

Architecture des ordinateurs

Chapitre 3:

Mémoires et Processeur

Chapitre 3 : Mémoires

1. Définitions

- Avec une bascule c'est possible de mémoriser une information sur 1 seul bit.
- Avec un registre c'est possible de mémoriser une information sur n bits.
- Si on veut mémoriser une information de taille importante \Rightarrow il faut utiliser une mémoire.

C'est quoi une **mémoire** ?

- Une mémoire est un dispositif capable :
 - d'enregistrer une information,
 - de la **conserver** (mémoriser)
 - et de la **restituer** (possible de la lire ou la récupérer par la suite). [6]
- Exemple de mémoire :
 - La mémoire centrale
 - Un disque dur
 - Une disquette
 - Une mémoire flash
 -
- La mémoire peut être dans le processeur (des registres), interne (Mémoire centrale ou principale) ou externe (Mémoire secondaire) [6].



- Une mémoire communique avec les autres composants de l'ordinateur à travers :
 - Les entrées d'adresses
 - Les entrées de données
 - Les sorties de données
 - Les entrées de commandes :
 - Une entrée de sélection de lecture ou d'écriture (R/W)
 - Une entrée de sélection du circuit (CS)

2. Caractéristiques

a. Capacité d'une mémoire

La capacité (taille) d'une mémoire est le nombre (quantité) d'informations qu'on peut enregistrer (mémoriser) dans cette mémoire.

La capacité peut s'exprimer en :

- Bit : un bit est l'élément de base pour la représentation de l'information.
- Octet : 1 Octet = 8 bits
- Kilo-octet (Ko) : 1 Ko = 1024 octets = 2^{10} octets
- Méga-octet (Mo) : 1 Mo = 1024 Ko = 2^{20} octets
- Giga-octet (Go) : 1 Go = 1024 Mo = 2^{30} octets
- Téra-octet (To) : 1 To = 1024 Go = 2^{40} octets

b. Volatilité

Si une mémoire perd son contenu (les informations) lorsque la source d'alimentation est coupée alors la mémoire est dite **volatile**.

Si une mémoire ne perd pas (conserve) son contenu lorsque la source d'alimentation est coupée alors la mémoire est dite non volatile (mémoire permanente ou stable).

c. Mode d'accès à l'information

Sur une mémoire on peut effectuer une opération de :

- **Lecture** : récupérer / restituer une information à partir de la mémoire.
- **Écriture** : enregistrer une nouvelle information ou modifier une information déjà existante dans la mémoire.

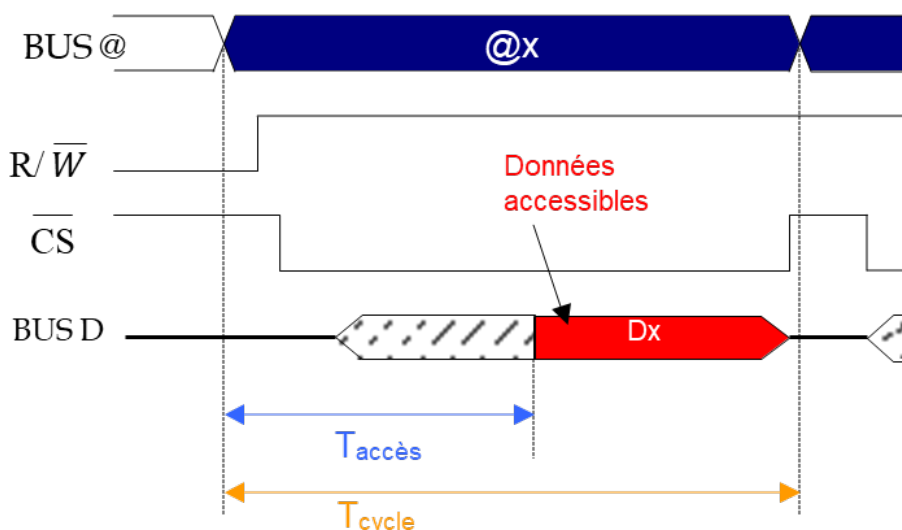
Il existe des mémoires qui offrent les deux modes lecture/écriture, ces mémoires s'appellent mémoires vives.

Il existe des mémoires qui offrent uniquement la possibilité de la lecture (ce n'est pas possible de modifier le contenu). Ces mémoires s'appellent mémoires mortes.

d. Temps d'accès

C'est le temps nécessaire pour effectuer une opération de lecture ou d'écriture.

Par exemple pour l'opération de lecture, le temps d'accès est le temps qui sépare la demande de la lecture de la disponibilité de l'information. [6]



Le temps d'accès est un critère important pour déterminer les performances d'une mémoire ainsi que les performances d'une machine.

3. Types de mémoires

a. Les registres

Ce sont les éléments de mémoire les plus rapides. Ils sont situés au niveau du processeur et servent au stockage des opérandes et des résultats intermédiaires.

b. La mémoire cache

C'est une mémoire rapide de faible capacité destinée à accélérer l'accès à la mémoire centrale en stockant les données les plus utilisées.

c. La mémoire principale

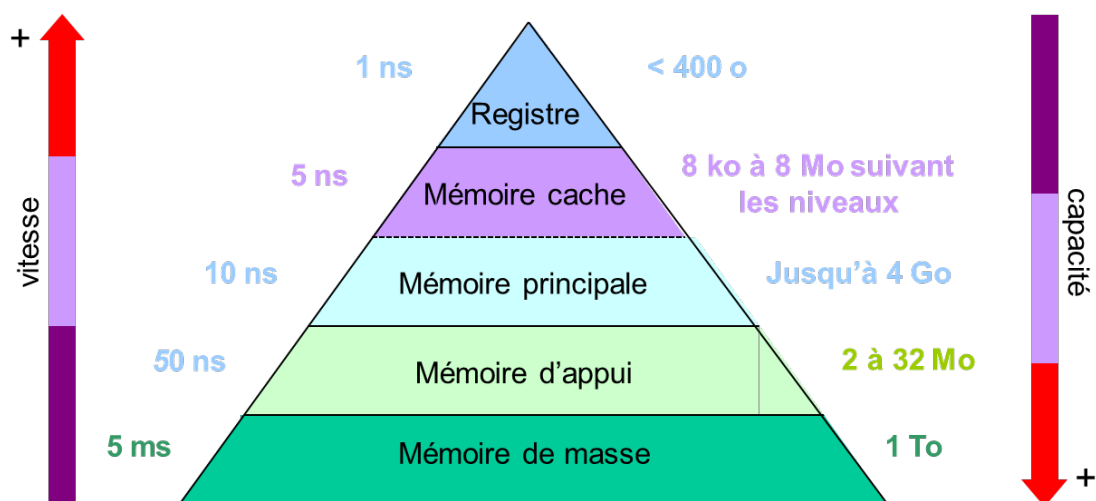
C'est l'organe principal de rangement des informations. Elle contient les programmes (instructions et données) et est plus lente que les deux mémoires précédentes.

d. La mémoire d'appui

Elle sert de mémoire intermédiaire entre la mémoire centrale et les mémoires de masse. Elle joue le même rôle que la mémoire cache.

e. La mémoire de masse

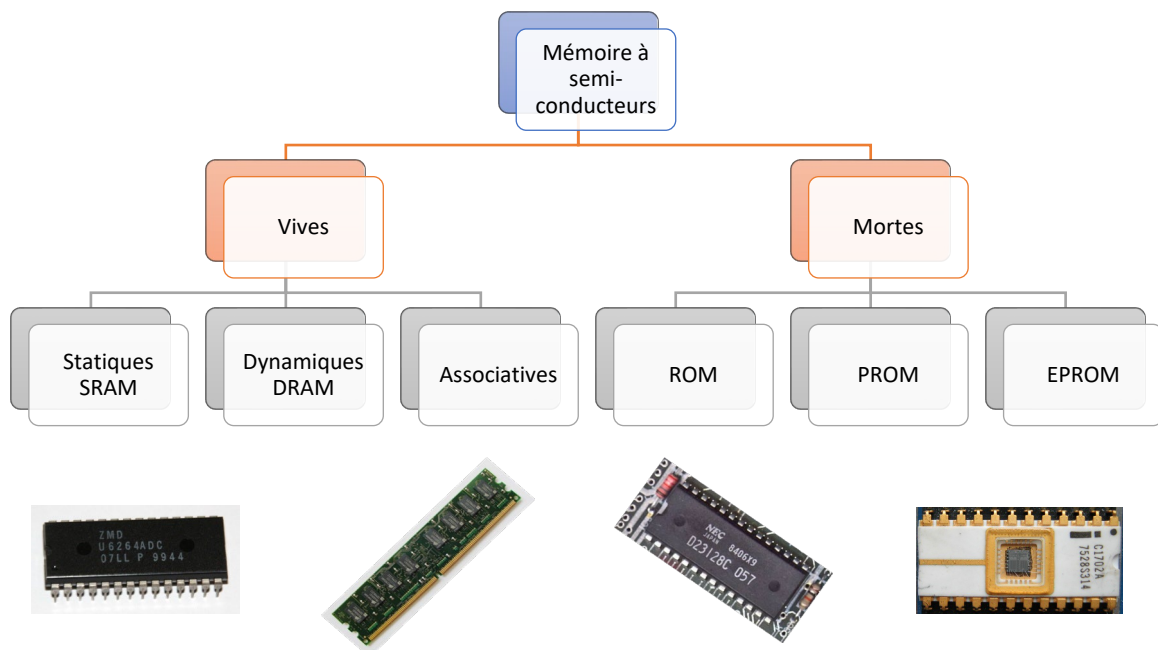
C'est une mémoire périphérique de grande capacité utilisée pour le stockage permanent ou la sauvegarde des informations. Elle utilise pour cela des supports magnétiques (disque dur, ZIP) ou optiques (CDROM, DVDROM) .[6]



4. Classification des mémoires

Les mémoires peuvent être classées en trois catégories selon la technologie utilisée :

- Mémoire à semi-conducteur (mémoire centrale, ROM, PROM, ...) : très rapide mais de taille réduite.
- Mémoire magnétique (disque dur, disquette, ...) : moins rapide mais stocke un volume d'informations très grand.
- Mémoire optique (DVD, CDROM, ...)



5. Mémoire centrale

La mémoire centrale (MC) représente l'espace de travail de l'ordinateur (calculateur). C'est l'organe principal de rangement des informations utilisées par le processeur.

Dans une machine (ordinateur/calculateur) pour exécuter un programme il faut le charger (copier) dans la mémoire centrale. Le temps d'accès à la mémoire centrale et sa capacité sont deux éléments qui influent sur le temps d'exécution d'un programme (performance d'une machine).

La mémoire centrale est une mémoire vive (accès en lecture et écriture), réalisée à base de semi-conducteurs. Elle est dite à accès aléatoire (RAM : Random Access Memory) c'est-à-dire que le temps d'accès à l'information est indépendant de sa place en mémoire. Un temps d'accès à une mémoire centrale est moyen mais plus rapide que les mémoires magnétiques.

La mémoire centrale est volatile : la conservation de son contenu nécessite la permanence de son alimentation électrique. La capacité d'une mémoire centrale est limitée mais il y a toujours une possibilité d'une extension.

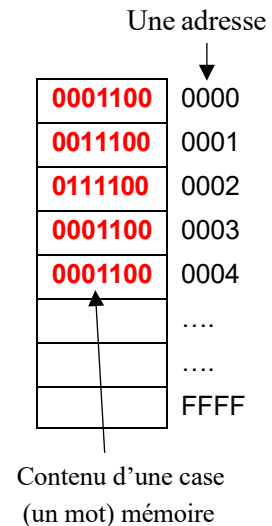
Pour la communication avec les autres organes de l'ordinateur, la mémoire centrale utilise les bus (bus d'adresses et bus de données)

Ils existent deux grandes familles des mémoires centrales : les mémoires statiques (SRAM) et les mémoires dynamiques (DRAM).

- Les mémoires statiques sont à base de bascules de type D, elles possèdent un faible taux d'intégration mais un temps d'accès rapide (Utilisation pour les mémoires cache).
- Les mémoires dynamiques à base de condensateurs, ces mémoires possèdent un très grand taux d'intégration, elles sont plus simples que les mémoires statiques mais avec un temps d'accès plus long.

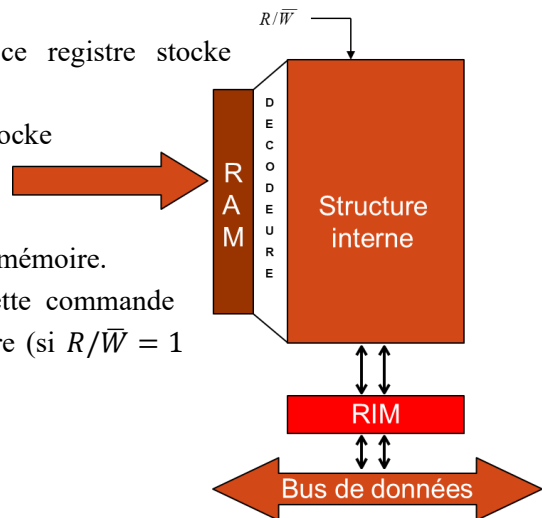
a. Vue logique de la mémoire centrale

- La mémoire centrale peut être vue comme un large **vecteur** (tableau) de mots ou octets.
- Un **mot mémoire** stocke une information sur n bits.
- Un mot mémoire contient plusieurs cellules mémoire.
- Une **cellule** mémoire stocke 1 seul bit.
- Chaque mot possède sa propre adresse.
- Une **adresse** est un numéro unique qui permet d'accéder à un mot mémoire.
- Les adresses sont séquentielles (consécutives).
- La taille de l'adresse (le nombre de bits) dépend de la capacité de la mémoire.



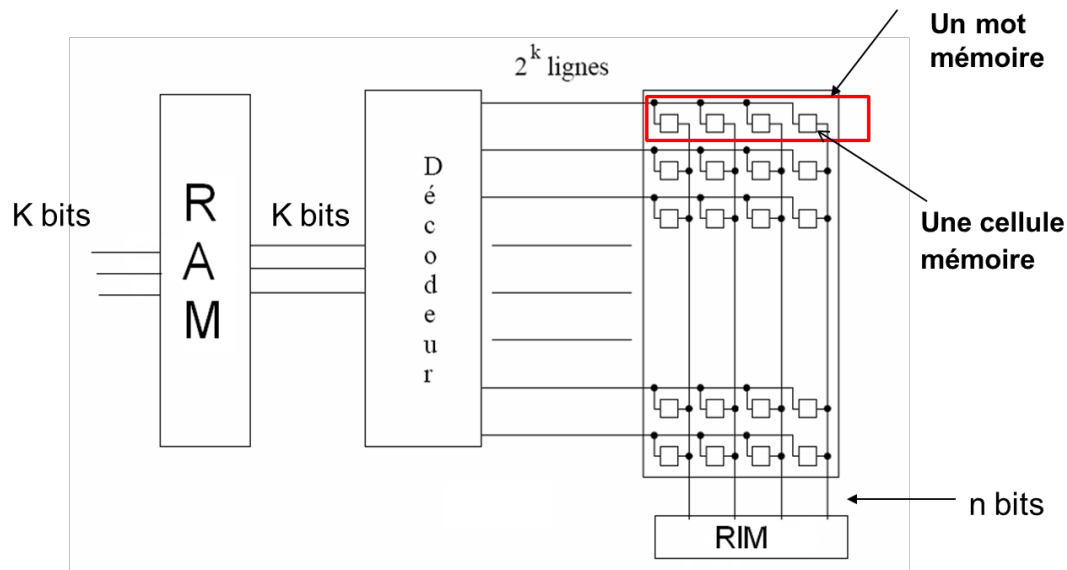
b. Structure physique d'une mémoire centrale

- **RAM** (Registre d'adresse Mémoire) : ce registre stocke l'adresse du mot à lire ou à écrire.
- **RIM** (Registre d'information mémoire) : stocke l'information lu à partir de la mémoire ou l'information à écrire dans la mémoire.
- **Décodeur** : permet de sélectionner un mot mémoire.
- R/\overline{W} : commande de lecture/écriture, cette commande permet de lire ou d'écrire dans la mémoire (si $R/\overline{W} = 1$ alors lecture sinon écriture).
- **Bus d'adresses** de taille k bits.
- **Bus de données** de taille n bits.



► Comment sélectionner un mot mémoire ?

Lorsqu'une adresse est chargée dans le registre *RAM*, le décodeur va recevoir la même information que celle du *RAM*. A la sortie du décodeur nous allons avoir une seule sortie qui est active. Cette sortie va nous permettre de sélectionner un seul mot mémoire.



► Comment calculer la capacité d'une MC ?

- Soit k la taille du bus d'adresses (taille du registre RAM).
 - Soit n la taille du bus de données (taille du registre RIM ou la taille d'un mot mémoire)
 - On peut exprimer la capacité de la mémoire centrale soit en nombre de mots mémoire ou en bits (octets, kilo-octets, ...)
- La capacité = 2^k Mots mémoire
 - La capacité = $2^k \times n$ Bits

► Exemple :

Dans une mémoire la taille du bus d'adresses $K=14$ et la taille du bus de données $n=4$.
Calculer la capacité de cette mémoire ?

$$C = 2^{14} = 16384 \text{ mots de 4 bits}$$

$$C = 2^{14} \times 4 = 65536 \text{ Bits} = 8192 \text{ Octets} = 8 \text{ Ko}$$

► Comment lire une information ?

Pour lire une information en mémoire centrale il faut effectuer les opérations suivantes :

- 1) Charger dans le registre *RAM* l'adresse du mot à lire.
- 2) Lancer la commande de lecture ($R/\bar{W} = 1$).
- 3) L'information est disponible dans le registre *RIM* au bout d'un certain temps (temps d'accès).

► Comment écrire une information ?

Pour écrire une information en MC il faut effectuer les opérations suivantes :

- 1) Charger dans le *RAM* l'adresse du mot où se fera l'écriture.
- 2) Placer dans le *RIM* l'information à écrire.
- 3) Lancer la commande d'écriture pour transférer le contenu du *RIM* dans la mémoire.

Chapitre 4 : Processeurs

Un MICROPROCESSEUR est un composant électronique minuscule, fabriqué le plus souvent en silicium, qui regroupe un certain nombre de transistors élémentaires interconnectés. Le microprocesseur exécute les fonctions d'unité centrale d'ordinateur (CPU), c'est-à-dire d'exécuter des instructions envoyées par un programme. [6]

Les principales caractéristiques d'un microprocesseur :

- Le **format** des mots de données : 8 bits, 16 bits, etc.
- La **taille** de l'espace adressable : dépend du nombre de bits d'adresses (ex: 65536 emplacements pour 16 bits).
- La **puissance** de traitement : s'exprime en MIPS (Millions d'Instructions Par Seconde)
- Le **jeu d'instructions** :
 - Etendu (CISC)
 - Réduit (RISC)

1. Microprocesseur

Le microprocesseur a été inventé par Marcian Hoff (surnommé Ted Hoff) en 1971, alors qu'il était ingénieur chez Intel. En 1990, Gilbert Hyatt a revendiqué la paternité du microprocesseur en se basant sur un brevet qu'il avait déposé en 1970. La reconnaissance de l'antériorité du brevet de Hyatt lui aurait permis de réclamer des redevances sur tous les microprocesseurs fabriqués de par le monde. [7]

Cependant, le brevet de Hyatt a été invalidé en 1995 par l'office américain des brevets sur la base du fait que le microprocesseur décrit dans la demande de brevet n'avait pas été réalisé et n'aurait d'ailleurs pas pu l'être avec la technologie disponible au moment du dépôt du brevet [8]. Il semble que Gilbert Hyatt n'ait pas abandonné et espère faire revoir cette décision.

Le premier microprocesseur commercialisé, le 15 novembre 1971, est l'Intel 4004 4-bits. Il fut suivi par l'Intel 8008. Ce microprocesseur a servi initialement à fabriquer des contrôleurs graphiques en mode texte, mais jugé trop lent par le client qui en avait demandé la conception, il devint un processeur d'usage général. Ces processeurs sont les précurseurs des Intel 8080, Zilog Z80, et de la future famille des Intel x86. [9]



Le microprocesseur exécute le programme, qui est une suite d'instructions. [10]

Une instruction est une opération SIMPLE sur un (ou plusieurs) mot(s) de données :

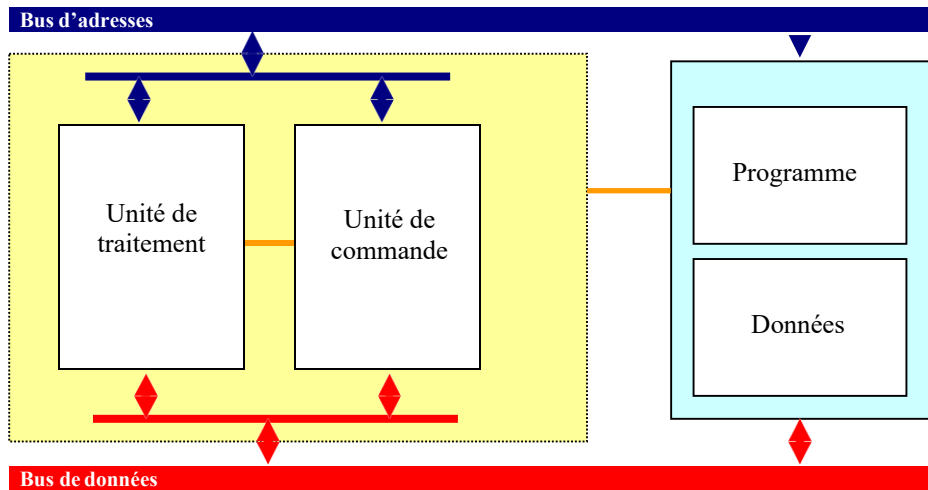
- Lecture (LOAD) ou Ecriture (STORE) en mémoire
- Opération logique (ET, OU, etc.)
- Opération arithmétique (addition, soustraction, etc.)

2. Architecture interne

Un microprocesseur est construit autour de deux éléments principaux :

- Une unité de commande.
- Une unité de traitement.

associés à des registres chargées de stocker les différentes informations à traiter. Ces trois éléments sont reliés entre eux par des bus interne permettant les échanges d'informations [6].



Il existe deux types de registres :

- Les registres d'usage général permettent à l'unité de traitement de manipuler des données à vitesse élevée. Ils sont connectés au bus des données interne au microprocesseur.
- Les registres d'adresses (pointeurs) connectés sur le bus adresses.

a. L'unité de commande

Elle permet de séquencer le déroulement des instructions. Elle effectue la recherche en mémoire de l'instruction. Comme chaque instruction est codée sous forme binaire, elle en assure le décodage pour enfin réaliser son exécution puis effectue la préparation de l'instruction suivante. Pour cela, elle est composée de :

- **Compteur de programme** : constitué par un registre dont le contenu est initialisé avec l'adresse de la première instruction du programme. Dès le lancement du programme ce compteur contient l'adresse de la première instruction à exécuter :
 - Soit par incrémentation automatique dans le cas où les adresses des instructions se suivent.
 - Soit par chargement de l'adresse de branchement dans le cas de sauts programmés.
- **Registre d'instruction et le décodeur d'instruction** : chacune des instructions à exécuter dont le format dépend de l'architecture du processeur est rangée dans le registre instruction puis est décodée par le décodeur d'instruction. Le premier mot est toujours le code de l'opération que le décodeur d'instruction doit identifier.
- **Bloc logique de commande (ou séquenceur)** : Il organise l'exécution des instructions au rythme d'une horloge. Il élabore tous les signaux de synchronisation internes ou externes (bus de commande) du microprocesseur en fonction des divers signaux de commande

provenant du décodeur d'instruction ou du registre d'état par exemple. Il s'agit d'un automate réalisé soit de façon câblée (obsolète), soit de façon micro-programmée, on parle alors de microprocesseur.

- **Registre d'adresses** : est un registre tampon qui assure l'interfaçage entre le microprocesseur et son environnement. Il conditionne le bus externe des adresses.
- **Registre de données** : est un registre tampon qui assure l'interfaçage entre le microprocesseur et son environnement ou inversement. Il conditionne le bus externe ou le bus interne des données

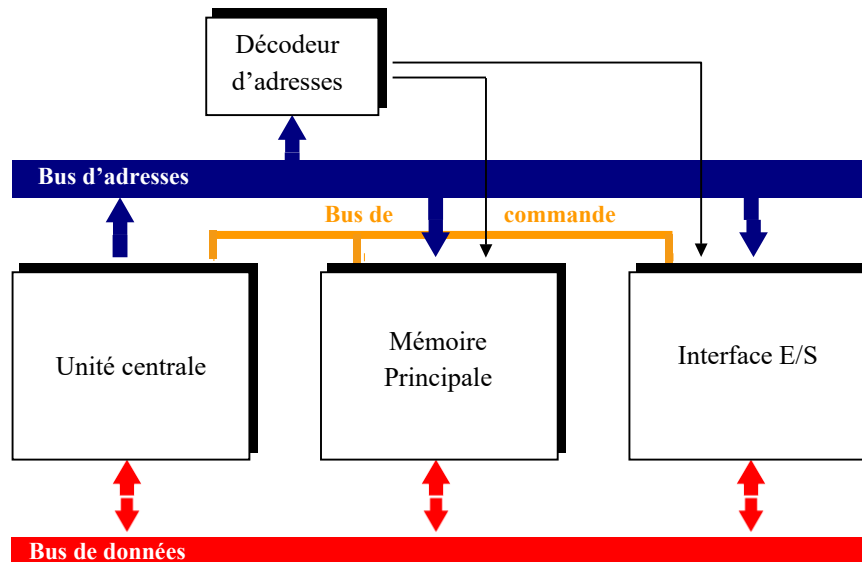
b. L'unité de traitement

C'est le cœur du microprocesseur. Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions :

- **L'Unité Arithmétique et Logique (UAL)** est un circuit complexe constituée par un certain nombre de circuits tels que : complémenteur, additionneur, décaleur, portes logiques, et qui assure les fonctions :
 - arithmétique (Addition, soustraction).
 - logiques (ET, OU, Comparaison, Décalage, etc...)
 - comparaison, décalage à droite ou à gauche, incrémentation, décrémentation, mise à 1 ou à 0 d'un bit, test de bit.
- **Le registre d'état** est généralement composé de 8 bits à considérer individuellement. Chacun de ces bits est un indicateur dont l'état dépend du résultat de la dernière opération effectuée par l'UAL. On les appelle indicateur d'état ou flag ou drapeaux. Dans un programme le résultat du test de leur état conditionne souvent le déroulement de la suite du programme. On peut citer par exemple les indicateurs de :
 - retenue (carry : C)
 - retenue intermédiaire (Auxiliary-Carry : AC)
 - signe (Sign : S)
 - débordement (overflow : OV ou V)
 - zéro (Z)
 - parité (Parity : P)
- **Les accumulateurs** sont des registres de travail qui servent à stocker un opérande au début d'une opération arithmétique et le résultat à la fin de l'opération. Ils permettent aussi de stocker temporairement des données en provenance de l'extérieur du microprocesseur avant leur reprise pour être rangées en mémoire ou des données provenant de la mémoire ou de l'UAL pour les présenter vers l'extérieur du microprocesseur.
- **Les registres auxiliaires** permettent de stocker le résultat des instructions exécutées par l'ALU.

c. Les bus de communication

Ils relient électriquement les composants internes du microprocesseur (bus internes) et relient les périphériques (mémoires et interfaces E/S) au processeur (bus externes).



On distingue 3 types de bus :

i. Le bus de données

Il est bidirectionnel et assure le transfert des informations entre le microprocesseur et son environnement, et inversement. Son nombre de lignes est égal au format des mots de données du microprocesseur.

ii. Le bus d'adresses

Il est unidirectionnel et permet la sélection des informations à traiter dans un espace mémoire (ou espace adressable) qui peut avoir 2^k emplacements, avec k = nombre de conducteurs du bus d'adresses.

iii. Le bus de commande (ou bus de contrôle)

Il est constitué par quelques conducteurs qui assurent la synchronisation des flux d'informations sur les bus de données et d'adresses.

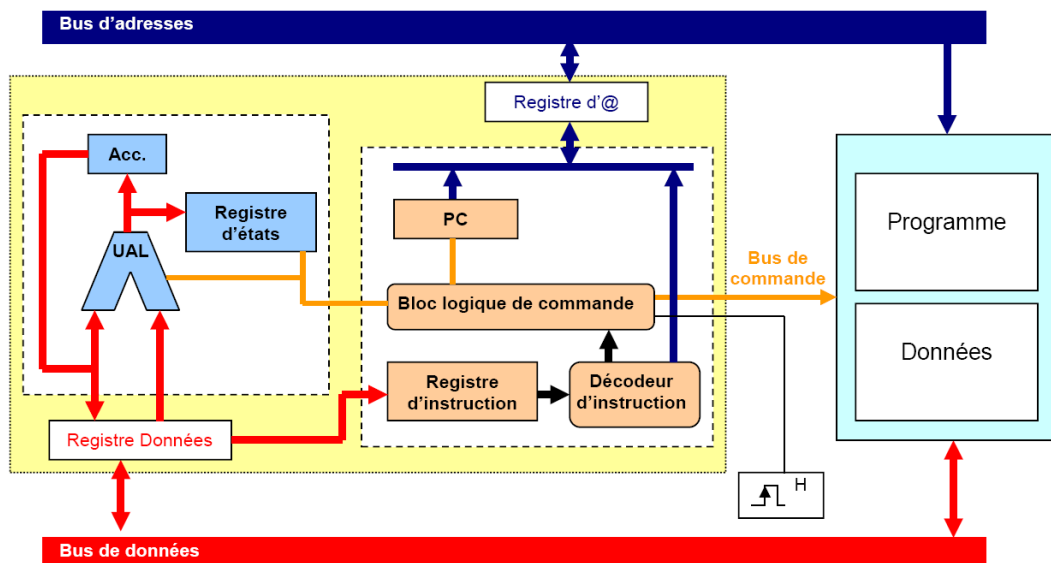


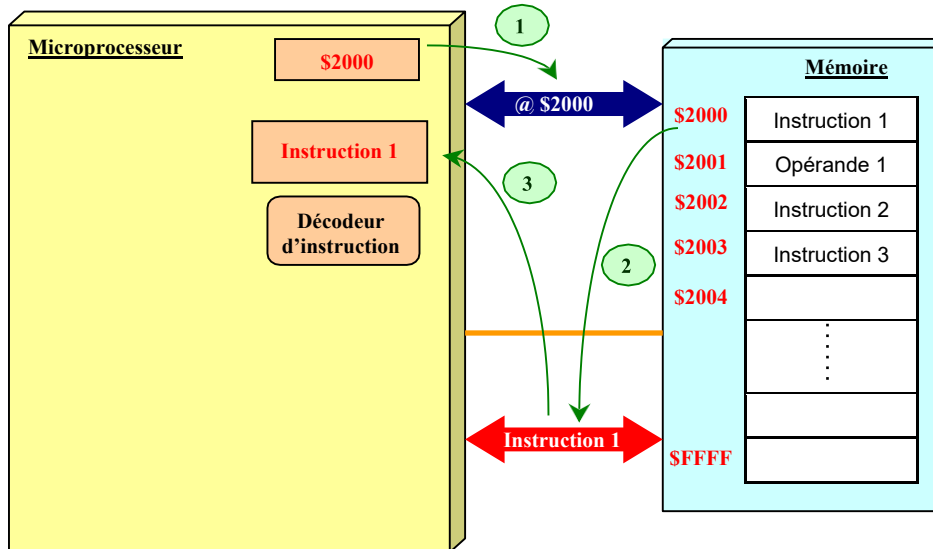
Figure 6 : Le schéma fonctionnel du microprocesseur

3. Cycle d'une instruction

Le microprocesseur ne comprend qu'un certain nombre d'instructions qui sont codées en binaire. Le traitement d'une instruction peut être décomposé en trois phases :

- **Phase 1 : Recherche de l'instruction à traiter**

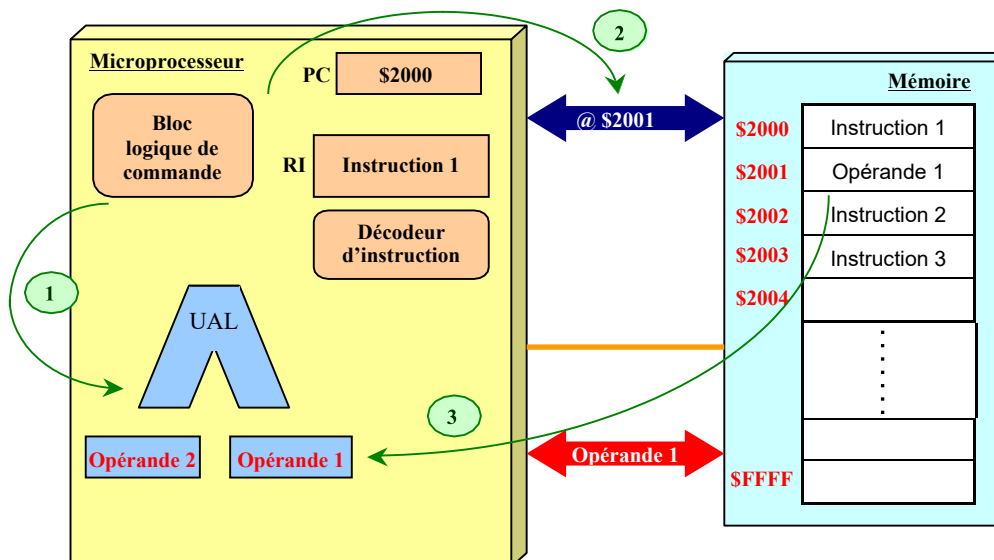
1. Le PC contient l'adresse de l'instruction suivante du programme. Cette valeur est placée sur le bus d'adresses par l'unité de commande qui émet un ordre de lecture.
2. Au bout d'un certain temps (temps d'accès à la mémoire), le contenu de la case mémoire sélectionnée est disponible sur le bus des données.
3. L'instruction est stockée dans le registre instruction du processeur.



- **Phase 2 : Décodage de l'instruction et recherche de l'opérande**

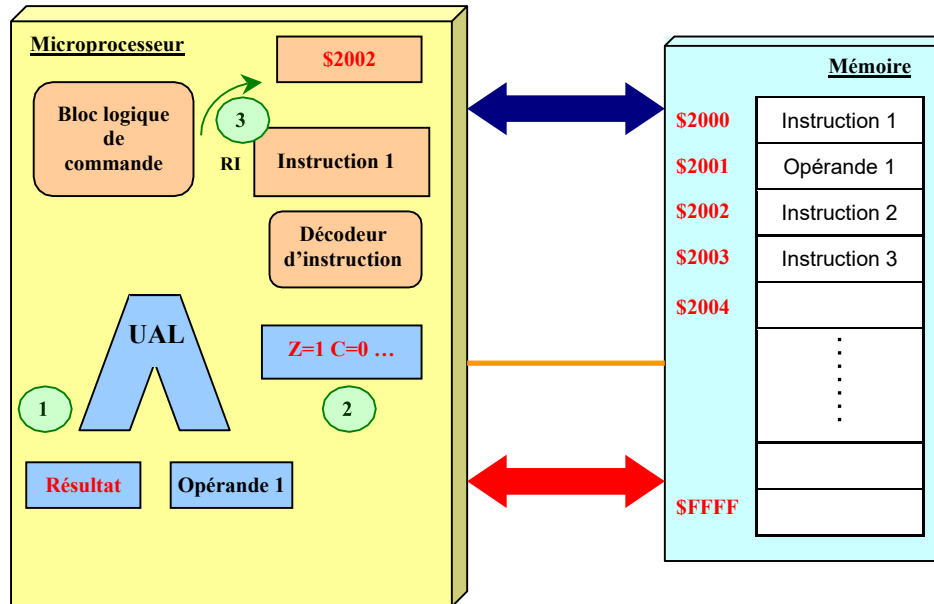
Le registre d'instruction contient maintenant le premier mot de l'instruction qui peut être codée sur plusieurs mots. Ce premier mot contient le code opératoire qui définit la nature de l'opération à effectuer (addition, rotation, ...) et le nombre de mots de l'instruction.

1. L'unité de commande transforme l'instruction en une suite de commandes élémentaires nécessaires au traitement de l'instruction.
2. Si l'instruction nécessite une donnée en provenance de la mémoire, l'unité de commande récupère sa valeur sur le bus de données.
3. L'opérande est stocké dans un registre.



Phase 3 : Exécution de l'instruction

1. Le séquenceur réalise l'instruction est exécuté.
2. Les drapeaux sont positionnés (registre d'état).
3. L'unité de commande positionne le PC pour l'instruction suivante.



4. Jeu d'instructions

a. Définition

La première étape de la conception d'un microprocesseur est la définition de son jeu d'instructions. Le jeu d'instructions décrit l'ensemble des opérations élémentaires que le microprocesseur pourra exécuter. Il va donc en partie déterminer l'architecture du microprocesseur à réaliser et notamment celle du séquenceur. A un même jeu d'instructions peut correspondre un grand nombre d'implémentations différentes du microprocesseur. [6]

b. Type d'instructions

Les instructions que l'on retrouve dans chaque microprocesseur peuvent être classées en 4 groupes :

- **Transfert de données** pour charger ou sauver en mémoire, effectuer des transferts de registre à registre, etc...
- **Opérations arithmétiques** : addition, soustraction, division, multiplication
- **Opérations logiques** : ET, OU, NON, NAND, comparaison, test, etc...
- **Contrôle de séquence** : branchement, test, etc...

c. Codage

Les instructions et leurs opérandes (paramètres) sont stockés en mémoire principale. La taille totale d'une instruction (nombre de bits nécessaires pour la représenter en mémoire) dépend du type d'instruction et aussi du type d'opérande. Chaque instruction est toujours codée sur un nombre entier d'octets afin de faciliter son décodage par le processeur. Une instruction est composée de deux champs :

- Le code instruction, qui indique au processeur quelle instruction réaliser
- Le champ opérande qui contient la donnée, ou la référence à une donnée en mémoire (son adresse).

► Exemple :

| Code instruction | Code Opérande |
|------------------|---------------|
| 1001 0011 | 0011 1110 |

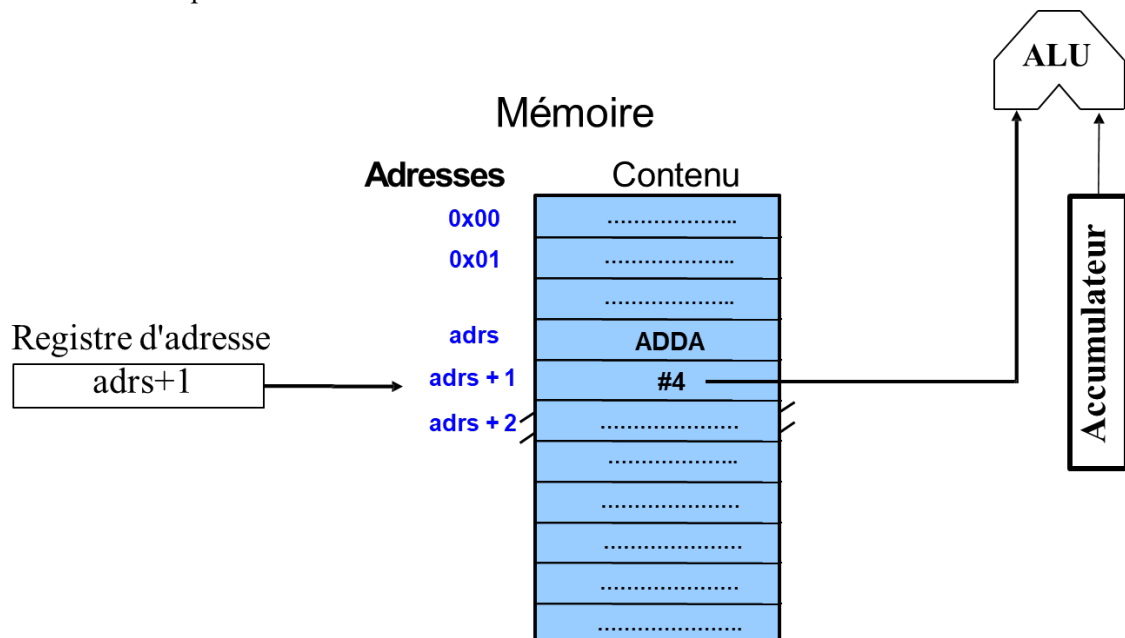
Le nombre d'instructions du jeu d'instructions est directement lié au format du code instruction. Ainsi un octet permet de distinguer au maximum 256 instructions différentes.

d. Mode d'adressage

Un mode d'adressage définit la manière dont le microprocesseur va accéder à l'opérande. Les différents modes d'adressage dépendent des microprocesseurs mais on retrouve en général :

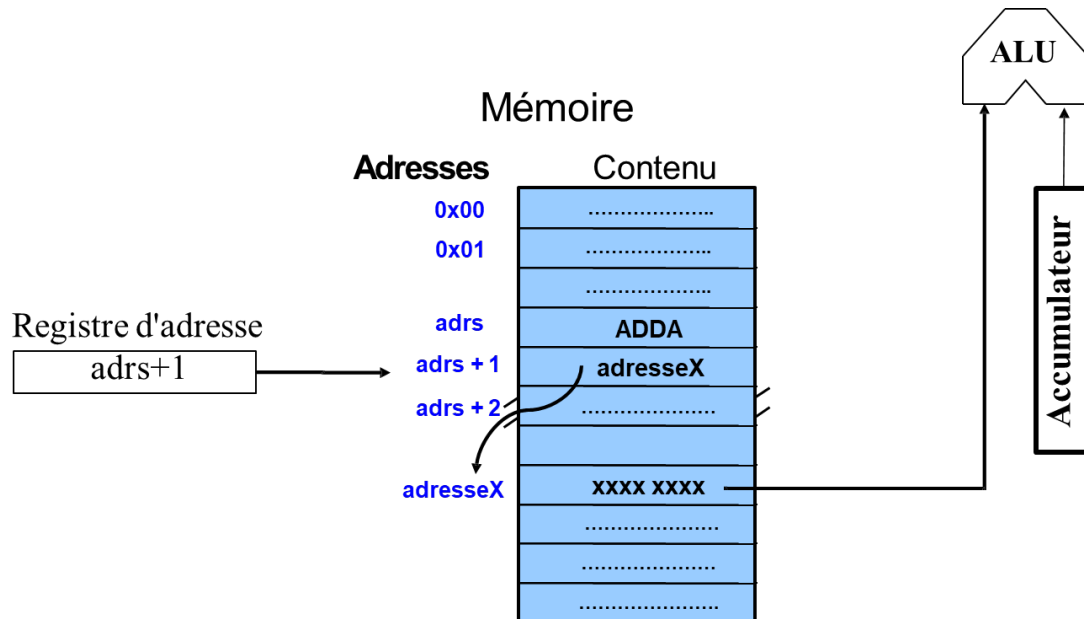
i. Adressage immédiat

► Exemple : ADDA #4



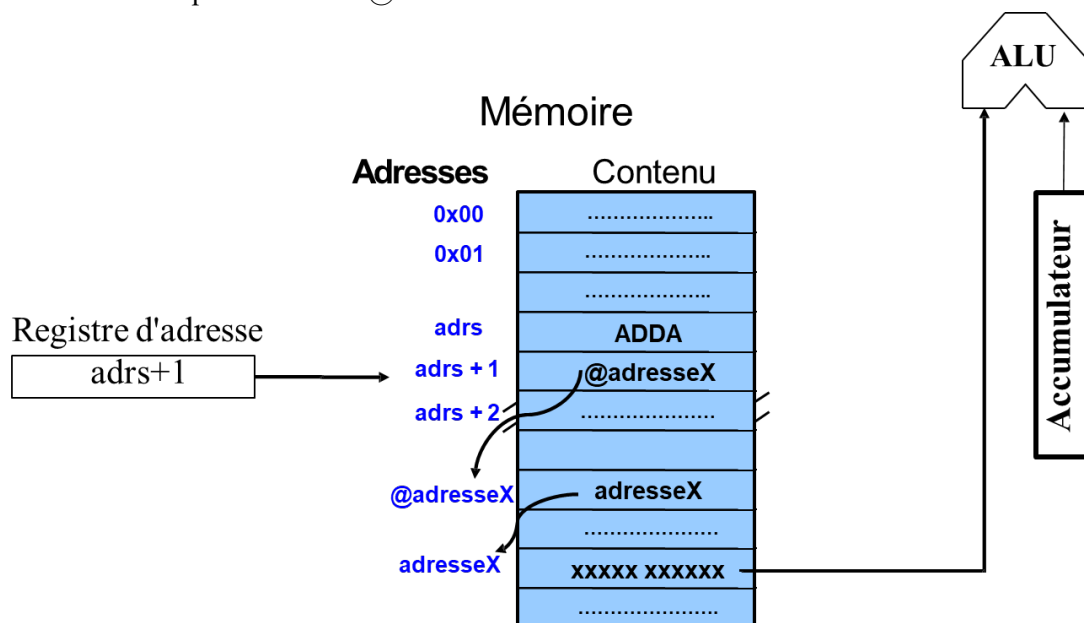
ii. Adressage direct

► Exemple : ADDA adresseX



iii. Adressage indirect

► Exemple : ADDA @adresseX



e. *Notion d'architecture RISC et CISC*

Actuellement l'architecture des microprocesseurs se compose de deux grandes familles :

- L'architecture **CISC** (*Complex Instruction Set Computer*)
- L'architecture **RISC** (*Reduced Instruction Set Computer*)

Le choix dépendra des applications visées. En effet, si on diminue le nombre d'instructions, on crée des instructions complexes (CISC) qui nécessitent plus de cycles pour être décodées et si on diminue le nombre de cycles par instruction, on crée des instructions simples (RISC) mais on augmente alors le nombre d'instructions nécessaires pour réaliser le même traitement.

| Architecture RISC | Architecture CISC |
|--|--|
| <ul style="list-style-type: none"> • Instructions simples ne prenant qu'un seul cycle | <ul style="list-style-type: none"> • Instructions complexes prenant plusieurs cycles |
| <ul style="list-style-type: none"> • Instructions au format fixe | <ul style="list-style-type: none"> • Instructions au format variable |
| <ul style="list-style-type: none"> • Décodeur simple (câblé) | <ul style="list-style-type: none"> • Décodeur complexe (microcode) |
| <ul style="list-style-type: none"> • Beaucoup de registres | <ul style="list-style-type: none"> • Peu de registres |
| <ul style="list-style-type: none"> • Seules les instructions LOAD et STORE ont accès à la mémoire | <ul style="list-style-type: none"> • Toutes les instructions sont susceptibles d'accéder à la mémoire |
| <ul style="list-style-type: none"> • Peu de modes d'adressage | <ul style="list-style-type: none"> • Beaucoup de modes d'adressage |
| <ul style="list-style-type: none"> • Compilateur complexe | <ul style="list-style-type: none"> • Compilateur simple |

Durant longtemps, les CISC et les RISC eurent chacun leurs admirateurs et leurs détracteurs. Mais au final, on ne peut pas dire qu'un processeur CISC sera meilleur qu'un RISC ou l'inverse : chacun a des avantages et des inconvénients, qui rendent le RISC/CISC adapté ou pas selon la situation.