

Ordinateurs de plus en plus puissants, basés sur des microprocesseurs performants : Pentium, Power PC, ...

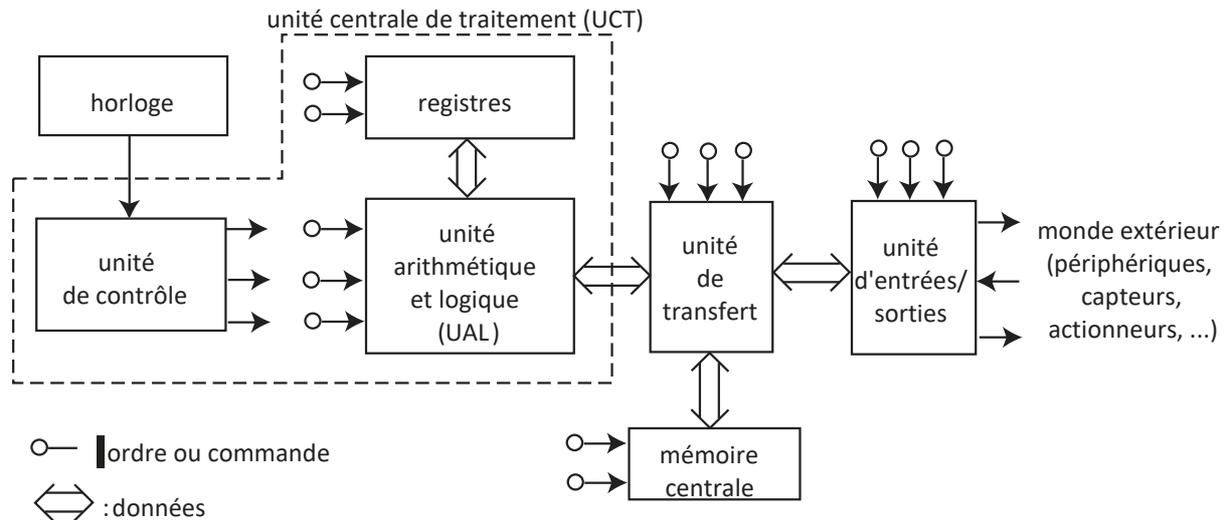
Nouvelles architectures de microprocesseurs : RISC.

Applications multimédia, réseaux, ...

Systèmes embarqués : microcontrôleurs, processeurs de traitement de signal (DSP), ...

Architecture et fonctionnement d'un microprocesseur

2.1 Structure d'un ordinateur



L'élément de base d'un ordinateur est constitué par l'**unité centrale de traitement** (UCT, CPU : Central Processing Unit). L'UCT est constituée :

- d'une **unité arithmétique et logique** (UAL, ALU : Arithmetic and Logic Unit) : c'est l'organe de calcul du ordinateur ;
- de **registres** : zones de stockage des données de travail de l'UAL (opérandes, résultats intermédiaires) ;
- d'une **unité de contrôle** (UC, CU : Control Unit) : elle envoie les ordres (ou commandes) à tous les autres éléments du ordinateur afin d'exécuter un **programme**.

La **mémoire centrale** contient :

- le programme à exécuter : suite d'instructions élémentaires ;
- les données à traiter.

L'**unité d'entrées/sorties** (E/S) est un intermédiaire entre le ordinateur et le monde extérieur.

L'**unité de transfert** est le support matériel de la circulation des données.

Les échanges d'ordres et de données dans le ordinateur sont synchronisés par une **horloge** qui délivre des impulsions (signal d'horloge) à des intervalles de temps fixes.

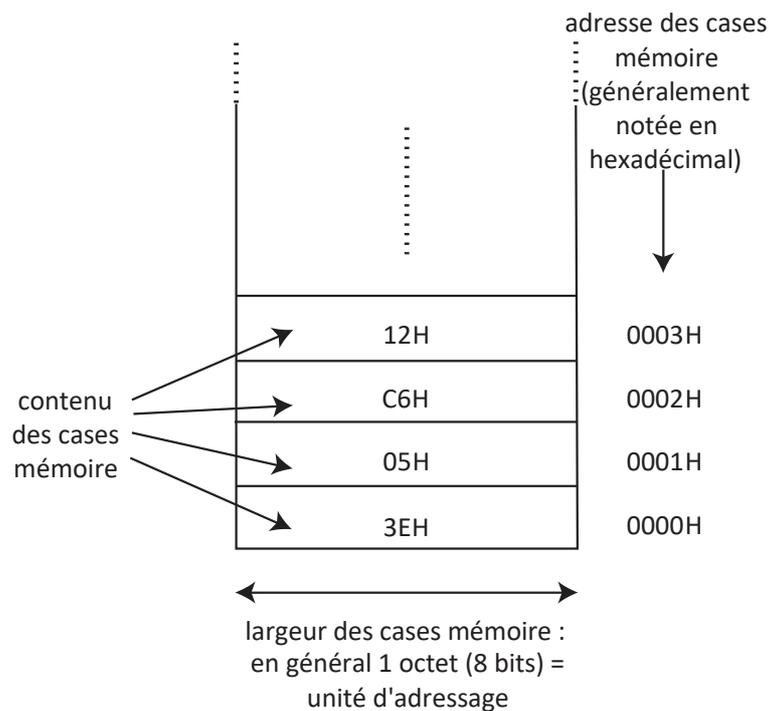
Définition : un microprocesseur consiste en une unité centrale de traitement (UAL + registres + unité de contrôle) entièrement contenue dans **un seul circuit intégré**. Un calculateur construit autour d'un microprocesseur est un **microcalculateur** ou un **microordinateur**.

Remarque : un circuit intégré qui inclut une UCT, de la mémoire et des périphériques est un **microcontrôleur**.

2.2 Organisation de la mémoire centrale

La mémoire peut être vue comme un ensemble de **cellules** ou **cases** contenant chacune une information : une instruction ou une donnée. Chaque case mémoire est repérée par un numéro d'ordre unique : son **adresse**.

Représentation :

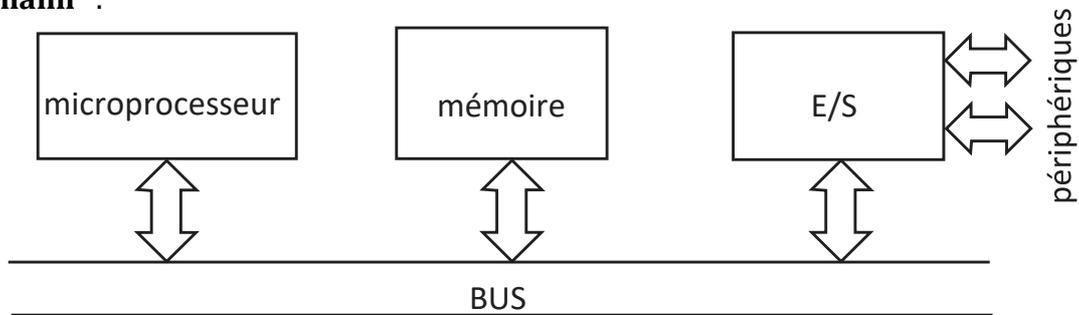


Une case mémoire peut être lue ou écrite par le microprocesseur (cas des **mémoires vives**) ou bien seulement lue (cas des **mémoires mortes**).

2.3 Circulation de l'information dans un ordinateur

La réalisation matérielle des ordinateurs est généralement basée sur l'architecture de **Von Neumann** :

Neumann :

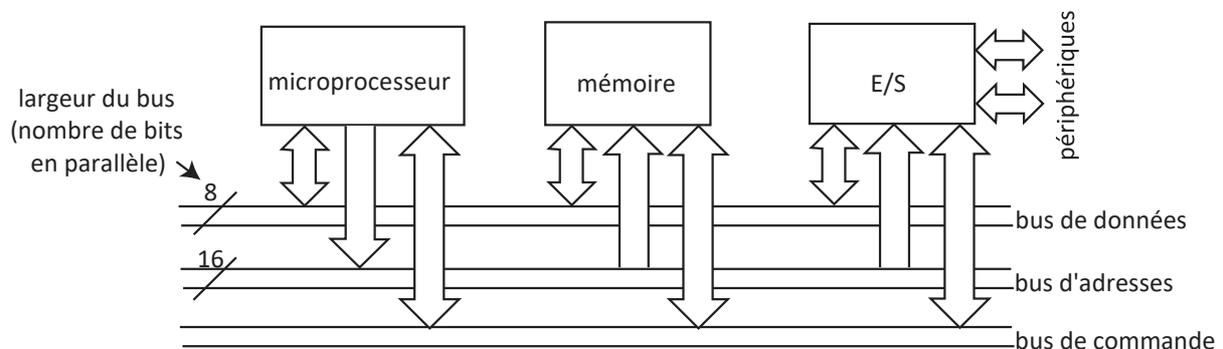


Le microprocesseur échange des informations avec la mémoire et l'unité d'E/S, sous forme de mots binaires, au moyen d'un ensemble de connexions appelé **bus**. Un bus permet de transférer des données sous forme **parallèle**, c'est-à-dire en faisant circuler n bits simultanément.

Les microprocesseurs peuvent être classés selon la longueur maximale des mots binaires qu'ils peuvent échanger avec la mémoire et les E/S : microprocesseurs 8 bits, 16 bits, 32 bits, ...

Le bus peut être décomposé en trois bus distincts :

- le **bus d'adresses** permet au microprocesseur de spécifier l'adresse de la case mémoire à lire ou à écrire ;
- le **bus de données** permet les transferts entre le microprocesseur et la mémoire ou les E/S;
- le **bus de commande** transmet les ordres de lecture et d'écriture de la mémoire et des E/S.



Remarque : les bus de données et de commande sont **bidirectionnels**, le bus d'adresse est **unidirectionnel** : seul le microprocesseur peut délivrer des adresses (il existe une d'exception pour les circuits d'accès direct à la mémoire, DMA).

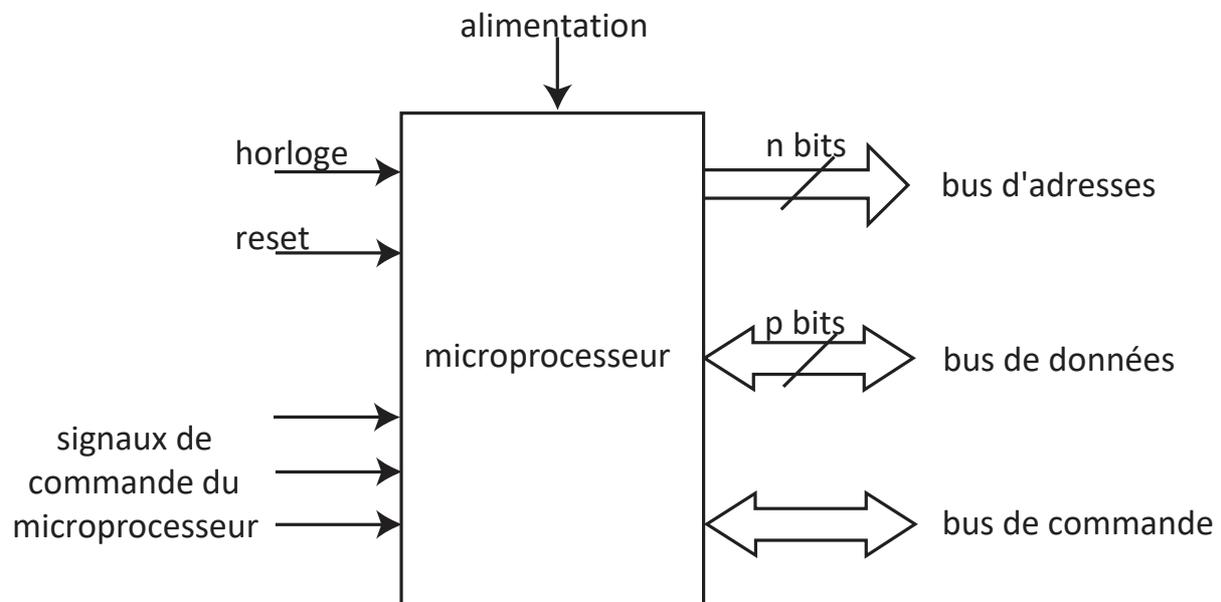
2.4 Description matérielle d'un microprocesseur

Un microprocesseur se présente sous la forme d'un circuit intégré muni d'un nombre généralement important de broches. Exemples :

- Intel 8085, 8086, Zilog Z80 : 40 broches, DIP (Dual In-line Package);
- Motorola 68000 : 64 broches, DIP;
- Intel 80386 : 196 broches, PGA (Pin Grid Array).

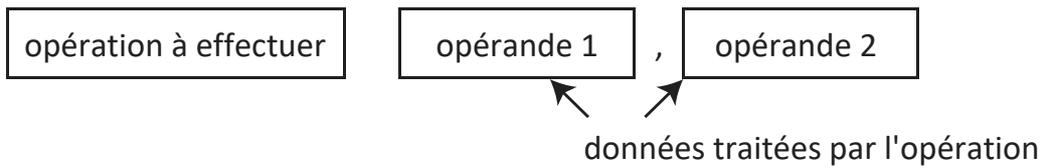
Technologies de fabrication : NMOS, PMOS, CMOS.

On peut représenter un microprocesseur par son **schéma fonctionnel** :



2.5 Fonctionnement d'un microprocesseur

Un microprocesseur exécute un **programme**. Le programme est une suite d'instructions stockées dans la mémoire. Une instruction peut être codée sur **un ou plusieurs octets**. Format d'une instruction :

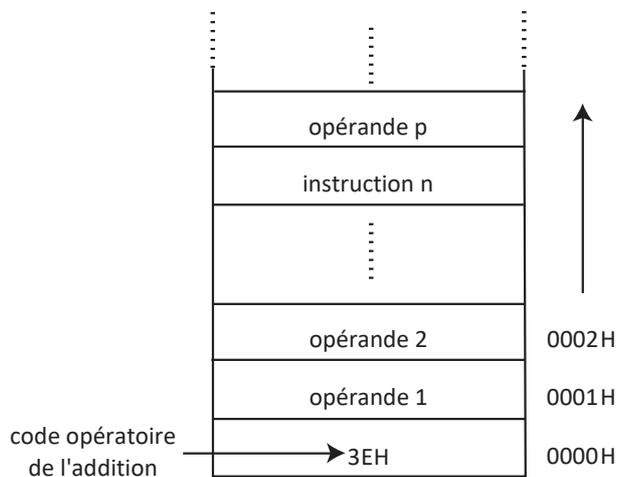


Exemple :

ADDITIONNER case mémoire 1, case mémoire 2
 opération opérandes

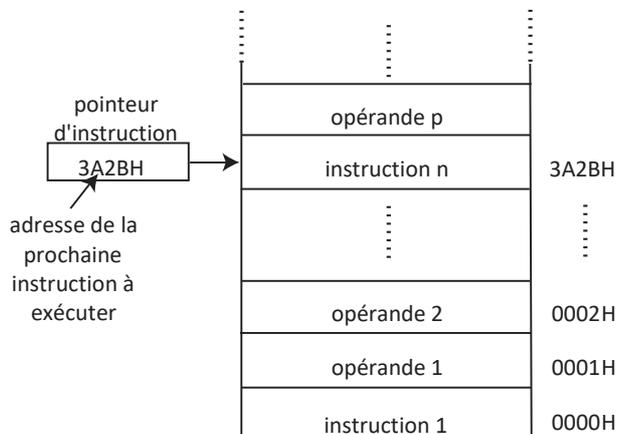
2.5 Fonctionnement d'un microprocesseur Rangement en

mémoire :



Pour exécuter les instructions dans l'ordre établi par le programme, le microprocesseur doit savoir à chaque instant l'adresse de la prochaine instruction à exécuter. Le microprocesseur utilise un registre contenant cette information. Ce registre est appelé **pointeur d'instruction** (IP : Instruction Pointer) ou **compteur d'instructions** ou **compteur ordinal**.

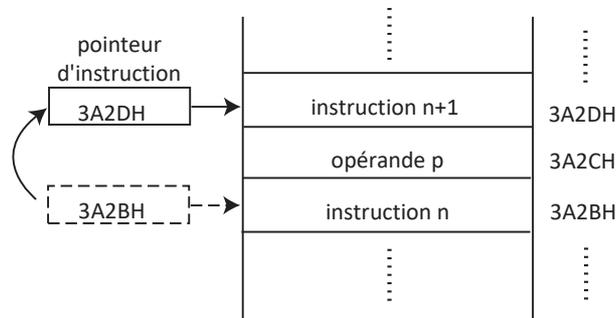
Exemple :



Remarque : la valeur initiale du pointeur d'instruction est fixée par le constructeur du microprocesseur. Elle vaut une valeur bien définie à chaque mise sous tension du microprocesseur ou bien lors d'une remise à zéro (reset).

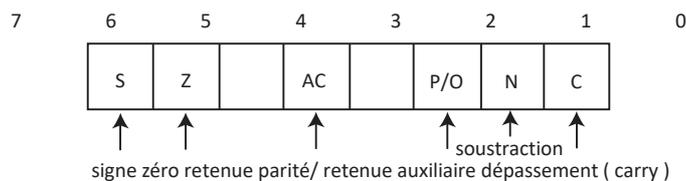
Pour savoir quel type d'opération doit être exécuté (addition, soustraction, ...), le microprocesseur lit le premier octet de l'instruction pointée par le pointeur d'instruction (code opératoire) et le range dans un registre appelé **registre d'instruction**. Le code opératoire est **décodé** par des circuits de décodage contenus dans le microprocesseur. Des signaux de commande pour l'UAL sont produits en fonction de l'opération demandée qui est alors exécutée.

Remarque : pour exécuter une instruction, l'UAL utilise des **registres de travail**, exemple : l'**accumulateur**, registre temporaire recevant des données intermédiaires. Pendant que l'instruction est décodée, le pointeur d'instruction est incrémenté de façon à pointer vers l'instruction suivante :



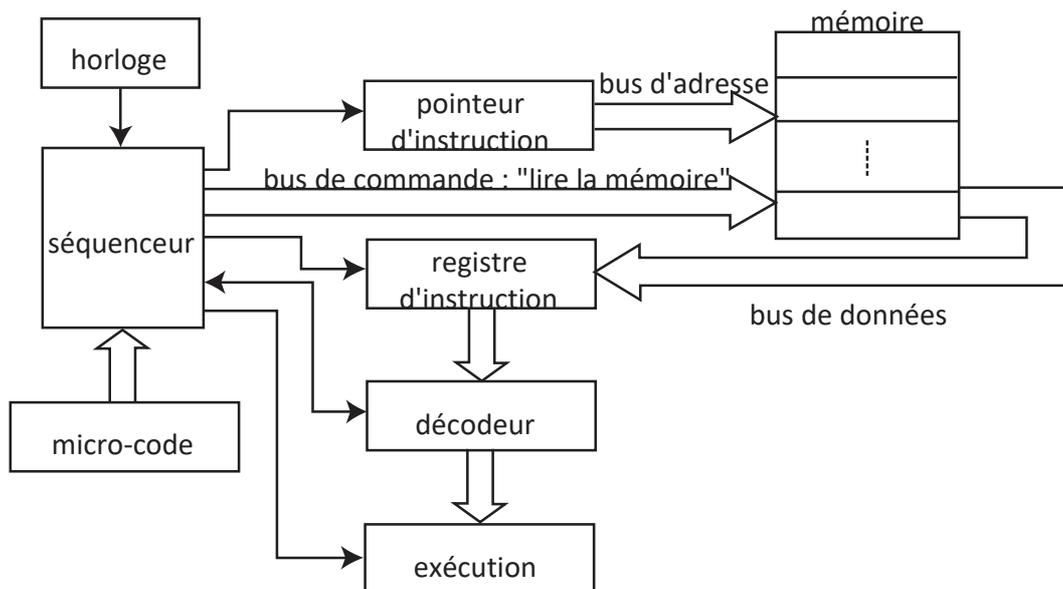
puis le processus de lecture et de décodage des instructions recommence.

A la suite de chaque instruction, un registre du microprocesseur est actualisé en fonction du dernier résultat : c'est le **registre d'état** du microprocesseur. Chacun des bits du registre d'état est un **indicateur d'état** ou **flag** (drapeau). Exemple : registre d'état du microprocesseur Z80 :



Les indicateurs d'état sont activés lorsqu'une certaine condition est remplie, exemple : le flag Z est mis à 1 lorsque la dernière opération a donné un résultat nul, le flag C est mis à un lorsque le résultat d'une addition possède une retenue, ...

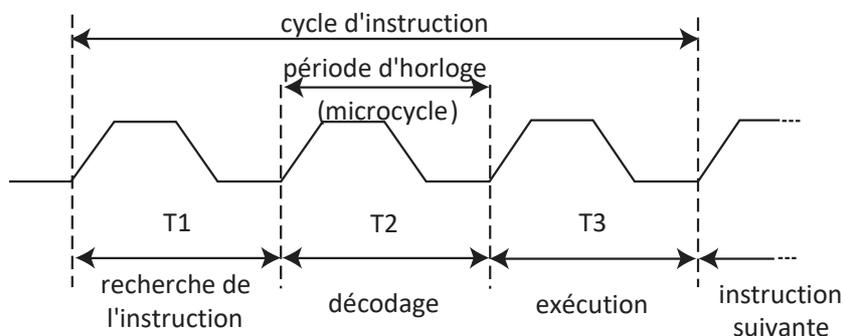
Les indicateurs d'état sont utilisés par les instructions de **saut conditionnels** : en fonction de l'état d'un (ou plusieurs) flags, le programme se poursuit de manière différente. Toutes ces étapes (lecture de l'instruction, décodage, exécution) sont synchronisées par un séquenceur qui assure le bon déroulement des opérations :



2.5 Fonctionnement d'un microprocesseur

Pour exécuter le programme contenu dans la mémoire centrale, le séquenceur du microprocesseur exécute lui-même un programme appelé **micro-code**, contenu dans une mémoire morte à l'intérieur du microprocesseur.

Le séquenceur est dirigé par une horloge qui délivre un signal de fréquence donnée permettant d'enchaîner les différentes étapes de l'exécution d'une instruction :

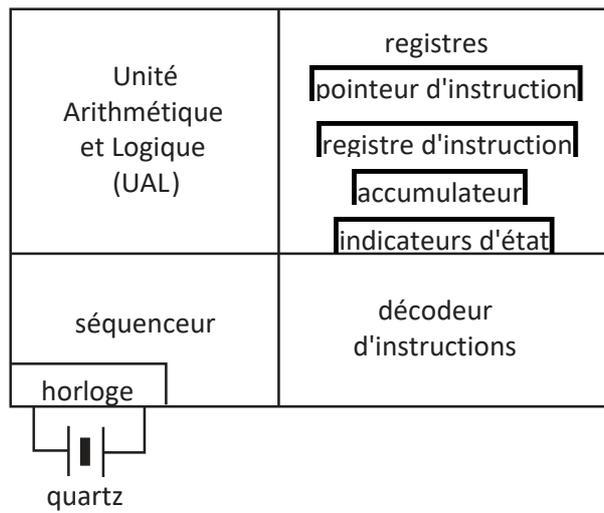


Chaque instruction est caractérisée par le nombre de périodes d'horloge (ou microcycles) qu'elle utilise (donnée fournie par le fabricant du microprocesseur).

Exemple : horloge à 5 MHz, période $T = 1/f = 0,2 \mu s$. Si l'instruction s'exécute en 3 microcycles, la durée d'exécution de l'instruction est : $3 \times 0,2 = 0,6 \mu s$.

L'horloge est constituée par un oscillateur à quartz dont les circuits peuvent être internes ou externes au microprocesseur.

Structure complète d'un microprocesseur simple : pour fonctionner, un microprocesseur nécessite donc au minimum les éléments suivants :



Les mémoires

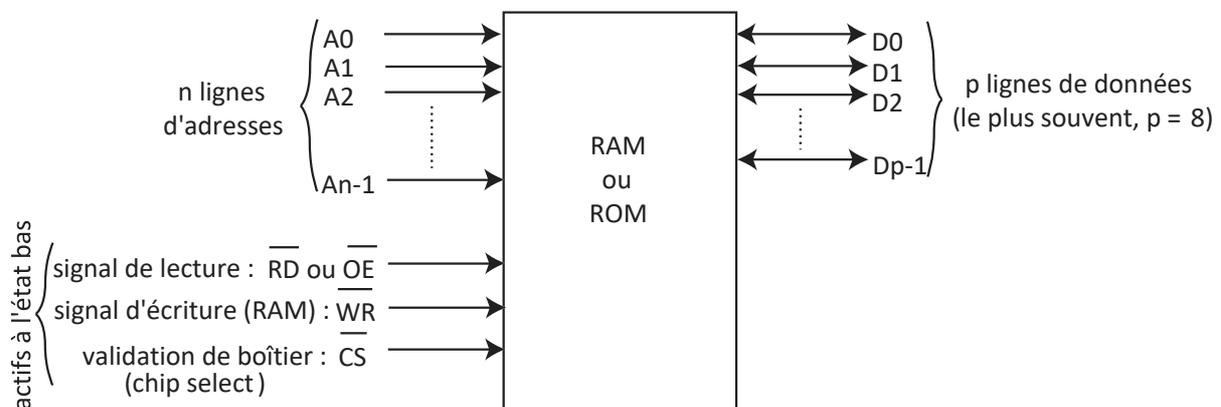
1 Mémoires ROM et RAM

On distingue deux types de mémoires :

- les **mémoires vives** (RAM : Random Access Memory) ou mémoires volatiles. Elles perdent leur contenu en cas de coupure d'alimentation. Elles sont utilisées pour stocker temporairement des données et des programmes. Elles peuvent être lues et écrites par le microprocesseur;
- les **mémoires mortes** (ROM : Read Only Memory) ou mémoires non volatiles. Elles conservent leur contenu en cas de coupure d'alimentation. Elles ne peuvent être que lues par le microprocesseur (pas de possibilité d'écriture). On les utilise pour stocker des données et des programmes de manière définitive.

Les mémoires sont caractérisées par leur **capacité** : nombre total de cases mémoire contenues dans un même boîtier.

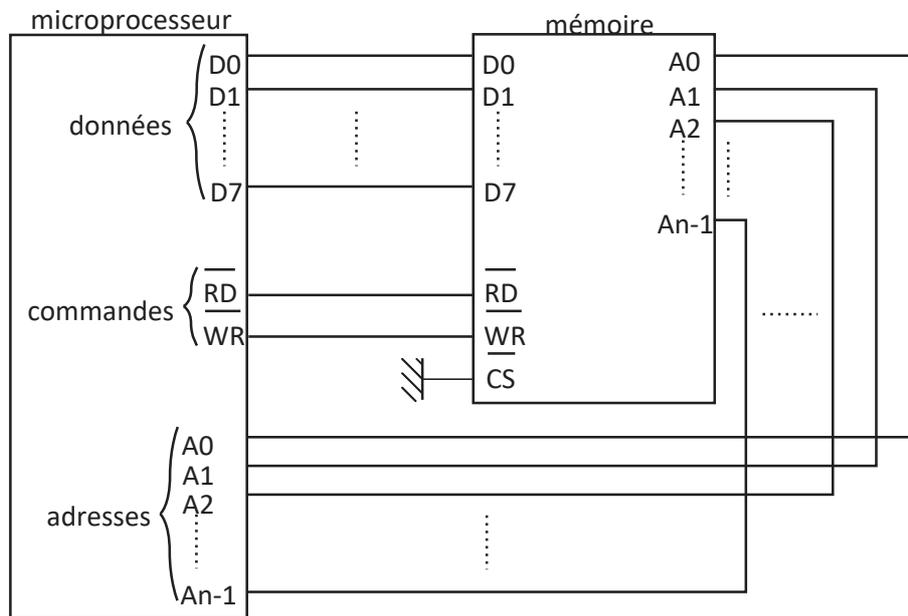
2 Schéma fonctionnel d'une mémoire



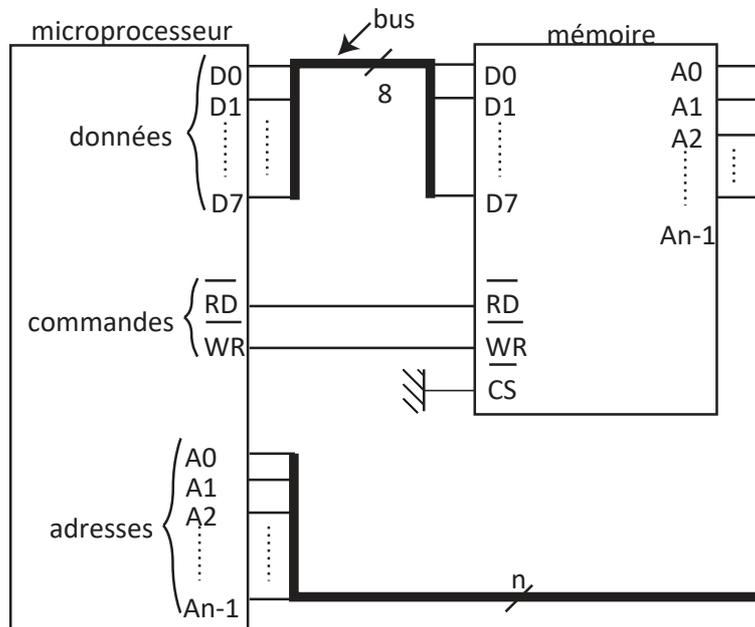
Le nombre de lignes d'adresses dépend de la capacité de la mémoire : n lignes d'adresses permettent d'adresser 2^n cases mémoire : 8 bits d'adresses permettent d'adresser 256 octets, 16 bits d'adresses permettent d'adresser 65536 octets (= 64 Ko), ...

Exemple : mémoire RAM 6264, capacité = 8K×8 bits : 13 broches d'adresses A0 à A12, $2^{13} = 8192 = 8$ Ko.

3 Interfaçage microprocesseur/mémoire



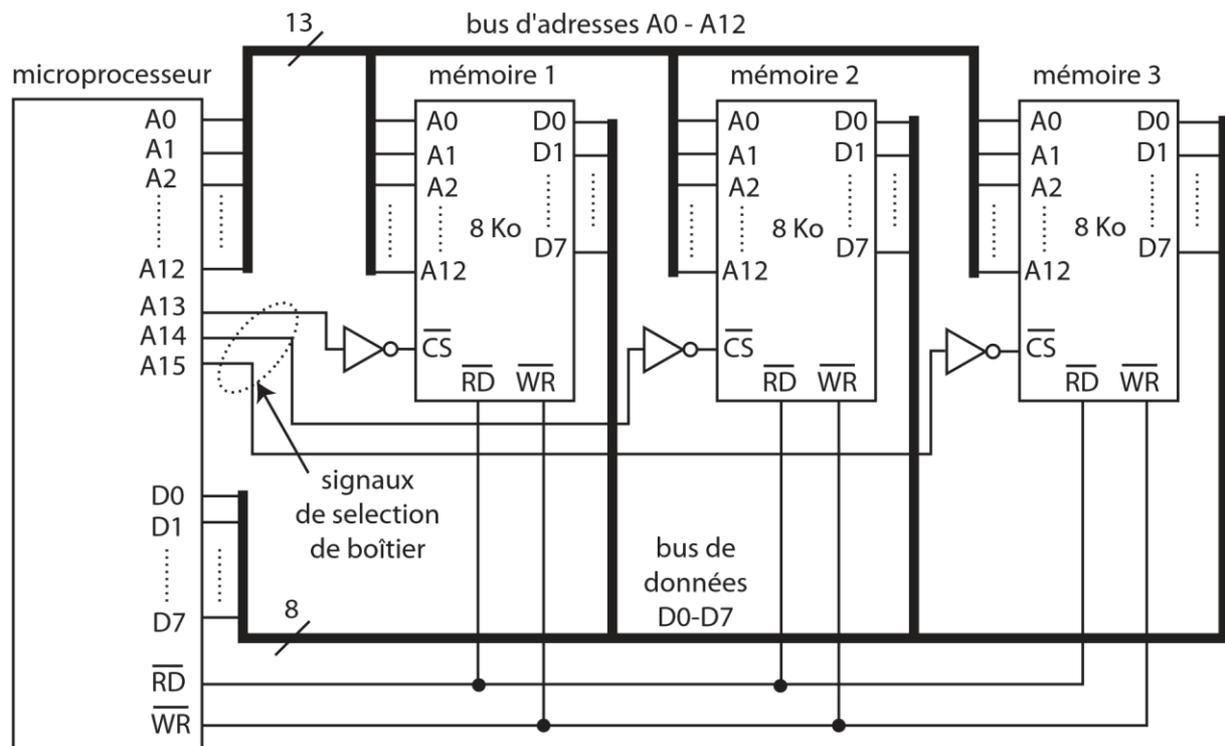
Représentation condensée (plus pratique) :



4. Connexion de plusieurs boîtiers mémoires sur le bus d'un microprocesseur

Les boîtiers m' mémoire possèdent une broche notée CS : Chip Select. Lorsque cette broche est active (état bas), le circuit peut être lu ou écrit. Lorsqu'elle est inactive (état haut), le circuit est exclu du service. Ses broches de données D0 à D7 passent à l'état de haute impédance : tout se passe comme si la mémoire était déconnectée du bus de données du microprocesseur, d'où la possibilité de connecter plusieurs boîtiers mémoire sur un même bus : un seul signal CS doit être actif à un instant donné pour éviter les conflits entre les différents boîtiers.

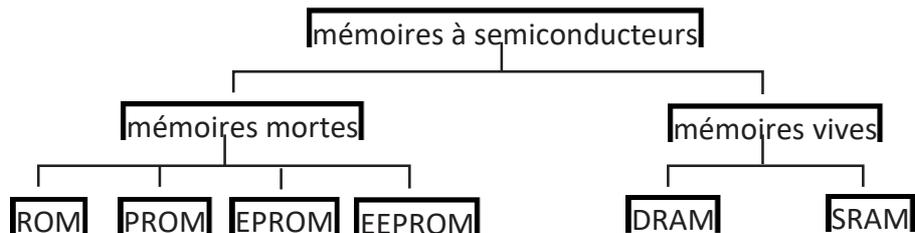
Exemple : connexion de trois boîtiers m' mémoire d'une capacité de 8 Ko chacun (13 lignes d'adresses) sur un bus d'adresse de 16 bits :



mémoire totale
adressable par le
microprocesseur

5. Classification des mémoires

Jusqu'à la fin des années 1970, on utilisait des mémoires à tores magnétiques, lentes et de faibles capacités. Actuellement, on n'utilise plus que des mémoires à semi-conducteurs.



Mémoires mortes :

- **ROM** : Read Only Memory. Mémoire à lecture seule, sans écriture. Son contenu est programmé une fois pour toutes par le constructeur. Avantage : faible coût. Inconvénient : nécessite une production en très grande quantité.
- **PROM** : Programmable Read Only Memory. ROM programmable une seule fois par l'utilisateur (ROM OTP : One Time Programming) en faisant sauter des fusibles. Nécessite un programmeur spécialisé : application d'une tension de programmation (21 ou 25 V) pendant 20 ms.
- **EPROM** : Erasable PROM, appelée aussi UVPROM. ROM programmable électriquement avec un programmeur et effaçable par exposition à un rayonnement ultraviolet pendant 30 minutes. Famille 27nnn, exemple : 2764 (8 Ko), 27256 (32 Ko). Avantage : reprogrammable par l'utilisateur.
- **EEPROM** : Electrically Erasable PROM. ROM programmable et effaçable électriquement. Lecture à vitesse normale (≤ 100 ns). Écriture (= effacement) très lente (≈ 10 ms). Application : les EEPROM contiennent des données qui peuvent être modifiées de temps en temps, exemple : paramètres de configuration des ordinateurs. Avantage : programmation sans extraction de la carte et sans programmeur. Inconvénient : coût élevé.

Mémoires vives :

- **SRAM** : Static Random Access Memory. Mémoire statique à accès aléatoire, à base de bascules à semi-conducteurs à deux états (bascules RS). Famille 62nnn, exemple : 62128 (16 Ko). Avantage : très rapide, simple d'utilisation. Inconvénient : compliqué à réaliser.

- **DRAM** : Dynamic RAM. Basée sur la charge de condensateurs : condensateur chargé = 1, condensateur déchargé = 0. Avantage : intégration élevée, faible coût. Inconvénient : nécessite un rafraîchissement périodique à cause du courant de fuite des condensateurs. Application : réalisation de la mémoire vive des ordinateurs (barrettes mémoire SIMM : Single In-line Memory module).