de bus sont mis en place.

L'un des systèmes utilisés est le **tri-state**, qui permet à plusieurs composants de partager un même bus sans provoquer de court-circuit ou de conflits. Ce mécanisme repose sur trois états possibles pour chaque conducteur du bus :

- **État haut (1)** : La tension est présente et l'information transmise est interprétée comme un "1".
- **État bas (0)** : L'absence de tension ou une tension faible est interprétée comme un "0".
- **État déconnecté** (ou en "l'air") : Le composant n'envoie pas de signal sur le bus, ce qui permet à un autre composant d'utiliser le bus sans interférence.

Le tri-state est essentiel dans les architectures informatiques modernes, car il permet à différents périphériques de partager les mêmes lignes de communication sans risque de perturbations ou de court-circuit.

Coordination et gestion des composants actifs

Même avec les bus de données et d'adresses, un problème persiste : comment savoir quel composant est actif à un moment donné ? C'est ici que le **bus de contrôle** entre en jeu. Le bus de contrôle, supervisé par le processeur (CPU), permet de sélectionner le composant qui doit être actif sur les bus de données et d'adresses à un moment donné.

Le bus de contrôle utilise plusieurs types de signaux pour gérer cette sélection :

- **Signaux de sélection** : Le processeur émet un signal pour désigner quel périphérique ou quelle mémoire doit être activé sur le bus.
- **Signaux de lecture/écriture** : Le processeur utilise ces signaux pour indiquer s'il souhaite lire des données d'un périphérique ou de la mémoire, ou bien y écrire des informations.
- **Signaux d'interruption** : Lorsqu'un périphérique doit interrompre le fonctionnement normal du processeur pour demander une action (par exemple, lorsqu'un périphérique I/O a terminé une opération), il envoie un signal d'interruption via le bus de contrôle.

Le bus de contrôle est essentiel pour synchroniser les activités des différents composants, évitant ainsi que plusieurs d'entre eux n'essaient d'utiliser les mêmes ressources au même moment.

L'organisation des bus dans un système informatique

Les bus d'un ordinateur ne se limitent pas à connecter le processeur à la mémoire et aux périphériques. Ils constituent une **infrastructure hiérarchique** complexe, où certains bus sont spécialisés et optimisés pour des tâches spécifiques.

- **Bus local** : C'est le bus le plus proche du processeur. Il permet au processeur de communiquer directement avec des composants critiques tels que la mémoire vive (RAM) et les caches. Ce bus est souvent très rapide et large, pour minimiser la latence des accès mémoire.
- **Bus système** : Le bus système, ou **front-side bus** (FSB), relie le processeur aux principaux composants de l'ordinateur, notamment la RAM, les interfaces I/O, et les cartes d'extension. Le FSB détermine en grande partie les performances globales de l'ordinateur, car il gère les transferts de données entre le processeur et la mémoire centrale.
- **Bus périphérique** : Des bus spécifiques, comme le **PCI** (Peripheral Component Interconnect) ou le **USB** (Universal Serial Bus), sont utilisés pour connecter des périphériques externes (claviers, souris, imprimantes, disques externes) au système. Ces bus permettent des communications rapides et standardisées entre les différents appareils et le reste du système.

Architecture des bus : Von Neumann vs Harvard

Deux architectures principales sont utilisées dans les systèmes informatiques modernes :

1. **Architecture de von Neumann** : Dans ce modèle, le CPU, la mémoire et les périphériques partagent un bus unique pour les données et les instructions.

Bien que cette architecture soit simple et économique, elle souffre du **goulot d'étranglement de von Neumann**, car le CPU doit attendre que les instructions et les données soient transmises sur le même bus.

2. **Architecture Harvard** : Cette architecture, souvent utilisée dans les systèmes embarqués et les microcontrôleurs, sépare les bus de données et les bus d'instructions. Cette séparation permet une meilleure performance, car le processeur peut accéder aux instructions et aux données simultanément, éliminant ainsi le goulot d'étranglement.

Le système de bus est la colonne vertébrale d'un ordinateur moderne. Il permet de connecter les différents composants de manière efficace et ordonnée, tout en évitant les conflits d'accès grâce à des mécanismes tels que le tri-state et le bus de contrôle. L'organisation des bus dans un ordinateur influe directement sur les performances du système, en particulier en ce qui concerne la vitesse des transferts de données et la capacité du processeur à gérer plusieurs tâches simultanément. Les architectures de bus, qu'elles soient de type von Neumann ou Harvard, offrent des solutions différentes mais adaptées aux besoins des systèmes informatiques d'aujourd'hui.

Différences entre les architectures Von Neumann et Harvard

Dans la conception des systèmes informatiques, il existe deux principales architectures utilisées pour organiser les composants d'un ordinateur : l'**architecture de Von Neumann** et l'**architecture Harvard**. Ces deux modèles influencent profondément la manière dont les processeurs communiquent avec la mémoire et, par conséquent, les performances et les applications des ordinateurs modernes.

L'architecture de Von Neumann

L'**architecture de Von Neumann**, nommée d'après le mathématicien John von Neumann qui l'a proposée en 1945, est le modèle le plus couramment utilisé dans les ordinateurs actuels. Ce modèle repose sur une idée simple : un **bus unique** pour transférer à la fois les données et les instructions entre le **processeur** (CPU), la **mémoire** et les **périphériques**. Cela signifie que les instructions du programme et les données qu'il traite sont stockées dans une seule et même mémoire.

Dans cette architecture, le **CPU** récupère les instructions et les données dans une mémoire partagée, en suivant un processus cyclique appelé **cycle machine**. Ce cycle se déroule en plusieurs étapes : d'abord, le processeur lit une instruction, puis il l'exécute, et ensuite, il peut lire ou écrire des données en fonction de l'instruction.

Bien que l'architecture de Von Neumann soit simple et efficace, elle présente une limitation majeure, connue sous le nom de **goulot d'étranglement de Von Neumann**.

Ce goulot d'étranglement se produit parce que le bus unique est utilisé à la fois pour transférer des instructions et des données, ce qui signifie que le processeur ne peut pas lire des instructions et des données simultanément. Le CPU doit donc attendre que les informations soient disponibles, ce qui peut entraîner des **baisse de performances**.

En dépit de cette limitation, l'architecture de Von Neumann reste largement utilisée dans la plupart des ordinateurs grand public, car elle est relativement simple à implémenter et offre une flexibilité suffisante pour de nombreuses applications. Elle est particulièrement adaptée aux **systèmes généralistes**, comme les ordinateurs personnels et les serveurs.

L'architecture Harvard

En réponse à certaines des limitations de l'architecture de Von Neumann, l'**architecture Harvard** a été développée. Cette architecture se distingue par la **séparation physique** entre la **mémoire de programme** (qui contient les instructions) et la **mémoire de données**. Chaque mémoire possède son propre **bus**, ce qui permet au processeur de lire simultanément des instructions et des données.

Cette séparation apporte deux avantages principaux :

1. **Performances accrues** : Grâce à des bus distincts, le processeur peut accéder simultanément aux instructions et aux données, éliminant ainsi le goulot d'étranglement observé dans l'architecture de Von Neumann. Cela permet d'exécuter les programmes plus rapidement, car les cycles de lecture et d'écriture des données peuvent être effectués en parallèle avec la lecture des instructions.
2. **Sécurité renforcée** : La séparation des mémoires permet également une meilleure protection contre certaines attaques, comme les **attaques par injection de code**. Dans une architecture Harvard, les programmes et les données sont physiquement séparés, ce qui réduit les risques que des données malveillantes soient interprétées comme des instructions. Cela rend cette architecture particulièrement intéressante dans les systèmes embarqués ou dans les dispositifs nécessitant un haut niveau de sécurité, comme les systèmes industriels et les applications critiques.

Comparaison et application des deux architectures

L'architecture de Von Neumann est encore dominante dans les **ordinateurs de bureau**, les **serveurs** et d'autres systèmes informatiques généralistes, en raison de sa simplicité et de sa flexibilité. En revanche, l'**architecture Harvard** est souvent utilisée

dans des systèmes spécialisés, tels que les **microcontrôleurs** et les **systèmes embarqués**, où les gains de performances et la sécurité sont primordiaux.

Il est intéressant de noter que de nombreux **processeurs modernes** intègrent des éléments des deux architectures, en adoptant une approche hybride. Par exemple, ils peuvent utiliser des caches séparés pour les instructions et les données (comme dans l'architecture Harvard), tout en ayant une mémoire unifiée à plus grande échelle (comme dans l'architecture de Von Neumann). Cela permet de combiner les avantages des deux modèles et de maximiser l'efficacité des processeurs dans des environnements variés.

Le processus de démarrage de l'ordinateur

Le démarrage d'un ordinateur, ou **boot sequence**, est une série d'étapes qui permettent à la machine de passer de l'état hors tension à un état pleinement fonctionnel. Une fois tous les composants d'un ordinateur connectés correctement aux bus, le processus de démarrage peut commencer en alimentant la carte mère.

Étape 1 : Mise sous tension et RESET

Le démarrage commence par l'**alimentation de la carte mère**. Lorsque l'ordinateur est allumé, l'ensemble des composants matériels, y compris le **processeur** (CPU), la mémoire centrale (RAM) et les périphériques, sont alimentés en électricité. À ce moment-là, le CPU, qui était en mode "arrêt", passe en mode "marche". Ce changement d'état est appelé **RESET**.

Le **RESET** est un signal qui réinitialise le CPU et d'autres composants critiques de l'ordinateur. Il peut être déclenché de différentes manières : soit par la mise sous tension de l'ordinateur, soit par l'activation d'un bouton dédié sur la carte mère. Ce signal met tous les composants dans un état initial et indique au CPU de commencer à exécuter les premières instructions.

Étape 2 : Lecture de l'adresse initiale en mémoire ROM

Une fois le signal de RESET envoyé, le CPU doit déterminer où commencer l'exécution des instructions. La première étape consiste pour le processeur à lire une **adresse particulière** stockée dans la **mémoire ROM** (Read-Only Memory), une mémoire non volatile qui conserve son contenu même lorsque l'ordinateur est éteint.

Cette adresse initiale est fixe et prédéfinie par le fabricant de la carte mère. Elle contient les premières instructions que le processeur doit exécuter pour initialiser le système. Ces instructions résident dans une partie de la ROM appelée le **BIOS** (Basic

Input/Output System). Le rôle du BIOS est essentiel car il gère l'initialisation matérielle et prépare l'ordinateur à démarrer le système d'exploitation.

Étape 3 : Exécution des premières instructions

Une fois l'adresse initiale lue en ROM, le CPU commence à exécuter les **premières instructions** stockées à cet emplacement. Ces instructions sont généralement celles du BIOS, qui est responsable d'effectuer plusieurs tests et configurations de base, souvent appelés **POST** (Power-On Self Test). Ces tests vérifient que tous les composants matériels essentiels (comme la mémoire vive, les disques durs et les périphériques) sont correctement connectés et fonctionnent.

Le processeur récupère ces premières instructions via le **bus d'adresse**, en envoyant un signal sur le **bus de contrôle** pour indiquer qu'il souhaite lire l'instruction à l'adresse donnée. Ensuite, la ROM envoie l'instruction au CPU via le **bus de données**, où le processeur la lit et l'exécute.

Le **cycle d'instruction** du CPU, qui consiste à lire, décoder et exécuter des instructions, se répète constamment lors du processus de démarrage. Chaque nouvelle instruction est lue à une adresse mémoire, décodée par le CPU pour comprendre son rôle, puis exécutée. Ce cycle est fondamental pour l'opération continue de l'ordinateur.

Étape 4 : Initialisation du matériel

Pendant le processus d'exécution des premières instructions, le BIOS effectue plusieurs tâches importantes. En plus du POST, le BIOS **configure les périphériques d'entrée/sortie (I/O)** comme le clavier, la souris et les disques durs. Il détermine également la **séquence de démarrage**, c'est-à-dire l'ordre dans lequel les périphériques de stockage (disque dur, clé USB, etc.) seront vérifiés pour charger le système d'exploitation.

Le BIOS gère aussi les réglages de base du matériel, comme les **fréquences d'horloge** du processeur et la configuration de la **mémoire RAM**. Une fois tous les composants initialisés et fonctionnels, le BIOS peut alors chercher un périphérique de stockage pour charger le **bootloader**, qui est un petit programme responsable de démarrer le système d'exploitation.

Étape 5 : Chargement du système d'exploitation

Après l'initialisation du matériel, le BIOS doit localiser le **système d'exploitation**. Il vérifie les périphériques de stockage dans l'ordre défini par la séquence de démarrage. Lorsqu'il trouve un périphérique contenant un système d'exploitation, il charge le

bootloader en mémoire RAM. Le bootloader est chargé de démarrer le système d'exploitation en mémoire et de passer le contrôle de l'ordinateur au système d'exploitation.

À ce stade, le rôle du BIOS est terminé et le **système d'exploitation** prend le relais. Une fois le système d'exploitation chargé, il initialise ses propres services et prépare l'ordinateur pour l'utilisation, comme la gestion des processus, des fichiers et des périphériques. L'ordinateur est maintenant prêt à effectuer toutes les opérations prévues par l'utilisateur.

Différences d'accès entre la mémoire centrale et la mémoire de masse

Une fois l'ordinateur démarré, il accède à plusieurs types de mémoires. La **mémoire centrale**, constituée principalement de la RAM, est utilisée pour stocker temporairement les données et instructions qui sont en cours de traitement. La RAM offre un accès rapide mais est volatile, c'est-à-dire que toutes les informations qu'elle contient sont perdues lorsque l'ordinateur est éteint.

En plus de la RAM, l'ordinateur utilise des dispositifs de **mémoire de masse**, comme les **disques durs** (HDD), les **SSD** (Solid-State Drive) et les **clés USB**. Ces mémoires permettent de stocker des quantités de données beaucoup plus importantes de manière non volatile, ce qui signifie que les informations sont conservées même lorsque l'ordinateur est éteint.

Cependant, l'accès à la mémoire de masse est plus lent que l'accès à la RAM, car cela nécessite de nombreuses opérations supplémentaires, notamment pour localiser physiquement les informations sur le disque. Cela explique pourquoi les systèmes d'exploitation et les applications sont d'abord chargés en RAM, où ils peuvent être exécutés beaucoup plus rapidement.

Les registres internes du CPU

En plus de la RAM et de la mémoire de masse, le CPU dispose de **registres internes**, qui constituent une mémoire encore plus rapide. Ces **registres à usage général** (GPRs pour General Purpose Registers) permettent de stocker temporairement des données immédiatement disponibles pour le CPU lors de l'exécution d'instructions.

Les registres sont utilisés pour traiter des données au niveau du processeur sans avoir à faire appel constamment à la RAM, ce qui optimise les performances. Cette mémoire ultra-rapide est essentielle pour des opérations comme l'arithmétique, le contrôle de flux et le stockage temporaire des adresses.