

## Chapitre1 : Architecture de base d'un ordinateur

Dans cette partie, nous décrivons rapidement l'architecture de base d'un ordinateur et les principes de son fonctionnement.

Un *ordinateur* est une machine de traitement de l'information. Il est capable d'acquérir de l'information, de la stocker, de la transformer en effectuant des traitements quelconques, puis de la restituer sous une autre forme. Le mot *informatique* vient de la contraction des mots *information* et *automatique*.

Nous appelons *information* tout ensemble de données. On distingue généralement différents types d'informations : textes, nombres, sons, images, etc., mais aussi les instructions composant un programme. Comme on l'a vu dans la première partie, toute information est manipulée sous forme *binnaire* (ou numérique) par l'ordinateur.

### 1.1 Principes de fonctionnement

Les deux principaux constituants d'un ordinateur sont la *mémoire principale* et le *processeur*. La mémoire principale (MP en abrégé) permet de stocker de l'information (programmes et données), tandis que le processeur exécute pas à pas les instructions composant les programmes.

#### 1.1.1- Notion de programme

Un programme est une suite d'instructions élémentaires, qui vont être exécutées dans l'ordre par le processeur. Ces instructions correspondent à des actions très simples, comme additionner deux nombres, lire ou écrire une case mémoire, etc. Chaque instruction est codifiée en mémoire sur quelques octets.

Le processeur est capable d'exécuter des programmes en *langage machine*, c'est à dire composés d'instructions très élémentaires suivant un codage précis. Chaque type de processeur est capable d'exécuter un certain ensemble d'instructions, son *jeu d'instructions*.

Pour écrire un programme en *langage machine*, il faut donc connaître les détails du fonctionnement du processeur qui va être utilisé.

#### 1.1.2- Le processeur

Le processeur est un circuit électronique complexe qui exécute chaque instruction très rapidement, en quelques *cycles d'horloges*. Toute l'activité de l'ordinateur est cadencée par une horloge unique, de façon à ce que tous les circuits électroniques travaillent ensembles. La fréquence de cette horloge s'exprime en MHz (millions de battements par seconde). Par exemple, un ordinateur ``PC Pentium 133" possède un processeur de type Pentium et une horloge à 133 MHz.

Pour chaque instruction, le processeur effectue schématiquement les opérations suivantes :

1. lire en mémoire (MP) l'instruction à exécuter;
2. effectuer le traitement correspondant;

3. passer à l'instruction suivante.

Le processeur est divisé en deux parties (voir figure ), l'unité de commande et l'unité de traitement :

- l'unité de commande est responsable de la lecture en mémoire et du décodage des instructions;
- l'unité de traitement, aussi appelée *Unité Arithmétique et Logique (U.A.L.)*, exécute les instructions qui manipulent les données.

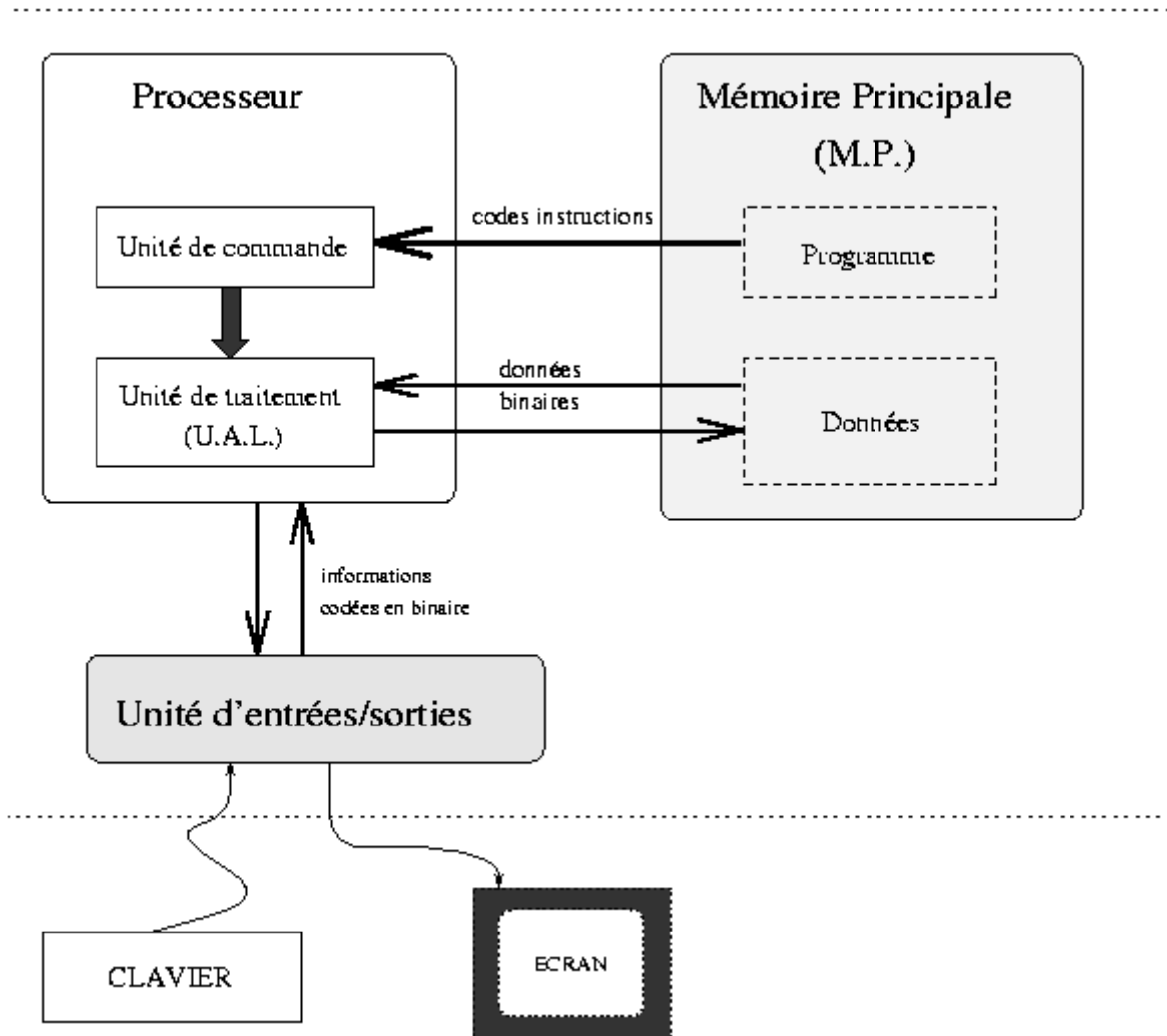


Figure 1.1: Architecture schématique d'un ordinateur.

## 1.2 La mémoire principale (MP)

### Structure de la MP



Chaque emplacement mémoire conserve les informations que le processeur y écrit jusqu'à coupure de l'alimentation électrique, où tout le contenu est perdu (contrairement au contenu des mémoires externes comme les disquettes et disques durs).

Les seules opérations possibles sur la mémoire sont :

- *écriture* d'un emplacement : le processeur donne une valeur et une adresse, et la mémoire range la valeur à l'emplacement indiqué par l'adresse;
- *lecture* d'un emplacement : le processeur demande à la mémoire la valeur contenue à l'emplacement dont il indique l'adresse. Le contenu de l'emplacement lu reste inchangé.

### Unité de transfert

Notons que les opérations de lecture et d'écriture portent en général sur plusieurs octets contigus en mémoire : un *mot* mémoire. La taille d'un mot mémoire dépend du type de processeur; elle est de

- 1 octet (8 bits) dans les processeurs 8 bits (par exemple Motorola 6502);
- 2 octets dans les processeurs 16 bits (par exemple Intel 8086);
- 4 octets dans les processeurs 32 bits (par ex. Intel 80486 ou Motorola 68030).

## 1.3 Le processeur central

Le processeur est parfois appelé CPU (de l'anglais *Central Processing Unit*) ou encore MPU (*Micro-Processing Unit*) pour les microprocesseurs .

Un *microprocesseur* n'est rien d'autre qu'un processeur dont tous les constituants sont réunis sur la même puce électronique (pastille de silicium), afin de réduire les coûts de fabrication et d'augmenter la vitesse de traitement. Les *microordinateurs* sont tous équipés de microprocesseurs.

L'architecture de base des processeurs équipant les gros ordinateurs est la même que celle des microprocesseurs.

### Les registres et l'accumulateur

Le processeur utilise toujours des *registres*, qui sont des petites mémoires internes très rapides d'accès utilisées pour stocker temporairement une donnée, une instruction ou une adresse. Chaque registre stocke 8, 16 ou 32 bits.

Le nombre exact de registres dépend du type de processeur et varie typiquement entre une dizaine et une centaine.

Parmi les registres, le plus important est le registre *accumulateur*, qui est utilisé pour stocker les résultats des opérations arithmétiques et logiques. L'accumulateur intervient dans une proportion importante des instructions.

Par exemple, examinons ce qu'il se passe lorsque le processeur exécute une instruction comme ``Ajouter 5 au contenu de la case memoire d'adresse 180" :

1. Le processeur lit et décode l'instruction;
2. le processeur demande à la mémoire la contenu de l'emplacement 180;
3. la valeur lue est rangée dans l'accumulateur;
4. l'unité de traitement (UAL ) ajoute 5 au contenu de l'accumulateur;
5. le contenu de l'accumulateur est écrit en mémoire à l'adresse 180.

C'est l'unité de commande (voir figure 1.1 page [pageref](#)) qui déclenche chacune de ces actions dans l'ordre. L'addition proprement dite est effectuée par l'UAL.

### Architecture d'un processeur à accumulateur

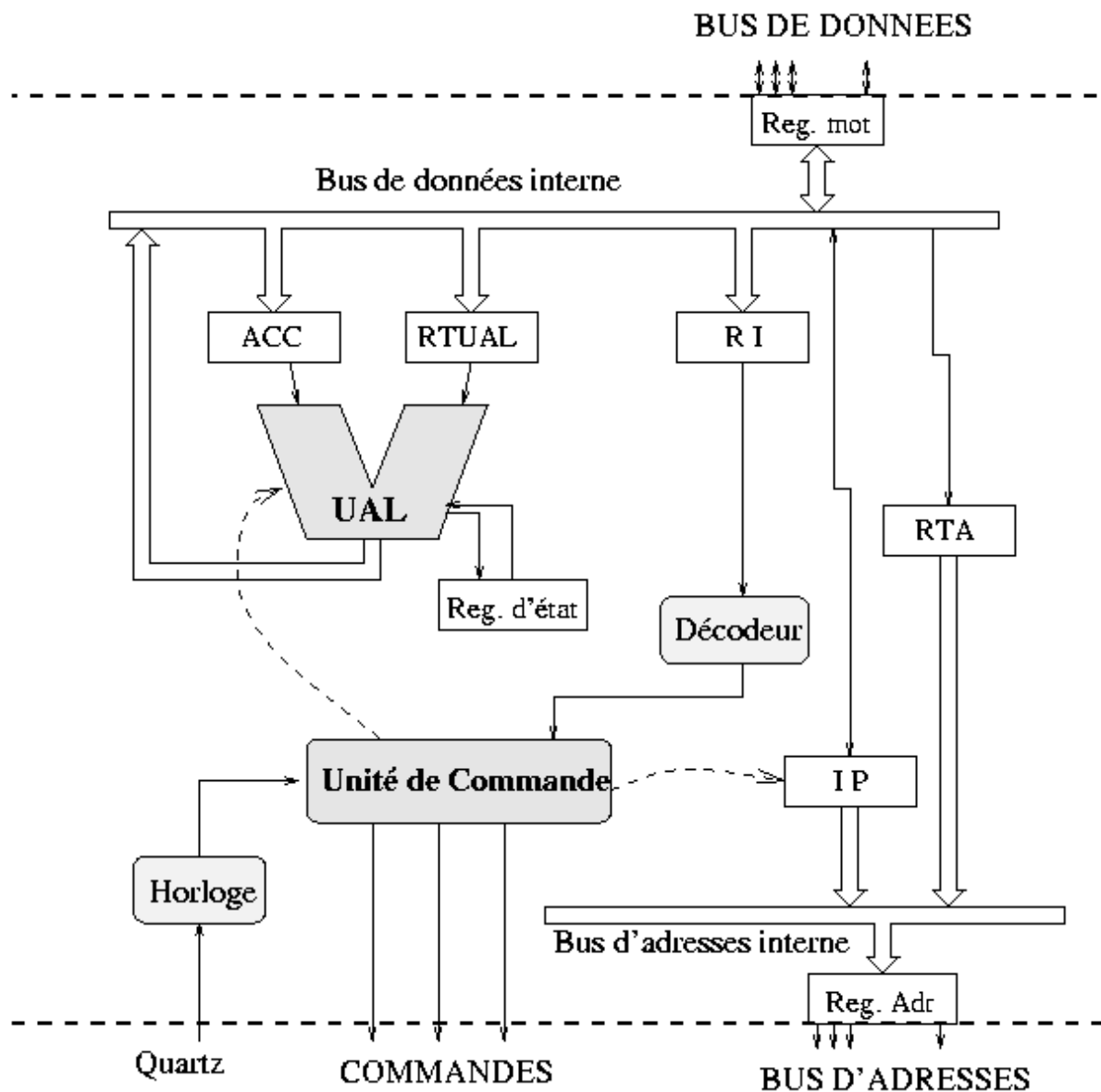


Figure 1.3: Schéma simplifié d'un processeur. Le processeur est relié à l'extérieur par les bus de données et d'adresses, le signal d'horloge et les signaux de commandes.

La figure 1.3 représente l'architecture interne simplifiée d'un MPU à accumulateur. On y distingue l'unité de commande, l'UAL, et le *décodeur* d'instructions, qui, à partir du code de l'instruction lu en mémoire actionne la partie de l'unité de commande nécessaire.

Les informations circulent à l'intérieur du processeur sur deux *bus internes*, l'un pour les données, l'autre pour les instructions.

On distingue les registres suivants :

**ACC**

: Accumulateur;

**RTUAL**

: Registre Tampon de l'UAL, stocke temporairement l'un des deux opérandes d'une instructions arithmétiques (la valeur 5 dans l'exemple donné plus haut);

**Reg. d'état**

: stocke les *indicateurs*, que nous étudierons plus tard;

**RI**

: Registre Instruction, contient le code de l'instruction en cours d'exécution (lu en mémoire via le bus de données);

**IP**

: *Instruction Pointer* ou Compteur de Programme, contient l'adresse de l'emplacement mémoire où se situe la prochaine instruction à exécuter;

**RTA**

: Registre Tampon d'Adresse, utilisé pour accéder à une donnée en mémoire.

Les signaux de commandes permettent au processeur de communiquer avec les autres circuits de l'ordinateur. On trouve en particulier le signal R/W (Read/Write), qui est utilisé pour indiquer à la mémoire principale si l'on effectue un accès en lecture ou en écriture.

### 1.3.4 Liaisons Processeur-Mémoire : les bus

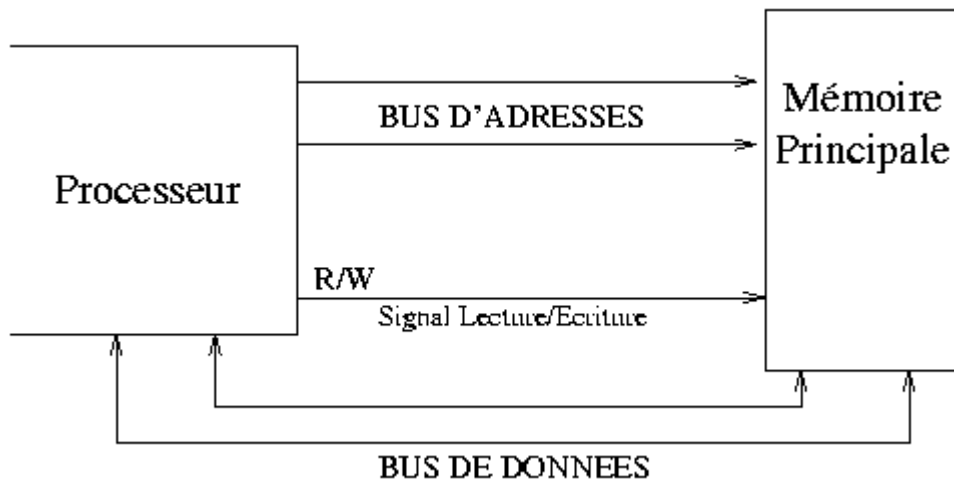


Figure 1.4: Connexions Processeur-Mémoire : bus de données, bus d'adresse et signal lecture/écriture.

Les informations échangées entre la mémoire et le processeur circulent sur des *bus*. Un *bus* est simplement un ensemble de  $n$  fils conducteurs, utilisés pour transporter  $n$  signaux binaires.

Le bus d'adresse est un bus unidirectionnel : seul le processeur envoie des adresses. Il est composé de  $a$  fils; on utilise donc des adresses de  $a$  bits. La mémoire peut posséder au maximum  $2^a$  emplacements (adresses 0 à  $2^a-1$ ).

Le bus de données est un bus bidirectionnel. Lors d'une lecture, c'est la mémoire qui envoie un mot sur le bus (le contenu de l'emplacement demandé); lors d'une écriture, c'est le processeur qui envoie la donnée.