

Informatique *Architecture des Ordinateurs*

1. Historique et évolution des ordinateurs

Historique et évolution des ordinateurs

□ Qu'est ce que l'informatique ?

- L'informatique, contraction d'**information** et **automatique**, est la science du traitement de l'information.
- La science **de traitement de l'information** est apparue au milieu du 20ème siècle, elle a connu une évolution extrêmement rapide. A sa motivation initiale qui était de faciliter et d'accélérer le calcul, ce sont ajouté de nombreuses fonctionnalités:
 - ✓ comme l'automatisation;
 - ✓ le contrôle et la commande de processus,
 - ✓ la communication
 - ✓ le partage de l'information.
- Une information est tout ensemble de données:
 - ✓ Textes,
 - ✓ nombres,
 - ✓ sons, images, etc.,
 - ✓ les instructions composant un programme.

Historique et évolution des ordinateurs

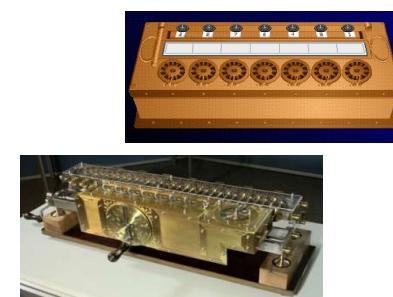
- L'homme a toujours cherché à s'aider pour le calcul
- Les premières machines à calculer



- 1623 : machine à calculer de SCHICKARD
- 1642 : machine à calculer de Pascal : addition et soustraction
- 1673 : Leibnitz : multiplication et division

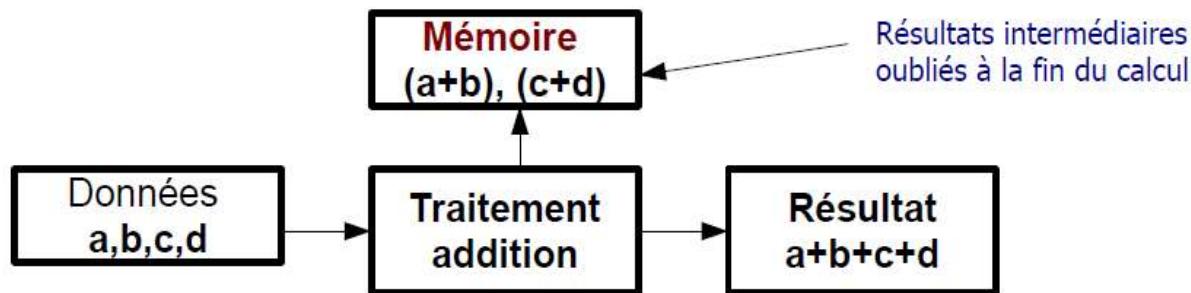


Problème: pas de résultats (traitements) intermédiaires



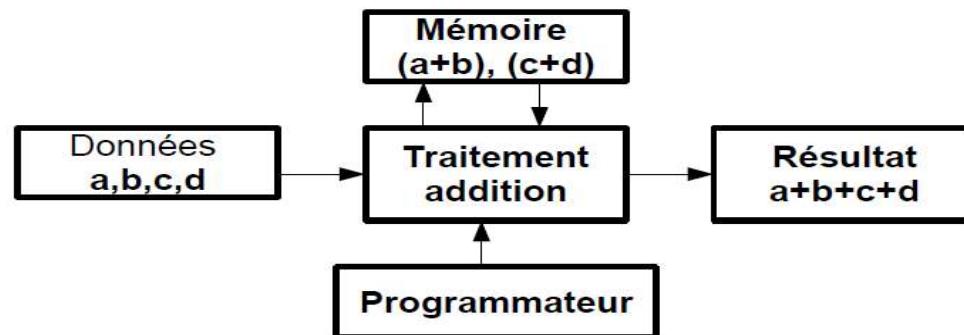
Historique et évolution des ordinateurs

- Machine à calculer dotée de mémoire



Problème: un seul traitement à la fois

- Machine à calculer dotée de mémoire et de programmateur



Programme : suite d'actions élémentaires consécutives

Programmateur : orchestre la suite des opérations

Problème : Le programme ne tient pas compte des résultats intermédiaires (ex : si a est vrai alors faire b)

Chap1: Historique et évolution des ordinateurs

• 1945-1958

- Ordinateurs dédiés, exemplaires uniques
- Machines volumineuses et peu fiables
- Technologie à lampes, relais, résistances
- 10^4 éléments logiques - programmation par cartes perforées

• 1958-1964

- Usage général, machines fiables
- Technologie à transistors
- 10^5 éléments logiques
- Apparition des langages de programmation évoluées (COBOL, FORTRAN, LISP)

Chap1: Historique et évolution des ordinateurs

• 1964-1971

- Technologie des circuits intégrés (S/MSI : Small/Medium Scale Integration)
- 106 éléments logiques
- Avènement du système d'exploitation complexe, des mini-ordinateurs
- Premier microprocesseur 4004 de INTEL : toutes les composants de la CPU sont réunies sur une même puce
- Passage du méga informatique aux mini-informatique

• 1972-1977

- Usage général, machines fiables
- Technologie LSI (L'intégration à très grande échelle)
- 107 éléments logiques
- Avènement de réseaux de machines
- Traitement distribué / reparti

Chap1: Historique et évolution des ordinateurs

- Après : Le but originel de cette cinquième génération était les machines langages dédiées à l'Intelligence Artificielle (IA)
- Technologies VL/WSI (Very Large/ Wafer)
- 108 éléments logiques (le PII contient 7,5 millions de transistors, mémoire non comprise)
- Systèmes distribués interactifs
- Multimédia, traitement de données non numériques (textes, images, paroles)
- Parallélisme massif

Évolution technologique

1945-1958 : 1^{ière} génération

Logiciel

- Exploitation séquentielle
 - 1^{ière} étape: Chargement
 - 2^{ième} étape: Exécution
 - 3^{ième} étape: Traitement

Langage

- Machine
- Assembleur

Évolution technologique

1958-1964: 2^{ième} génération

Logiciel

- Control de processus (microprogrammation)
- Mécanisme d'interruption: La simultanéité des traitements

Langage

- Simultanéité E/S calcul
- Traitement par lot
- Fonctions mathématiques
- Avantages:

Gagne du temps + Espace



Compréhension des programmes

Évolution technologique

1964-1971: 3^{ième} génération

Logiciel

- Temps partagé, multiprogrammation, partitionnement de mémoire, télétraitements, connexion de plusieurs sites entre eux, connexion directement sur la machine
- Gestion de la mémoire
- Chargement Continu

1972-1985: 4^{ième} génération

Logiciel

- Machine Virtuelle
- Temps partagé
- Traitement distribué / reparti (BD)
- Dialogue homme/Machine (clavier)
- Réseaux(local et à distance)

Résumé 1

génération	date approximative	technologie	vitesse (opérations/s)
1	1946-1957	tube à vide	40 000
2	1958-1964	transistor	200 000
3	1965-1971	SSI/MSI	1 000 000
4	1972-1977	LSI	10 000 000
5	1978-	VLSI	100 000 000

SSI – Small Scale Integration

MSI – Medium Scale Integration

LSI – Large Scale integration

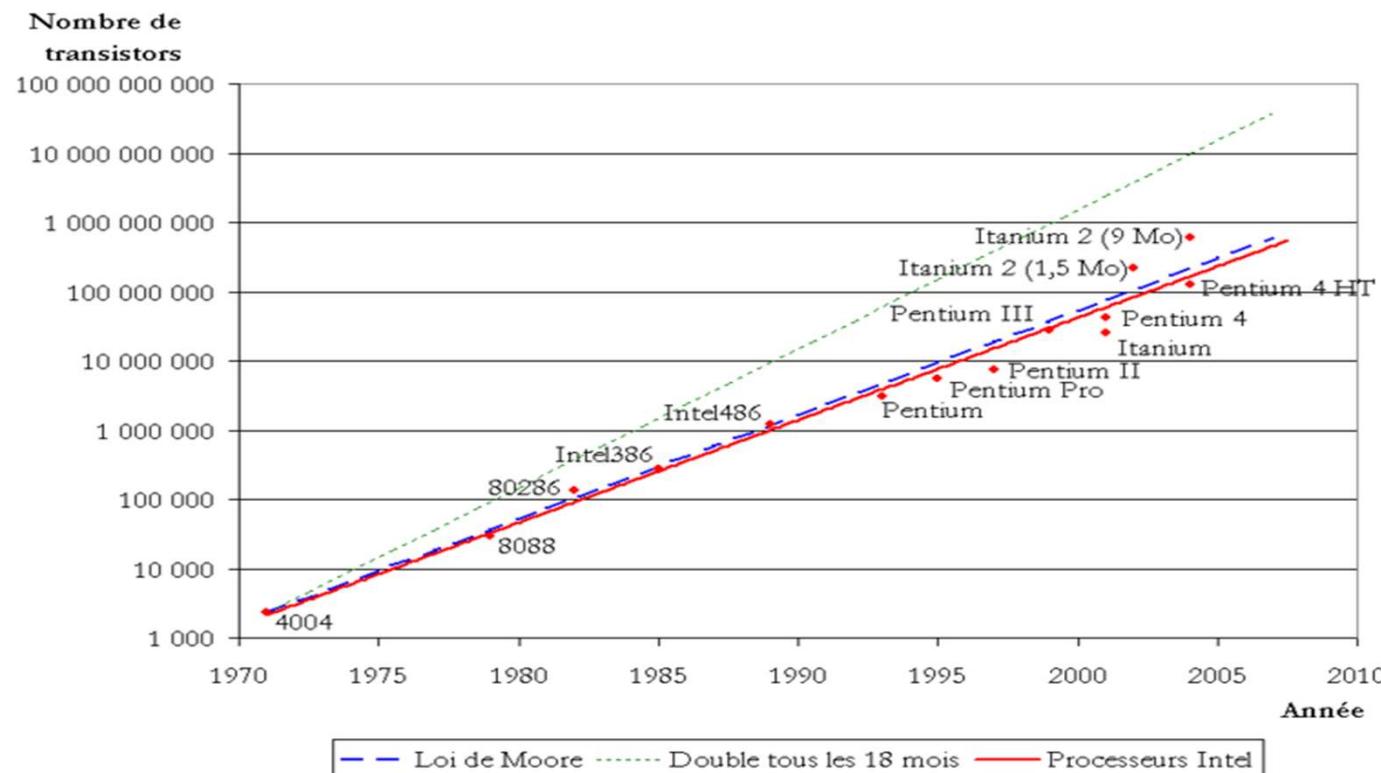
VLSI – Very Large Scale Integration

Résumé 2



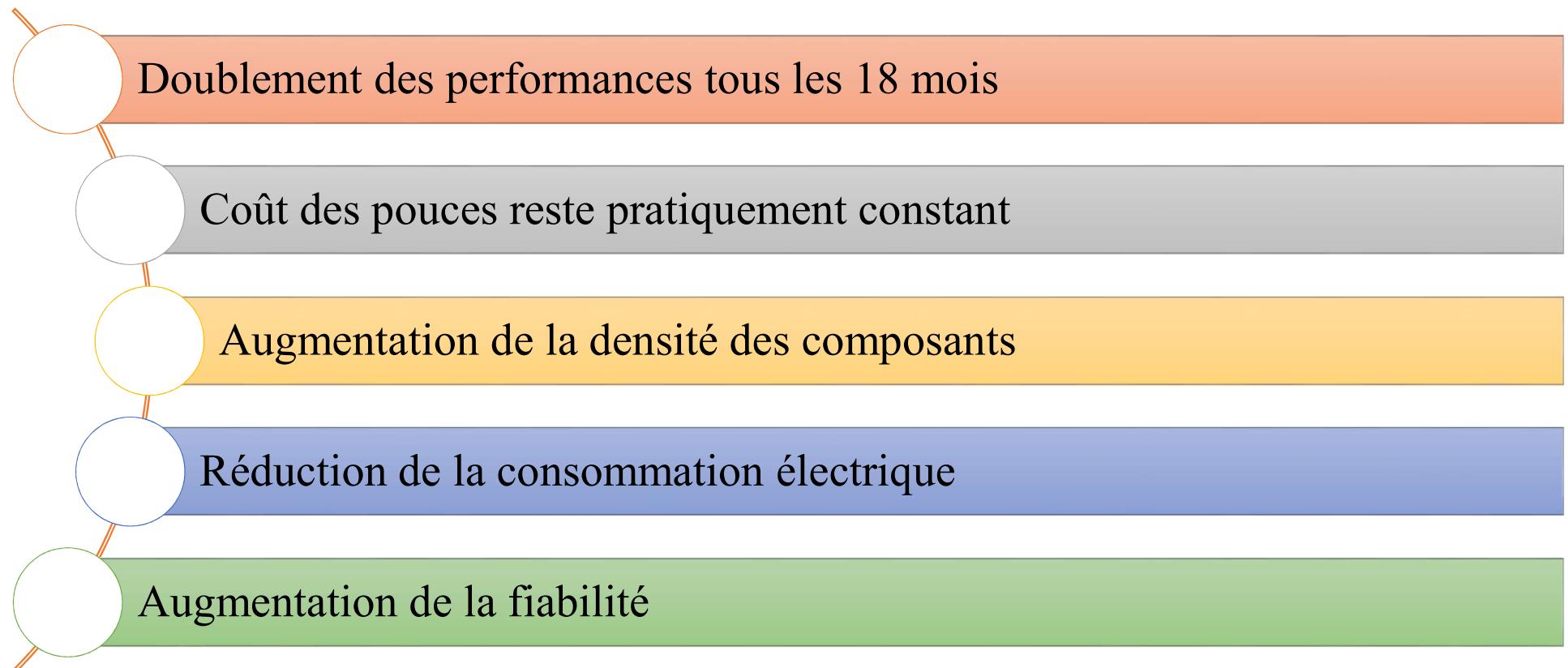
Loi de Moore (1975)

Développement de la puissance de calcul des ordinateurs

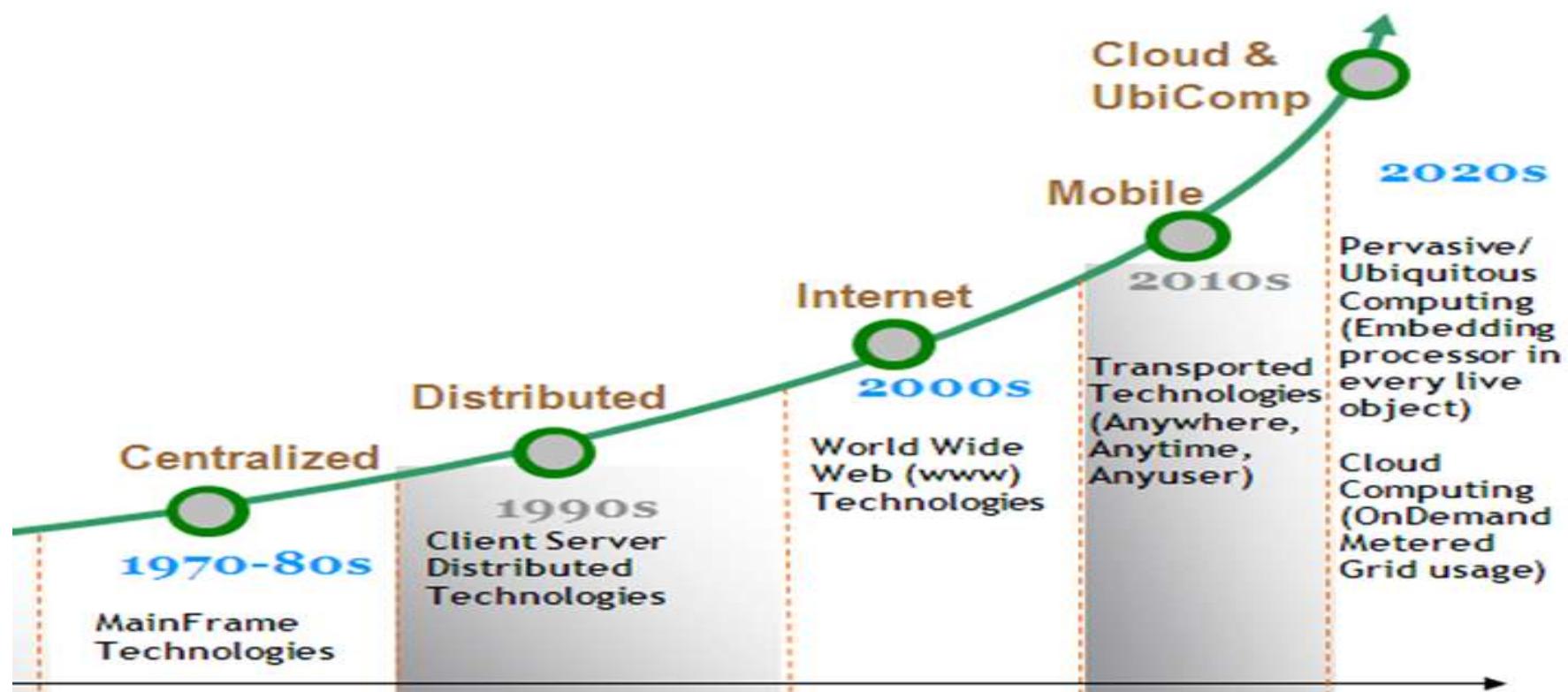


Réf: https://fr.wikipedia.org/wiki/Loi_de_Moore

Loi de Moore



Evolution of Computing



ENIAC

- 1945 l'ENIAC (Electronic Numerical Integrator And Calculator)
18000 Tubes – 30 tonnes
Multiplie 2 nombres de 10 chiffres en 3 millisecondes
- L' ENIAC est le premier ordinateur 100% électronique

Depuis l 'ENIAC Tous les **ordinateurs** sont fondés sur les mêmes principes de base :

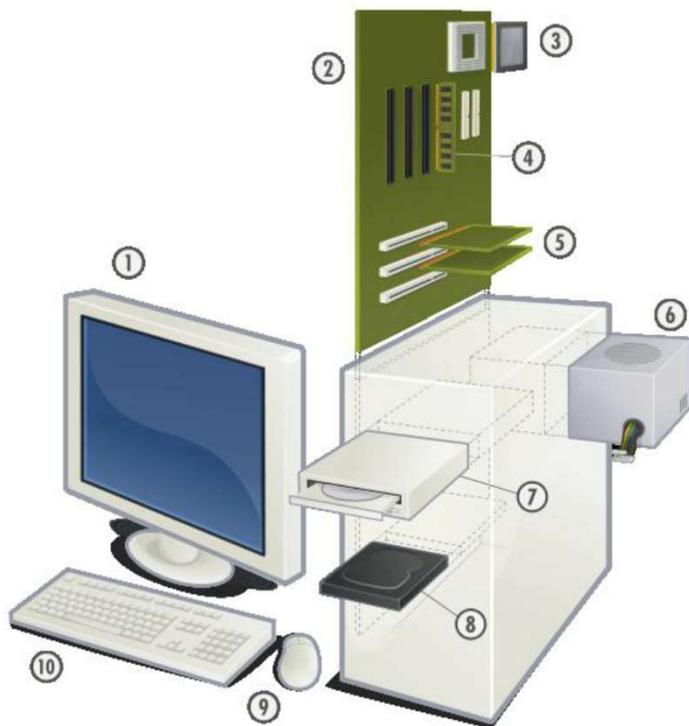
- programmes / données
- processeur / mémoire / périphériques
- informations codées en binaire
- traitements effectués en binaire

Seule la technologie a changée :

lampe à vide -> transistor -> circuit intégré -> microprocesseur

d 'où : plus rapide, plus petit, moins gourmand, plus fiable, moins coûteux

Ordinateur : Point de vue externe



1. Ecran
2. Carte mère
3. CPU (Microprocesseur)
4. Mémoire vive (RAM)
5. Cartes de périphériques
6. Alimentation
7. Lecteur de disques (ex.DVD)
8. Disque dur
9. Souris
10. Clavier

Ordinateur : Point de vue interne

L'unité centrale contient 3 unités fonctionnelles :

- L'automate : donne les ordres
- La partie calcul : exécute les ordres
- La mémoire : stocke les ordres et les données

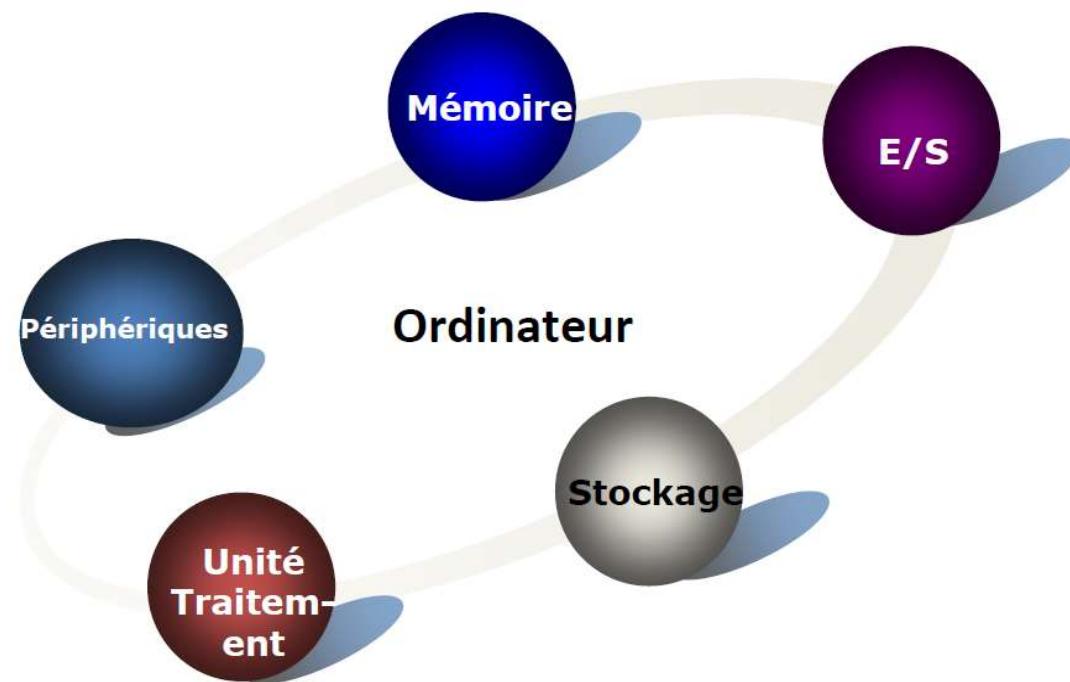
Une interaction se traduit par l'exécution d'une séquence d'opérations par ces unités

Ordinateur : Point de vue interne

- Les composantes d'un ordinateur se divisent en deux sections principales :
 - ❶ Equipment solide (Hardware) : exécute certaines opérations comme l'addition, la soustraction, le transfert des données et de simples tests
 - Périphériques d'entrée (clavier, souris...)
 - Périphériques de traitement (UAL)
 - Support de stockage (mémoire)
 - Dispositifs de sortie (écran, son ...)
 - Dispositifs de communication (Bus)
 - ❷ Logiciels (Software) : ensemble des instructions appelées programmes. Ces programmes sont indispensables pour faire marcher l'ordinateur.
 - Système d'exploitation (Windows , linux)
 - Application (Microsoft Office, Skype)

II. Architecture et fonctionnement d'un microprocesseur

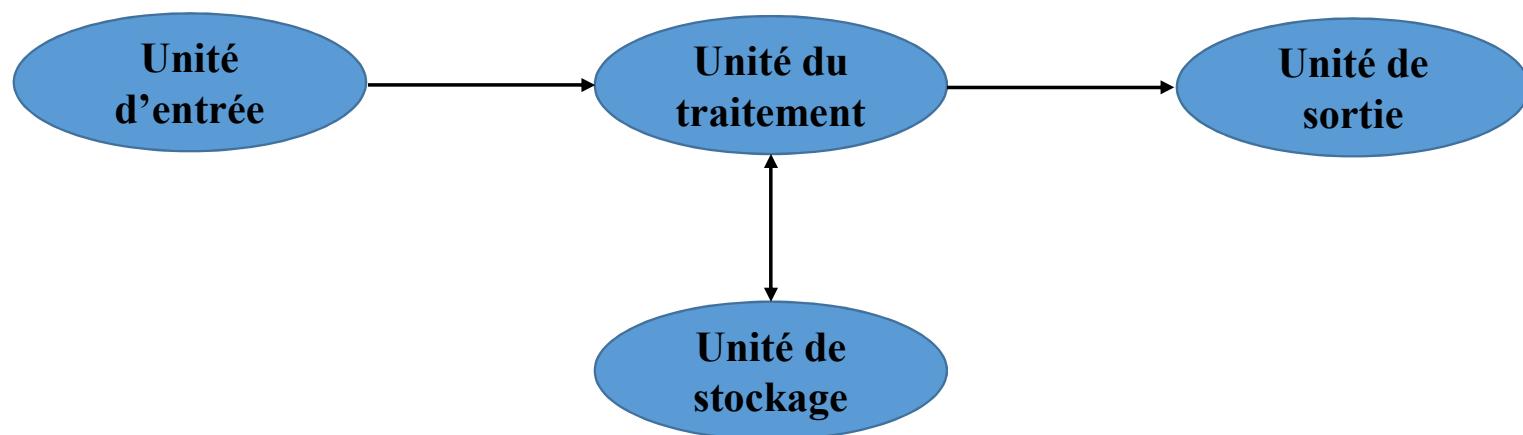
Principaux éléments de l'ordinateur



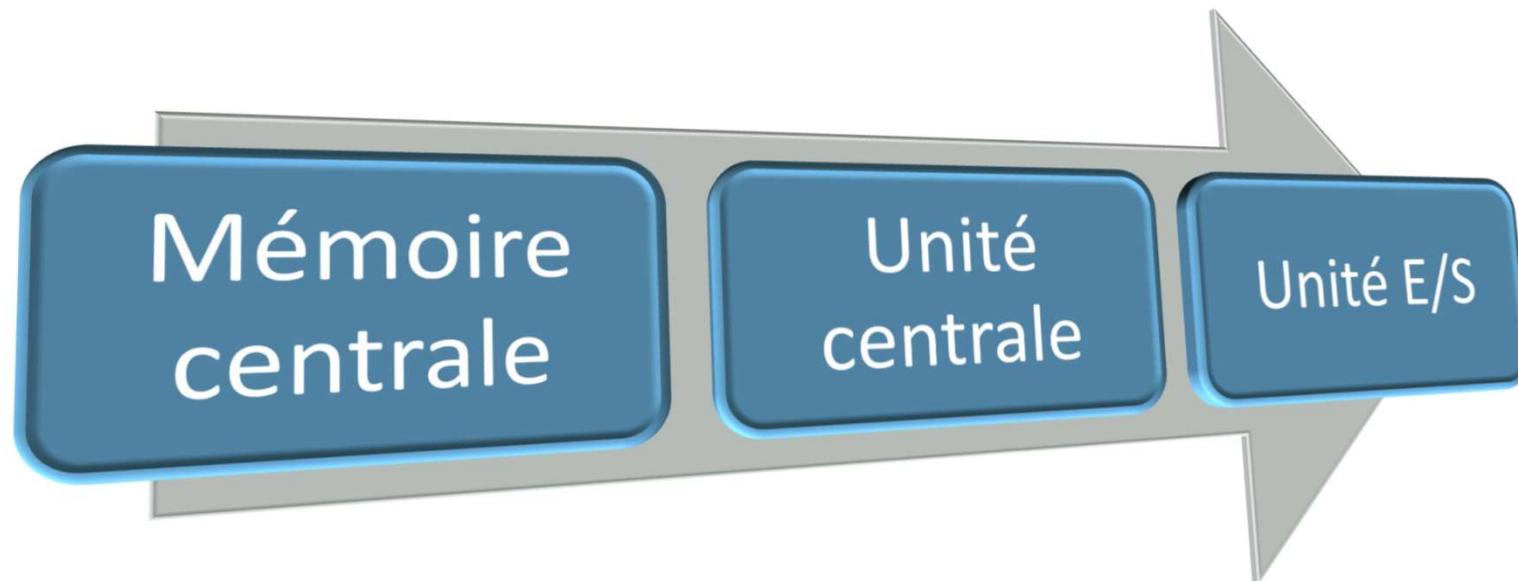
Définition de l'ordinateur:

- Un ordinateur est une machine de traitement automatisé de l'information.
- **Fonctionnalités:**
 - Acquérir l'information, la stocker, la modifier en appliquant des instructions prédéfinies (programmes)
 - Interagir avec l'environnement grâce à des périphériques (écran, clavier....).

Schéma fonctionnel



Unités fonctionnelles d'un ordinateur



L'unité centrale(Boîtier)

Toutes les unités sont connectées au boîtier de l'ordinateur.

Le boîtier contient plusieurs éléments comme :

- **La carte mère, Le processeur, Les mémoires, Les cartes d'extensions, Le disque dur, ...**



La carte mère

La carte mère est un **composant principal**, où sont **connectées tous les éléments** de l'ordinateur (processeur, RAM, disque dur, cartes d'extension, périphériques,...).



Le processeur(CPU)

Le processeur ou L'unité centrale de traitement est le cerveau de l'ordinateur. C'est un composant électronique (puce) qui **traite les informations** :

- Effectué des calculs ;
- Exécuté les instructions ;
- Géré les flux d'informations circulant dans l'ordinateur ;
- Organisé les échanges de données entre les différents composants (disque dur, RAM, carte wifi, périphérique, ...).



Le processeur(CPU)

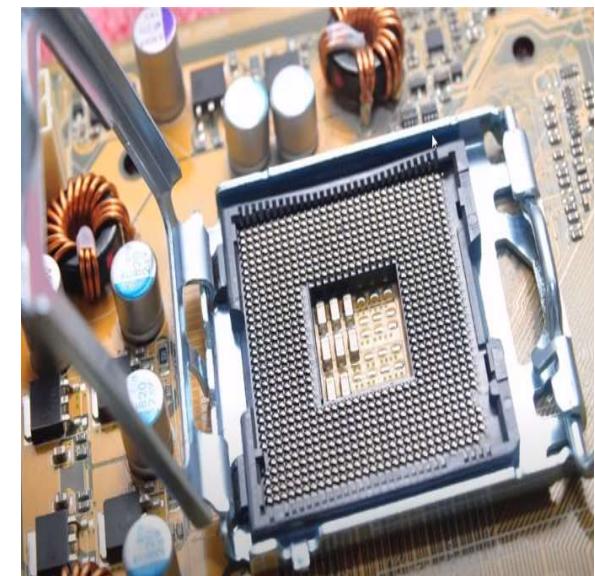
Remarque

Les deux principaux fabricants de microprocesseurs sont **AMD** et **INTEL**. Chacun propose deux gammes de CPU bien distinctes, bas de gamme et haut de gamme. Chaque constructeur a opté pour des technologies différentes.

Exemples

Intel : Core i9, Core i7, Core i5, Core i3, Core de duo, Pentium M, ...

AMD : Ryzen 7, Ryzen 5, Ryzen 3, K10, K9, K5, ...

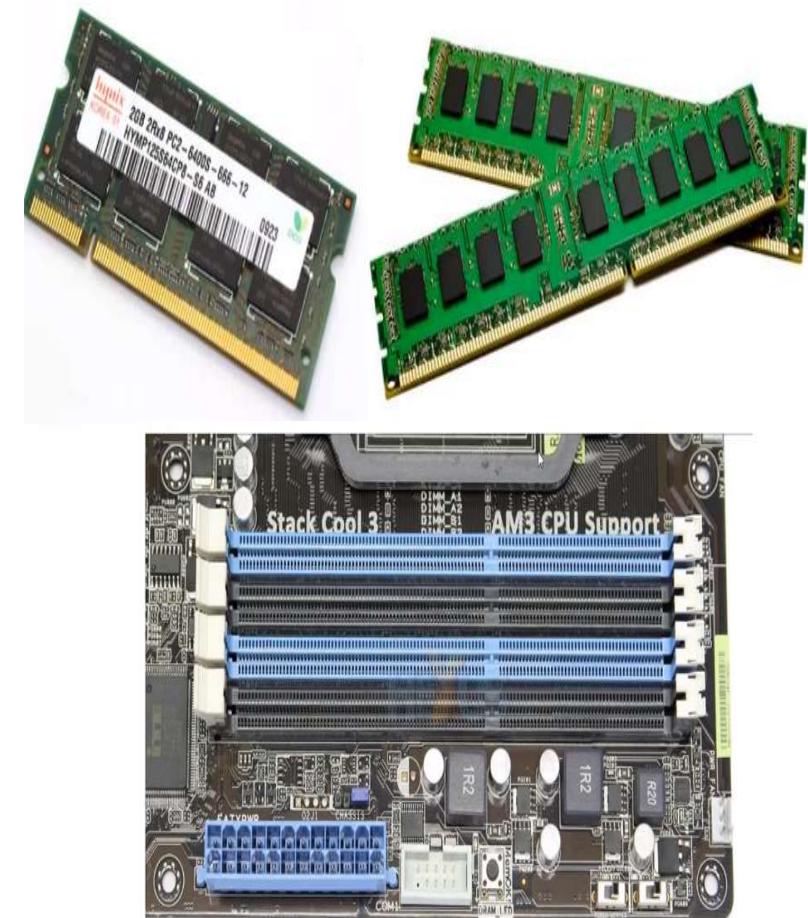


Les mémoires

a. La mémoire vive (RAM)

La mémoire vive ou la RAM est constituée d'un ensemble de cases mémoires, elle sert à mémoriser temporairement des données pour qu'elles soient chargées et traitées rapidement par le processeur.

- Cette mémoire est dite temporaire.
- Mémoire dans lequel on peut lire et écrire.



Les mémoires

a. La mémoire vive (RAM)

Remarque

Lorsque l'ordinateur est éteint, cette mémoire est **effacée**.

C'est pour cela qu'il est important de sauvegarder vos données sur le **disque dur**.

Les mémoires

b. La mémoire morte (ROM)

La mémoire morte ou la **ROM** contient les **programmes de constructeur** nécessaires au démarrage de l'ordinateur, il est fixé lors de la construction de l'ordinateur.

- Cette mémoire est dite permanente.
- Cette mémoire est en lecture seule.



Le disque dur (HDD) et le disque électrique(SSD)

Le disque dur ou le disque électrique sont des composants qui permettent un accès aux données à long terme. Ils stockent et accèdent aux fichiers, logiciels et système d'exploitation.



Le disque dur (HDD) et le disque électrique(SSD)

Remarque

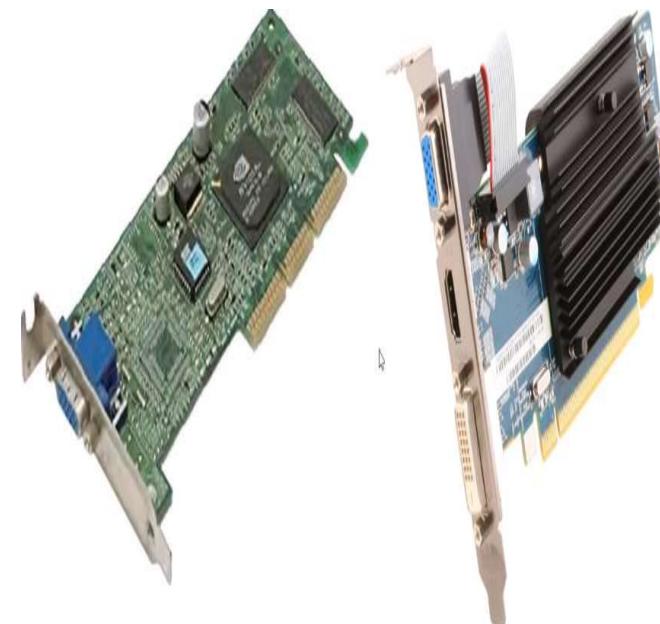
Il existe deux types : **HDD** et **SSD**, le plus important c'est que le **SSD (Solid State Drive)** est le plus rapide, plus chère et plus solide que le **HDD (Hard Disk Drive)**.

Les cartes d'extensions

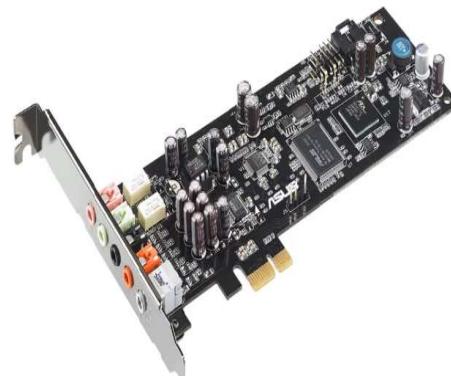
Les cartes d'extensions sont des composants connectés à la carte mère. Elles ajoutent des capacités ou des fonctionnalités nouvelles à un ordinateur.

Exemples :

Carte Wifi, carte son, carte graphique, une carte Tuner TV, ...



Les cartes d'extensions



Les unités(Les périphériques)

Un périphérique est un **matériel extérieur**, relié à l'unité centrale afin **d'envoyer, recevoir ou stocker les informations.**

Les unités(Les périphériques)

1. Les unités d'entrée (périphériques d'entrée)

Les unités d'entrée permettent **d'envoyer** les informations à l'unité centrale.

Exemples : Le clavier, le scanner, Web caméra, la souris, le microphone, lecteur CD/DVD, ...

Les unités(Les périphériques)



Les unités(Les périphériques)

2. Les unités de sortie (périphériques de sortie)

Les unités de sortie permettent de **recevoir** ou **d'afficher** les informations sortant de l'unité centrale.

Exemples : L'écran, L'imprimante, les haut-parleurs, le vidéo projecteur, le casque (audio),

Les unités(Les périphériques)



Les unités(Les périphériques)

3. Les unités d'entrée-sortie (périphériques d'entrée-sortie)

Les unités d'entrée-sortie permettent d'**envoyer** et d'**afficher** ou **recevoir** les informations.

Exemples : L'écran tactile, le casque (avec microphone), imprimante multifonction, ...



Les unités(Les périphériques)

4. Les unités de stockage (périphériques de stockage)

Les unités de stockage permettent de **stocker** et de **conserver** les informations.

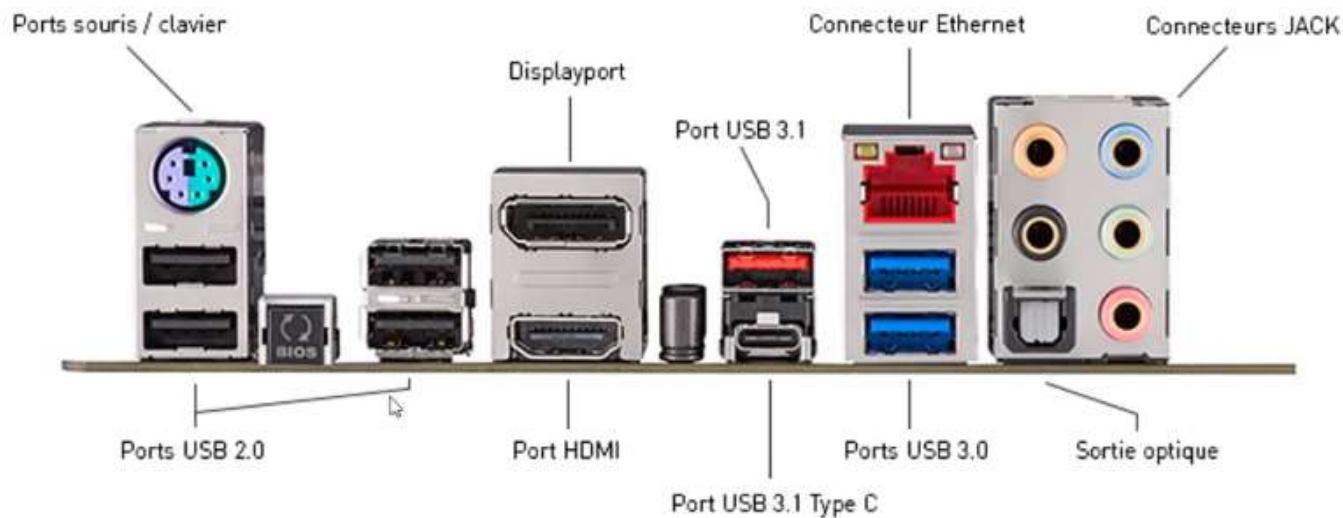
Exemples : Disque dur externe, Clé USB, DVDROM, CD-ROM, carte mémoire, ...

Les connecteurs(ports) des unités

La carte mère est muni d'un ensemble de ports destinés à **connecter (brancher)** les unités.



Les connecteurs(ports) des unités



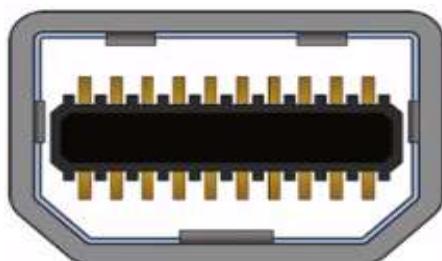
Voici par exemple ce que peut proposer une carte mère comme connexion

Les connecteurs(ports) des unités

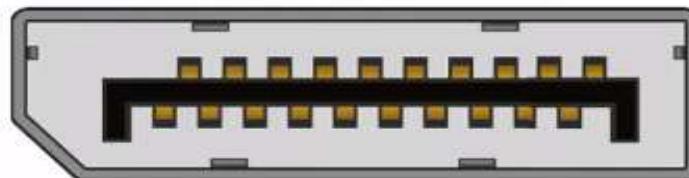
Le tableau ci-dessous présente des exemples des unités qui se connectent avec un type de ports

Port	USB (2.0, 3.0, 3.1 et type C)	HDMI, Dis- playport, VGA	Jack (Audio)	Ethernet (réseau)	PS/2
Unité (s)	La plupart des unités	Ecran, vidéo projecteur, ...	Microphone, haut-parleur, ...	Câble réseau	Souris et clavier

Les connecteurs(ports) des unités



Mini DisplayPort



DisplayPort



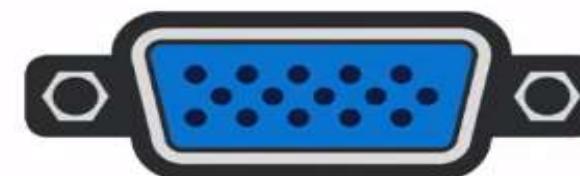
HDMI



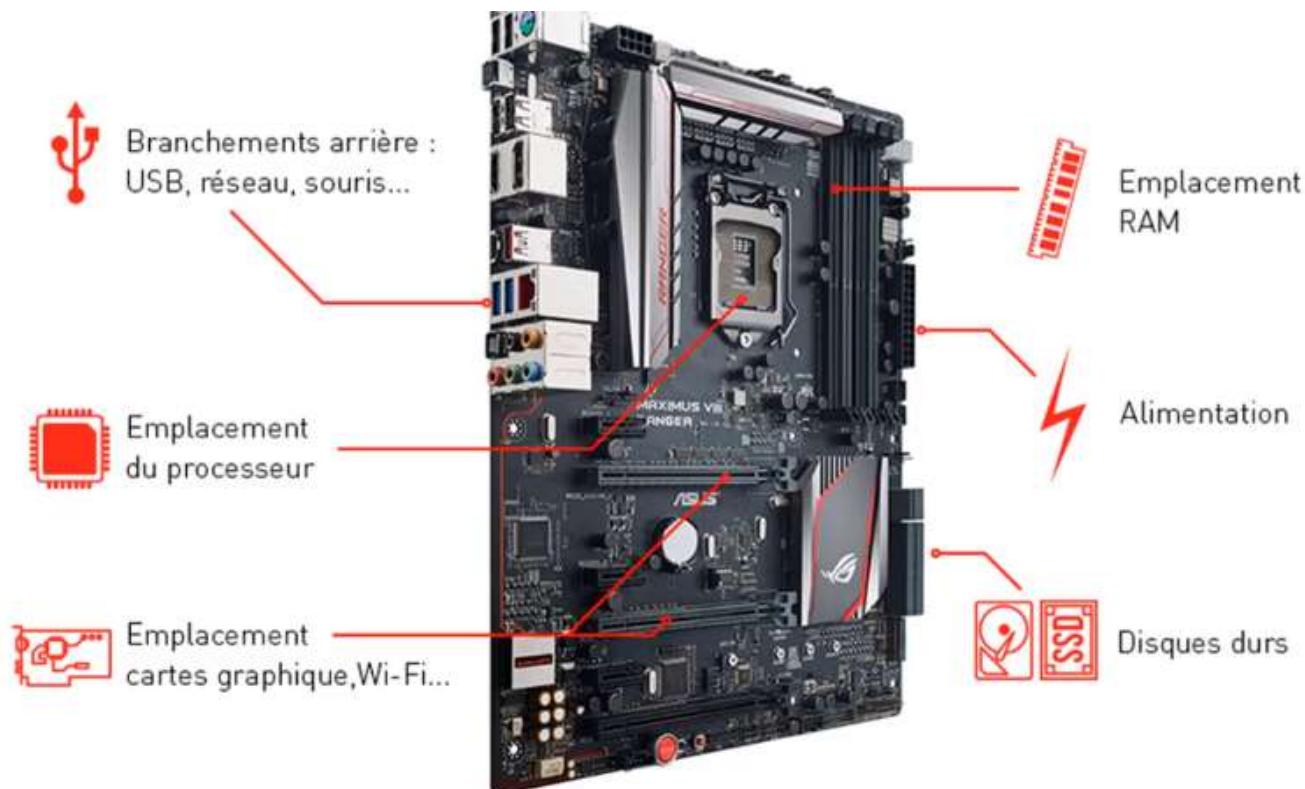
Dual-link DVI

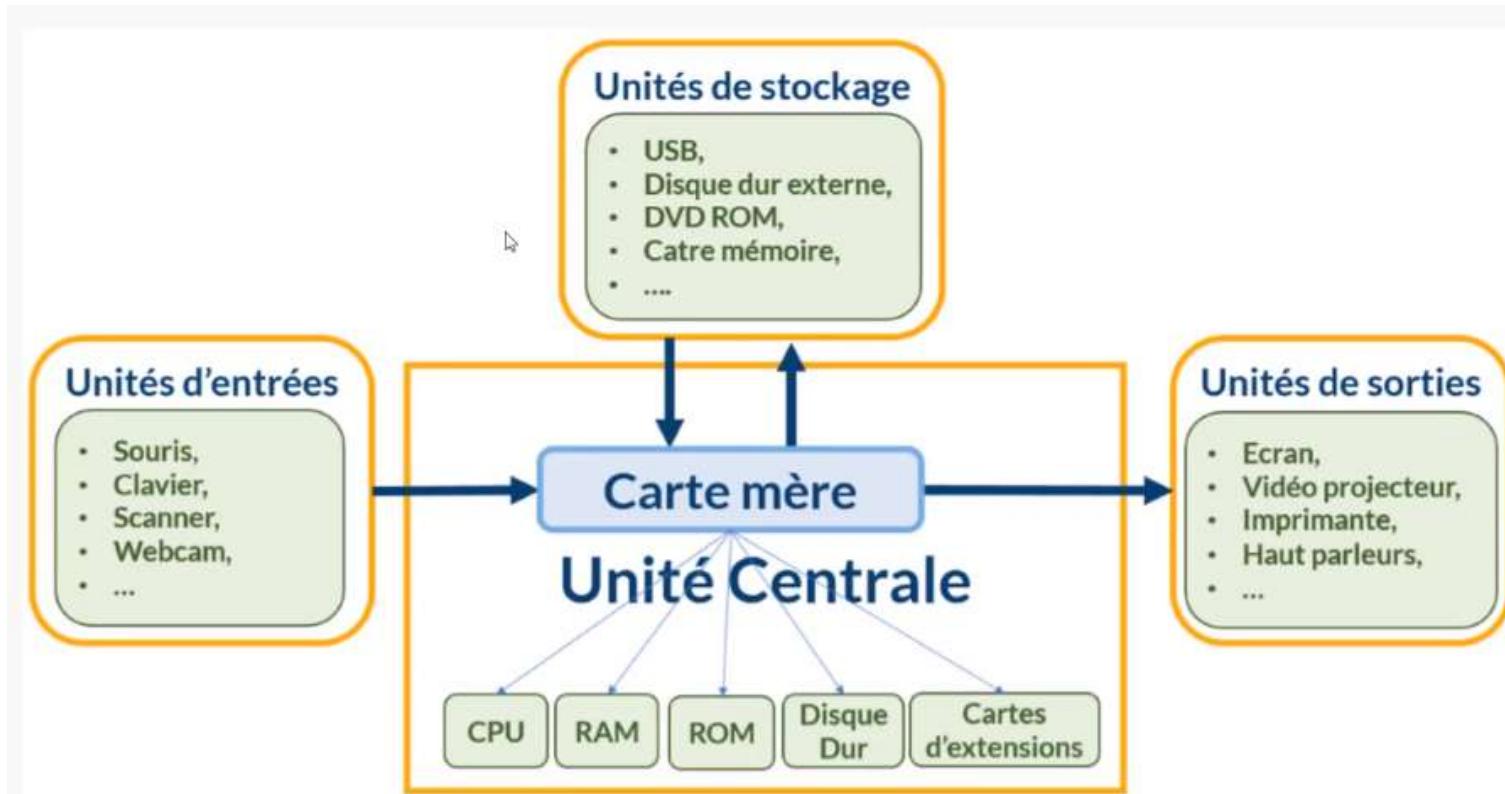


DVI-I

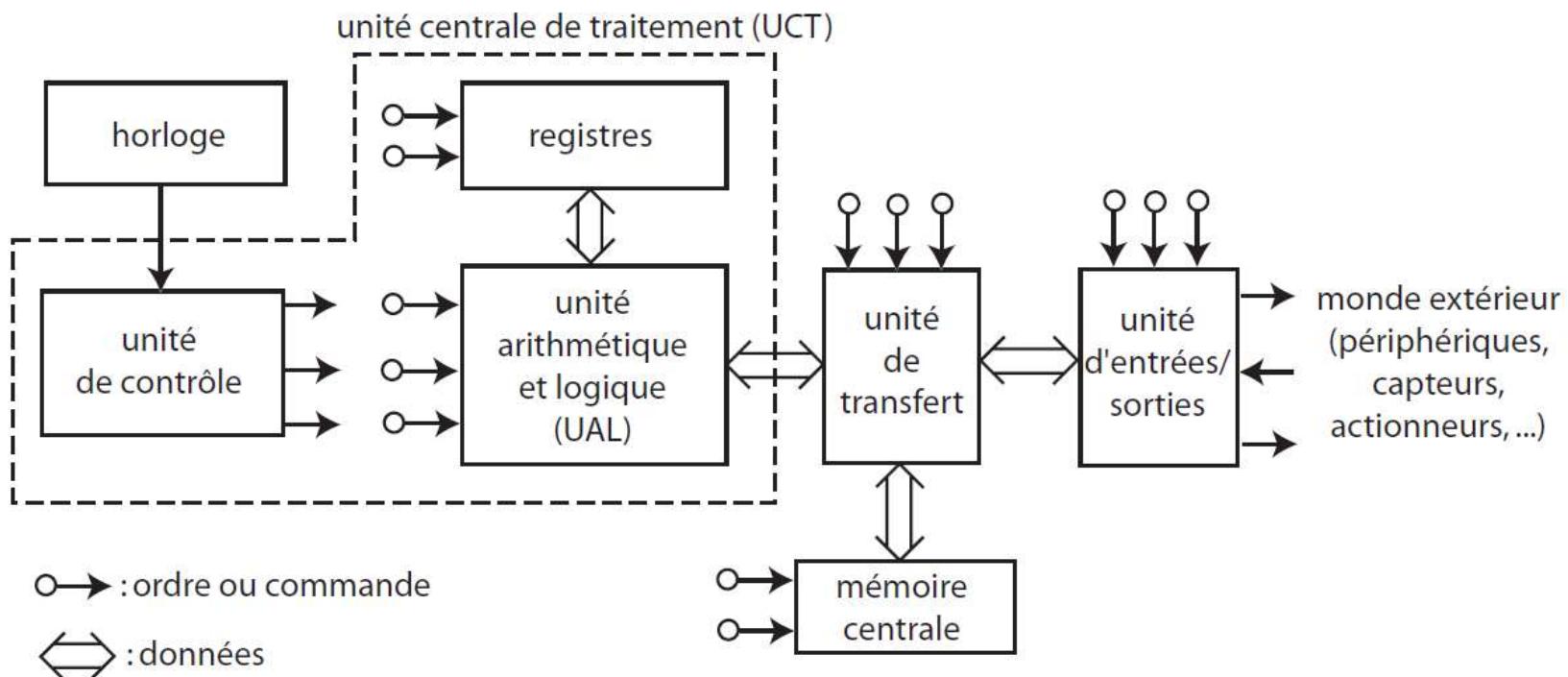


VGA





Structure d'un calculateur



Structure d'un calculateur

L'élément de base d'un calculateur 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 calculateur ;
- 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 calculateur afin d'exécuter un programme.

La **mémoire centrale** contient :

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

Structure d'un calculateur

- L'**unité d'entrées/sorties (E/S)** est un intermédiaire entre le calculateur 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 calculateur 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**.

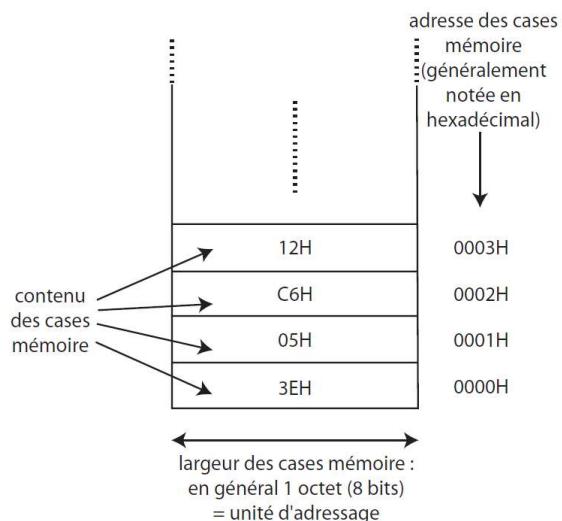
Un calculateur construit autour d'un microprocesseur est un **microcalculateur** ou un **microordinateur**.

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

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 :



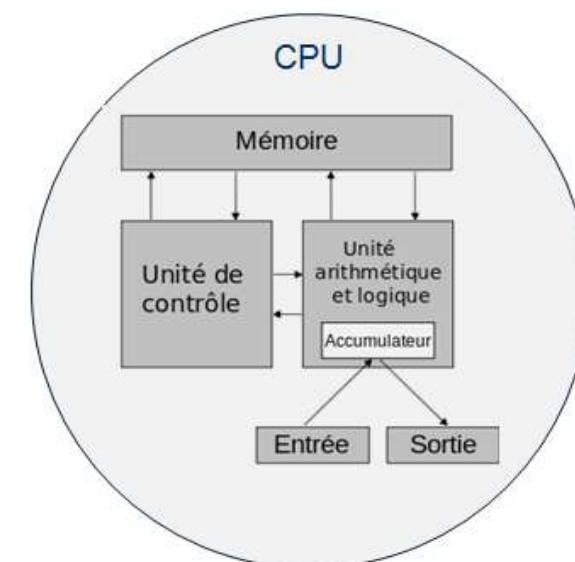
Une case mémoire peut être lue ou écrit par le microprocesseur (**cas des mémoires vives**) ou bien seulement lue (**cas des mémoires mortes**).

Circulation de l'information dans un calculateur

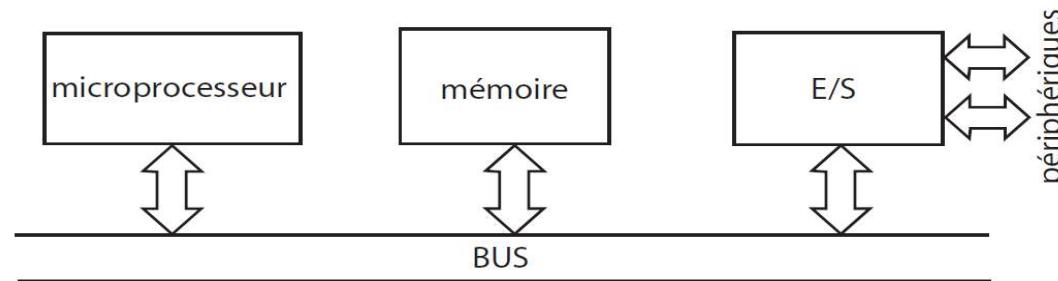
- La réalisation matérielle des ordinateurs est généralement basée sur l'architecture de **Von Neumann** :

En 1951 Von Neumann a défini ce que devrait être un ordinateur à programme enregistré:

- Unité arithmétique et logique (UAL) qui effectue les opérations de base,
- Unité de contrôle qui est chargée du séquençage des opérations,
- Mémoire qui contient à la fois les données et le programme,
- Dispositifs d'entrée-sortie qui permettent de communiquer avec le monde extérieur.



L'architecture de Von Neumann :



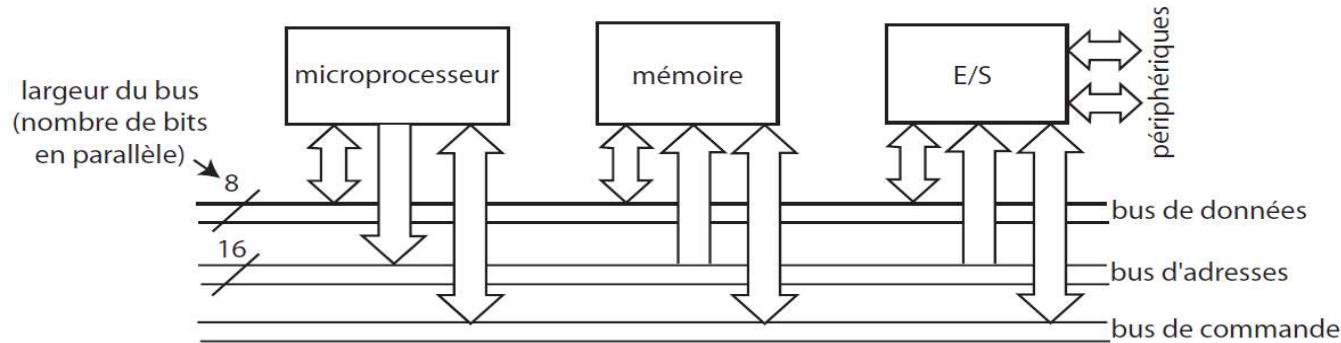
Le microprocesseur échange des informations avec la mémoire et l'unité d'E/S, sous forme de mots binaires, au moyen d'un ensemble de connexions appelé **bus**. Un bus permet de transférer des données sous forme **parallèle**, c'est-à-dire en faisant circuler n bits simultanément.

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, ...

l'architecture de Von Neumann :

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 dérogation pour les circuits d'accès direct à la mémoire, DMA).

Accès direct à la mémoire

- L'accès direct à la mémoire (Direct Memory Access DMA) est une technique matérielle facilitant les opérations d'entrée-sortie, c'est-à-dire les échanges de données entre le microcontrôleur, ou le microprocesseur, et le monde extérieur

Principe du DMA

Des données vont pouvoir être échangées entre la mémoire centrale et les circuits périphériques sans intervention du CPU

- Un sous-système indépendant du CPU va transférer les données du circuit périphérique vers la mémoire (entrée) ou de la mémoire vers le circuit périphérique (sortie)

L'architecture de Von Neumann :

- Bus de contrôle: envoie le signal (Ecriture ou lecture)
- Le processeur place le signal read sur le bus d contrôle
- La MC place la donnée sur le bus de données et le processeur la récupère
- Le processeur place la donnée dans le bus de donnée, l'adresse sur le bus d'adresse et envoie le signal d'écriture sur le bus de contrôle et la donnée est écrite dans la MC.

Unités d'changes

- Les périphériques sont généralement des appareils électromagnétiques
- Nécessité des contrôleurs (conversion des signaux)

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 par :

- **le compteur de programme** : (en anglais Program Counter PC) appelé aussi compteur ordinal (CO). Le CO est constitué par un registre dont le contenu représente l'adresse de la prochaine instruction à exécuter. Il est donc initialisé avec l'adresse de la première instruction du programme. Puis il sera incrémenté automatiquement pour pointer vers la prochaine instruction à exécuter.
- **le registre d'instruction** : Contient l'instruction en cours de traitement.
- **le décodeur d'instruction** :
- **Le 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 micro-microprocesseur.

Unité de commande (UC)

Program Count Register (PC)

- C'est un registre d'adresses.
- Contient l'adresse de la prochaine instruction à exécuter.
- Incrémenté après l'exécution de chaque instruction.
- Peut être changé lors de l'exécution: instruction "jump" .
- Initialiser à zéro ou « reset » au début.

Le registre d'instruction (IR)

C'est un registre de données, il contient une instruction à exécuter.

Unité de commande (UC)

RM: registre Mot

Connecter sur le bus des données, il récupère ou il envoie les données vers les unités fonctionnelles

Le registre SEQ

Le séquenceur qui permet de décoder les instructions

Unité de traitement

ALU: Unité arithmétique et logique qui permet de traiter des instructions arithmétiques et logiques

RT: Registre temporaire qui génère rapidement le programme

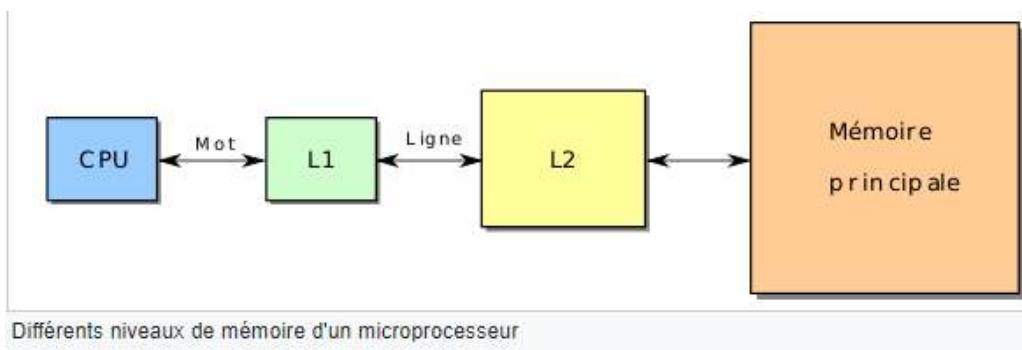
L'unité de traitement

- Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions. L'unité de traitement est Composé de trois principaux unités d'exécution, la première est l'unité arithmétique et logique (UAL) puis deux autres ont été ajoutés qui sont l'unité de calcul en virgule flottante et l'unité multimédia pour des raisons d'optimisation des performances des microprocesseur

Unité de traitement

Mémoire Cache:

- permet de soulager la MC
- Augmenter les performances des processus
- Elle est très rapide, mais aussi très chère. Il s'agit souvent de SRAM.



Une *ligne*

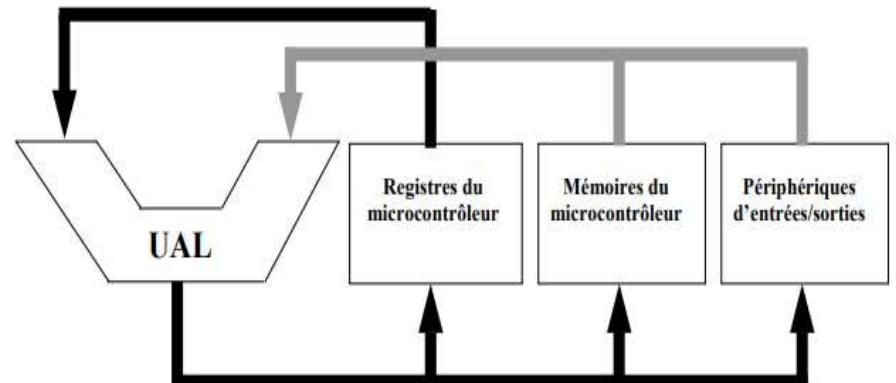
le plus petit élément de données qui peut être transféré entre la mémoire cache et la mémoire de niveau supérieur.

Un *mot*

le plus petit élément de données qui peut être transféré entre le processeur et la mémoire cache.

Unité arithmétique et logique

- Exécute les opérations :
 1. Les opérations arithmétiques : addition, soustraction, changement de signe, etc.
 2. les opérations logiques : compléments à un, à deux, et, ou, ou-exclusif, non, non-et, etc.
 3. les comparaisons : test d'égalité, supérieur, inférieur, et leur équivalents « ou égal ».
 4. éventuellement des décalages et rotations (mais parfois ces opérations sont externalisées).
- Elle contient 3 registres:
 - Un accumulateur: stockage de résultat
 - Deux registres d'entrée
 - Une commande d'opération (choix de l'opération)



Architecture des ordinateurs
Enseignant : Abdellah Amine

Les registres

- Les registres sont des petits composants de stockage intégrés au processeur lui-même. Les registres sont des emplacements de mémoire directement accessibles par le processeur. Les registres contiennent des instructions ou des opérandes auxquels le CPU a actuellement accès.
- Les registres sont les éléments de stockage accessibles à haute vitesse. Le processeur accède aux registres dans un cycle d'horloge du CPU. En fait, le processeur peut décoder les instructions et effectuer des opérations sur le contenu du registre à une cadence de plusieurs opérations par cycle d'horloge du CPU. On peut donc dire que le processeur peut accéder aux registres plus rapidement que la mémoire principale.
- le registre contient les données actuellement traitées par la CPU, tandis que la mémoire contient les instructions de programme et les données nécessaires à son exécution.

Tableau de comparaison

Base de comparaison	registre	Mémoire
De base	Les registres contiennent les opérandes ou les instructions en cours de traitement par la CPU.	La mémoire contient les instructions et les données requises par le programme en cours d'exécution dans la CPU.
Capacité	Register contient la petite quantité de données entre 32 et 64 bits.	La mémoire de l'ordinateur peut aller de quelques Go à plusieurs To.
Accès	La CPU peut fonctionner sur le contenu du registre à la cadence de plusieurs opérations au cours d'un cycle d'horloge.	La CPU accède à la mémoire plus lentement que le registre.
Type	Registre d'accumulateurs, compteur de programmes, registre d'instructions, registre d'adresses, etc.	RAM.

Mémoire cache

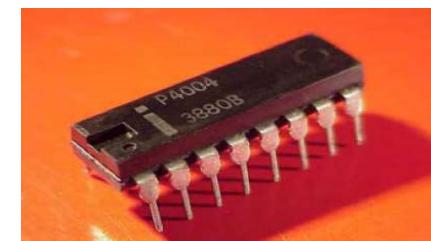
- Une **mémoire cache** ou **antémémoire** est, en informatique, une mémoire qui enregistre temporairement des copies de données provenant d'une autre source de donnée comme les mémoires, afin de diminuer le temps d'accès (en lecture ou en écriture) d'un matériel informatique (en général, un processeur) à ces données. La mémoire cache est plus rapide et plus proche du matériel informatique qui demande la donnée, mais plus petite que la mémoire pour laquelle elle sert d'intermédiaire.
- le niveau 1 (L1) est le type de mémoire cache le plus rapide car il est le plus petit et le plus proche du processeur.
- Le niveau 2 (L2) a une capacité plus élevée mais dispose d'une vitesse plus lente et est situé sur la puce du processeur.
- La mémoire cache de niveau 3 (L3) a la plus grande capacité et est située sur la partie de l'ordinateur qui utilise la mémoire cache L2.

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).

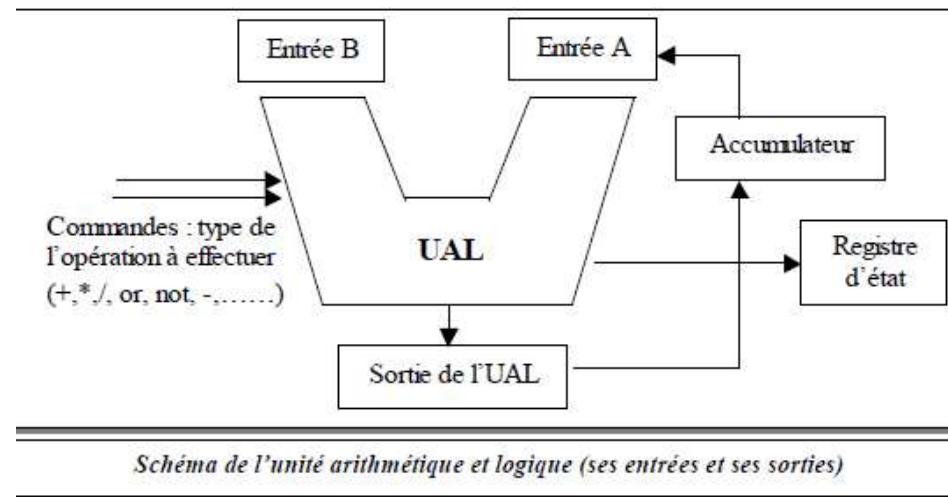
Technologies de fabrication : NMOS, PMOS, CMOS.



L'unité de traitement

Unité arithmétique et logique : (UAL)

Elle est composée de circuits logiques tels que les additionneurs, soustracteurs, comparateurs logiques...etc., afin d'effectuer les calculs et les opérations logiques des différents instructions à exécuter, les données à traiter se présentent aux entrées de l'UAL, sont traités, puis le résultat est fourni en sortie et généralement stocké dans un registre dit accumulateur. Les informations qui concernent l'opération sont envoyées vers le registre d'état.



L'unité de traitement

Unité de calcul en virgule flottante

- C'est une unité qui est capable de réaliser les opérations de calcul pour les réels ainsi que les calculs mathématiques et scientifiques complexes.
- A l'origine la tâche de cette unité était réalisée par tout un processeur à part, en 1989 elle a été intégré dans les microprocesseurs afin d'optimiser les calculs.

Unité multimédia

- C'est une unité qui est chargée d'accélérer l'exécution des programmes multimédia comportant des vidéos, du son, graphisme en 3D etc....
- Cette unité porte le nom de MMX pour les premiers pentium (MultiMedia eXtensions) intégrants des fonctions de gestion du multimédia, de même la technologie 3DNOW pour les AMD et SSE pour les pentiumIII.

Autres unités du microprocesseur

les registres du microprocesseur

Un registre est une zone mémoire à l'intérieur du microprocesseur de faible taille, qui permet de mémoriser des mots mémoires ou des adresses d'une façon temporaire lors de l'exécution des instructions.

Les registres généraux :

- Ce sont des mémoires rapides, à l'intérieur du microprocesseur, qui permettent à l'unité d'exécution de manipuler des données à vitesse élevée. Ils sont connectés au bus données interne au microprocesseur. L'adresse d'un registre est associée à son nom (on donne généralement comme nom une lettre) A,B,C...
- Ces registres permettent de sauvegarder des informations utilisées dans les programmes ou des résultats intermédiaires, cela évite des accès à la mémoire, accélérant ainsi l'exécution des programmes.

Autres unités du microprocesseur

- *Les registres généraux :*

Les registres généraux sont à la disposition du programmeur qui a normalement un choix d'instructions permettant de les manipuler comme

- Chargement d'un registre à partir de la mémoire ou d'un autre registre.
- Enregistrement dans la mémoire du contenu d'un registre.
- Transfert du contenu d'un registre dans l'accumulateur et vice versa.
- Incrémentation ou décrémentation d'un registre

Autres unités du microprocesseur

- *Les registres d'adresses (pointeurs)*

Ce sont des registres connectés sur le bus adresses, leur contenu est une adresse en mémoire centrale.

Il existent plusieurs types On peut citer comme registres:

- Le compteur ordinal (pointeur de programme PC)
- Le pointeur de pile (Stack pointer SP)
- Les registres d'index (Index source SI et index destination DI)

Autres unités du microprocesseur

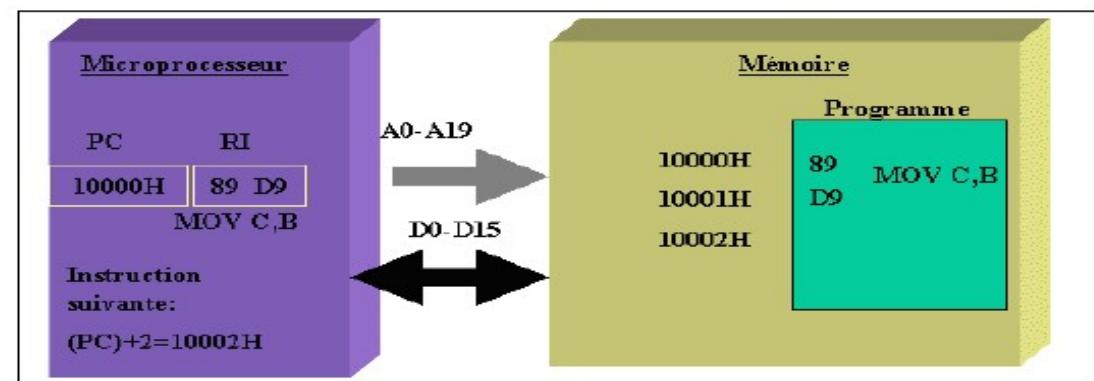
Le compteur ordinal : (pointeur de programme PC)

Il contient l'adresse de l'instruction à rechercher en mémoire. L'unité de commande incrémente le compteur ordinal (PC) du nombre d'octets sur lequel l'instruction, en cours d'exécution, est codée.

Le compteur ordinal contiendra alors l'adresse de l'instruction suivante.

Exemple : (PC)=10000H ; il pointe la mémoire qui contient l'instruction MOV C,B qui est codée sur deux octets (89 D9H) ; l'unité de commande incrémentera de deux le contenu du PC : (PC) = 10002H

(la mémoire sera supposée être organisée en octets).



Autres unités du microprocesseur

Le registre d'instruction

contient l'instruction qui doit être traitée par le microprocesseur, cette instruction est recherchée en mémoire puis placée dans ce registre pour être décodée par le décodeur et préparée pour l'exécution.

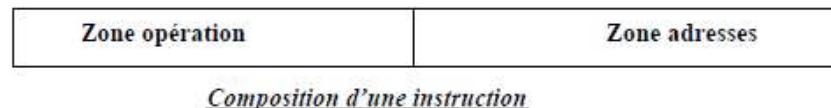
Une instruction est une opération élémentaire d'un langage de programmation, c'est à dire le plus petit ordre que peut comprendre un ordinateur

Autres unités du microprocesseur

Exemple : ADD A,B cette instruction correspond à un ordre donné à l'ordinateur en langage assembleur qui permet de faire l'addition entre les données A et B.

Toute instruction présente en fait deux types d'informations :

- ✓ Ce qu'il faut faire comme action (addition, affichage, division ...)
- ✓ Avec quelles données réaliser cette action.



- Zone opération : Cette zone permet de déterminer la nature de l'opération à effectuer par le microprocesseur.
- Zone adresse : contient les adresses en mémoire des opérandes qui participent dans une opération, dans certains cas elle contient l'opérande même. Il existe plusieurs modes d'adressages pour accéder aux données.

Autres unités du microprocesseur

- **Le registre d'adresse :**

C'est un registre qui contient l'adresse du mot à accéder en mémoire centrale. A chaque accès mémoire, l'adresse recherchée est stockée dans ce registre. Il a la taille d'une adresse qui est la même que celle du bus d'adresses ce qui permet de déterminer le nombre de mots mémoires adressables et l'espace mémoire adressable.

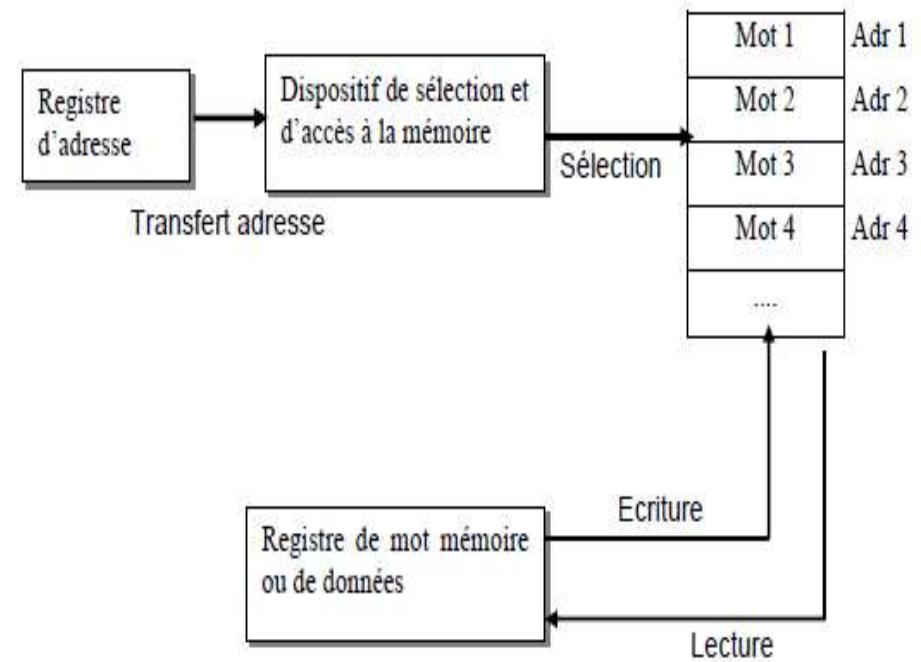
- **Le registre mot mémoire ou registre de données :**

Contient le mot mémoire faisant objet d'une opération de lecture ou d'écriture dans la mémoire centrale. Ce registre a la taille d'un mot mémoire qui est la même que celle des registres de travail et l'accumulateur qui est égale à la taille du bus de données.

Autres unités du microprocesseur

La lecture et l'écriture dans la mémoire centrale se fait de la façon suivante :

- **Lecture** : l'adresse du mot à lire est transférée au registre d'adresse, le dispositif d'accès à la mémoire se charge de chercher le mot et de le mettre dans le registre mot mémoire.
- **Ecriture** : le registre d'adresse est chargé par l'adresse mémoire où on va écrire, le mot à écrire est placé dans le registre mot mémoire. Lorsque l'ordre de l'écriture est donné, le contenu des cases mémoires sera écrasé et remplacé par la nouvelle valeur. Par contre, en cas de lecture, le contenu des cases mémoire n'est pas détruit.



Autres unités du microprocesseur

- **Le registre accumulateur :**

C'est un registre de travail très important de l'UAL, dans la plus part des opérations arithmétiques et logiques, l'accumulateur contient un des opérandes avant l'exécution et le résultat après. Il peut aussi servir de registre tampon dans les opérations d'entrée/sortie. Généralement, l'accumulateur a la même taille qu'un mot mémoire.

- **Le registre d'état: (PSW program status word)**

C'est un registre qui contient les différents bits appelés drapeaux (Flags) indiquant l'état d'une condition particulière dans le CPU. Ces bits peuvent être testés par un programme et donc décider des suites d'actions à prendre.

Autres unités du microprocesseur

- Le nombre de bits de ce registre change d'un microprocesseur à un autre, par exemple pour le cas d'un 8088 ce registre est de 16 bits :

U	U	U	U	O	D	I	T	S	Z	U	AC	U	P	U	C
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

U : undefined désigne un bit non défini.

Les différents bits du registre d'état sont :

- Bit de retenu** : c'est le bit C ou carry qui est le bit n°0. ce bit est à 1 s'il y a une retenu qui est générée lors d'une opération arithmétique.
- Bit de parité** : le bit P n°2, ce bit est mis à 1 si le nombre de 1 dans l'accumulateur est pair.
- Bit de retenu intermédiaire** : le bit AC (auxiliary carry), c'est le bit n°4, il est mis à 1 si une retenu est générée entre groupes de 4 bits (passage de retenu).
- Le bit zéro** : c'est le bit n°6, il est à 1 si le résultat d'une opération arithmétique est nul.
- Le bit de signe** : bit n°7, il est à 1 si le résultat d'une opération arithmétique est négatif.

Autres unités du microprocesseur

- ***Le bit de dépassement de capacité*** : le bit n°11, O pour overflow, il est à 1 s'il y a dépassement de capacité dans les opérations arithmétiques. Par exemple sur 8 bits en complément à 2, les nombres binaires sont codés sur 8 bits. On peut coder les nombres variants de -128 (10000000) à +127(0111111). Si on fait $107+28$ cela donne 135 qui ne peut pas être représenté, d'où génération de débordement.

Autres unités du microprocesseur

Exemple 1: addition de nombres binaire sur 8 bits

$$\begin{array}{r} 11111100 \\ + 10000010 \\ \hline \text{carry : } 1 = 01111110 \end{array} \quad \begin{array}{r} \text{FCH} \quad + 82H \quad = 17EH \\ (252)_{10} + (130)_{10} = (382)_{10} \end{array}$$

Exemple 2 : La **bascule C (carry)** sert aussi à capter le bit expulsé lors d'une opération de **décalage ou de rotation**

Décalage à gauche d'un bit de cet octet : **10010110**

la carry recueille le **1** du bit de poids fort carry : **1 00101100**

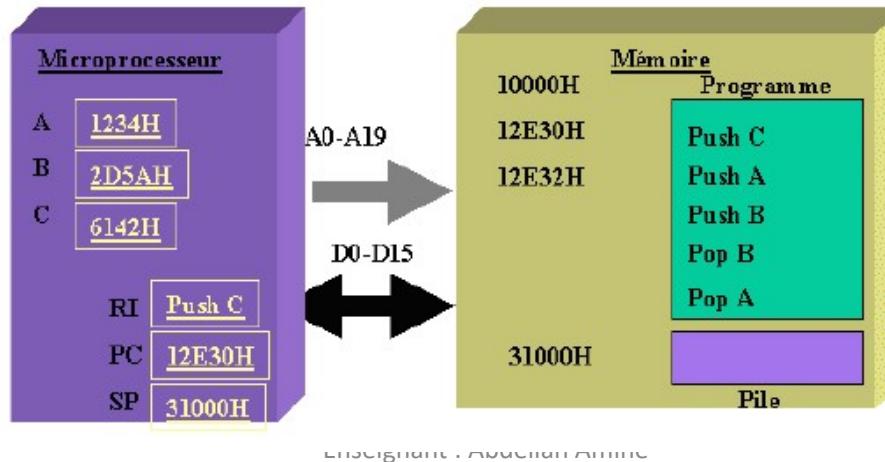
Exemple 3 : overflow

$$\begin{array}{r} 104 \quad 0110\ 1000 \\ + 26 \quad + \underline{0001\ 1010} \\ \hline = 130 \quad = 1000\ 0010 \end{array} \quad \begin{array}{r} - 18 \quad 1110\ 1110 \\ - \underline{118} \quad \underline{1000\ 1010} \\ \hline - 136 \quad 0111\ 1000 \end{array} \quad \begin{array}{l} (-126) \quad (-136) \quad (120) \text{ avec C=1} \end{array}$$

L'**indicateur débordement** est une fonction logique (**OU exclusif**) de la retenue (**C**) et du signe (**S**).

Autres unités du microprocesseur

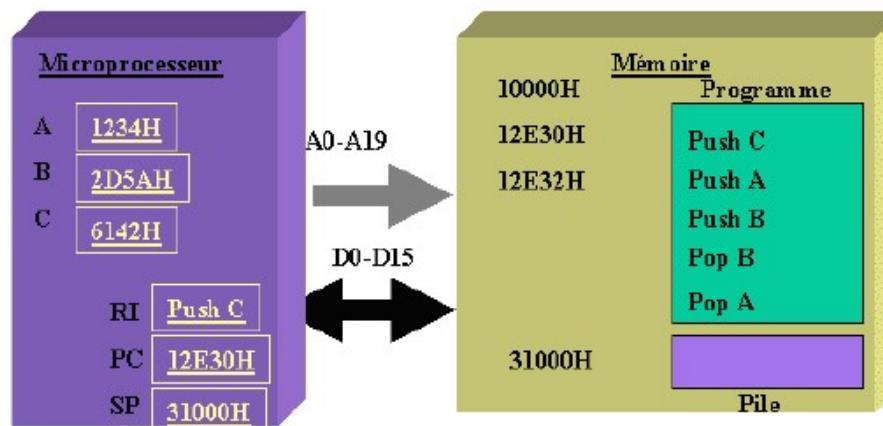
- **Le registre pointeur de pile : (Stack Pointer)**
- Il contient l'adresse de la pile. Celle-ci est une partie de la mémoire, elle permet de stocker des informations (le contenu des registres) relatives au traitement des interruptions et des sous-programmes
- La pile est gérée en **LIFO** : (Last IN First Out) dernier entré premier sorti. Le fonctionnement est identique à une pile d'assiette
- Le pointeur de pile SP pointe le haut de la pile (31000H dans le schéma), il est décrémenté avant chaque empilement, et incrémenté après chaque dépilement.
- Il existe deux instructions pour empiler et dépiler : PUSH et POP.



Autres unités du microprocesseur

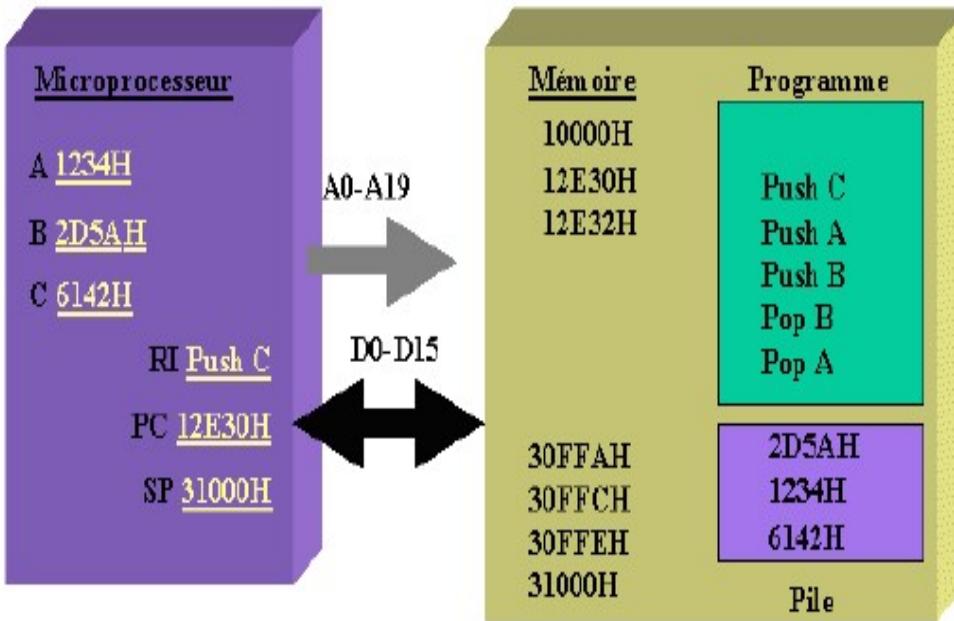
exemple:

PUSH A empilera le registre A et POP A le dépilerà.



Question? Que se passera-t-il durant l'exécution du programme commençant en 12E30H? Que vaudra SP et que contiendra la pile à cette adresse, à la fin du programme ?

Autres unités du microprocesseur



Réponse. Le programme commence par sauvegarder le contenu de C dans la pile (PUSH C). Pour cela (SP) est décrémenté de deux ($(SP)=31000H-2=30FFE$ H), puis on effectue l'écriture de (C) dans la mémoire à l'adresse (SP) : ($30FFE$ H) = 6142H.

Pour PUSH A on obtient : ($30FFCH$) = 1234H, et pour PUSH B : ($30FFAH$) = 2D5AH.

Pour l'instruction POP B, ((SP)) est chargé dans le registre B ($(SP)=30FFAH$; $(B)=2D5AH$) puis (SP) est incrémenté de deux ($(SP)=30FFAH+2=30FFCH$).

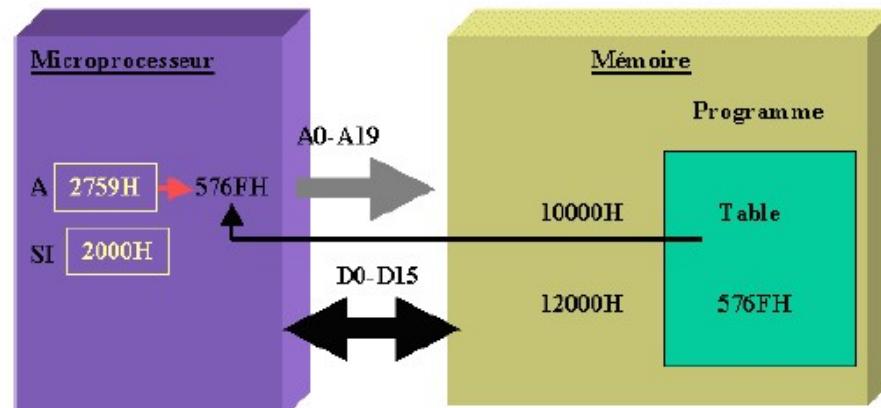
Enfin, pour POP A on obtient : ($A=1234H$ et $(SP)=30FFCH + 2 = 30FFE$ H)

Les registres d'index (index source SI et index destination DI)

- Les registres d'index peuvent être utilisés comme des registres généraux pour sauvegarder et pour compter. Mais en plus ils ont une fonction spéciale qui est de grande utilité dans la manipulation des tableaux de données. Ils peuvent en effet être utilisés pour manipuler des adresses et permettent de mémoriser une adresse, suivant une forme particulière d'adressage, appelée adressage indexée

Exemple :

MOV A,[SI+10000H] place le contenu de la mémoire d'adresse 10000H+le contenu de SI, dans le registre A.



JEU D'INSTRUCTIONS

- Chaque processeur possède un certain nombre limité d'instructions qu'il peut exécuter.
- Ces instructions s'appellent **jeu d'instructions**.
- Le jeu d'instructions décrit l'ensemble des opérations élémentaires que le processeur peut exécuter.
- Les instructions peuvent être classifiées en 4 catégories :

Instruction d'affectation

Instructions arithmétiques et logiques (ET , OU , ADD,...)

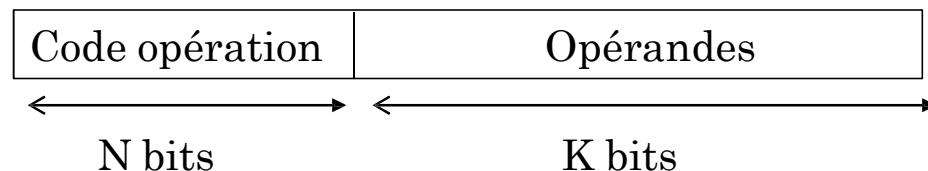
Instructions de branchement (conditionnelle et inconditionnelle)

Instructions d'entrées sorties.

FORMAT D'INSTRUCTION JEU D'INSTRUCTIONS

- ❖ L'instruction est composée de deux champs :

- **Code d'opération** représentant l'action que le processeur doit accomplir.
- **Champ des opérandes** définissant les paramètres de l'action. Un opérande peut s'agir d'une donnée ou bien d'une adresse mémoire.



- ❖ La taille d'une instruction dépend du type de l'instruction et du type de l'opérande.
- ❖ Les instructions et leurs opérandes sont stockés dans la mémoire.

EXEMPLE DU FORMAT D'INSTRUCTION JEU D'INSTRUCTIONS

- ❖ **Instruction à trois opérandes:** Il faut préciser le premier opérande, le deuxième opérande et l'emplacement du résultat

Code opération	Opérande1	Opérande2	Résultat
----------------	-----------	-----------	----------

Exemple: ADD A,B, C

ADD	A	B	C
-----	---	---	---

$$C \leftarrow A + B$$

La taille de l'instruction est grand → Pratiquement il n'existent pas d'instruction de ce type.

EXEMPLE DU FORMAT D'INSTRUCTION JEU D'INSTRUCTIONS

- ❖ **Instruction à deux opérandes:** Il faut préciser le premier opérande et le deuxième opérande. Le résultat est implicitement mis dans le deuxième opérande .

Code opération	Opérande1	Opérande2
----------------	-----------	-----------

Exemple: ADD A,B

ADD	A	B
-----	---	---

$$B \leftarrow A + B$$

EXEMPLE DU FORMAT D'INSTRUCTION JEU D'INSTRUCTIONS

- ❖ **Instruction à un opérande:** il faut préciser uniquement le deuxième opérande. Le premier opérande existe dans le registre accumulateur. Le résultat est mis dans le registre accumulateur.

Code opération	Opérande
----------------	----------

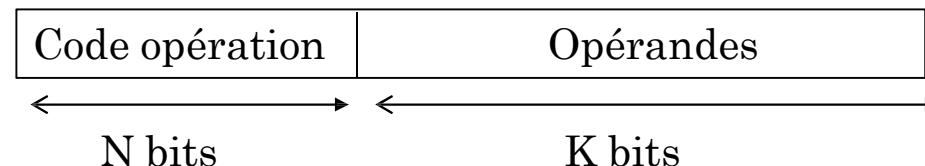
Exemple ADD B

ADD	B
-----	---

$ACC \leftarrow ACC + B$

Ce type d'instruction est le plus utilisé.

MODE D'ADRESSAGE JEU D'INSTRUCTIONS



- ❖ Le mode d'adressage définit la manière dont le processeur va accéder à l'opérande.
- ❖ Le code opération comporte un ensemble de bits pour indiquer le mode d'adressage.
- ❖ Les modes d'adressage les plus utilisés sont : immédiat, direct, indirect, indexé et relatif.

ADRESSAGE IMMÉDIAT JEU D'INSTRUCTIONS

- ❖ La valeur de l'opérande existe dans le **champ opérande** de l'instruction

Code opération	valeur
----------------	--------

Exemple :

ADD	150
-----	-----

ADD 150

Cette commande va avoir l'effet suivant : ACC \leftarrow ACC+ 150

Si le registre accumulateur contient la valeur 200 alors
après l'exécution son contenu sera égale à 350

ADRESSAGE DIRECT JEU D'INSTRUCTIONS

- ❖ Le champs opérande contient **l'adresse de l'opérande** (emplacement en mémoire)

Code opération	adresse de l'opérande
----------------	-----------------------
- ❖ Pour réaliser l'opération, il faut récupérer (lire) l'opérande à partir de la mémoire.

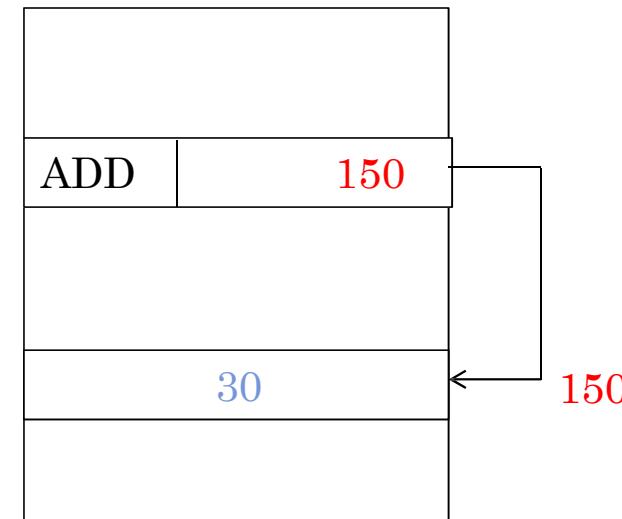
Exemple :

ADD 150

Cette commande va avoir l'effet suivant :

ACC \leftarrow ACC+ (150)

ACC \leftarrow ACC+ 30



ADRESSAGE INDIRECT JEU D'INSTRUCTIONS

- Le champs opérande contient l'**adresse de l'adresse de l'opérande**.

Code opération	adresse d'adresse de l'opérande
----------------	---------------------------------

- Pour réaliser l'opération, il faut d'abord récupérer l'adresse de l'opérande à partir de la mémoire, ensuite, chercher l'opérande à partir de la mémoire.

Exemple :

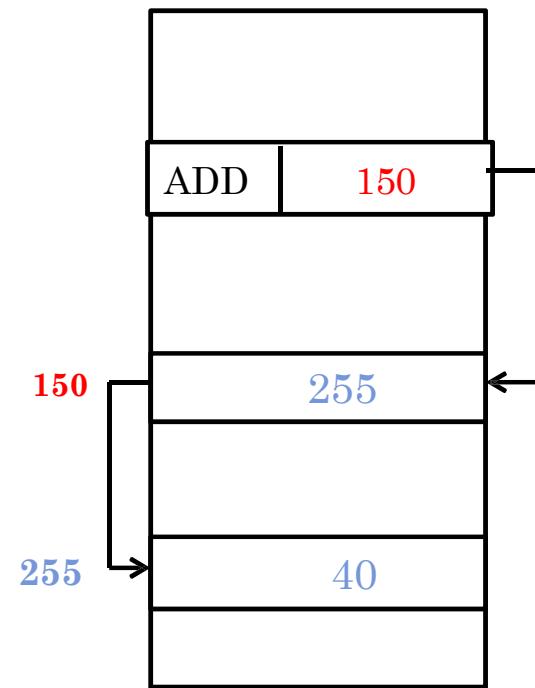
ADD 150

Cette commande va avoir l'effet suivant :

$ACC \leftarrow ACC + ((150))$

$ACC \leftarrow ACC + (255)$

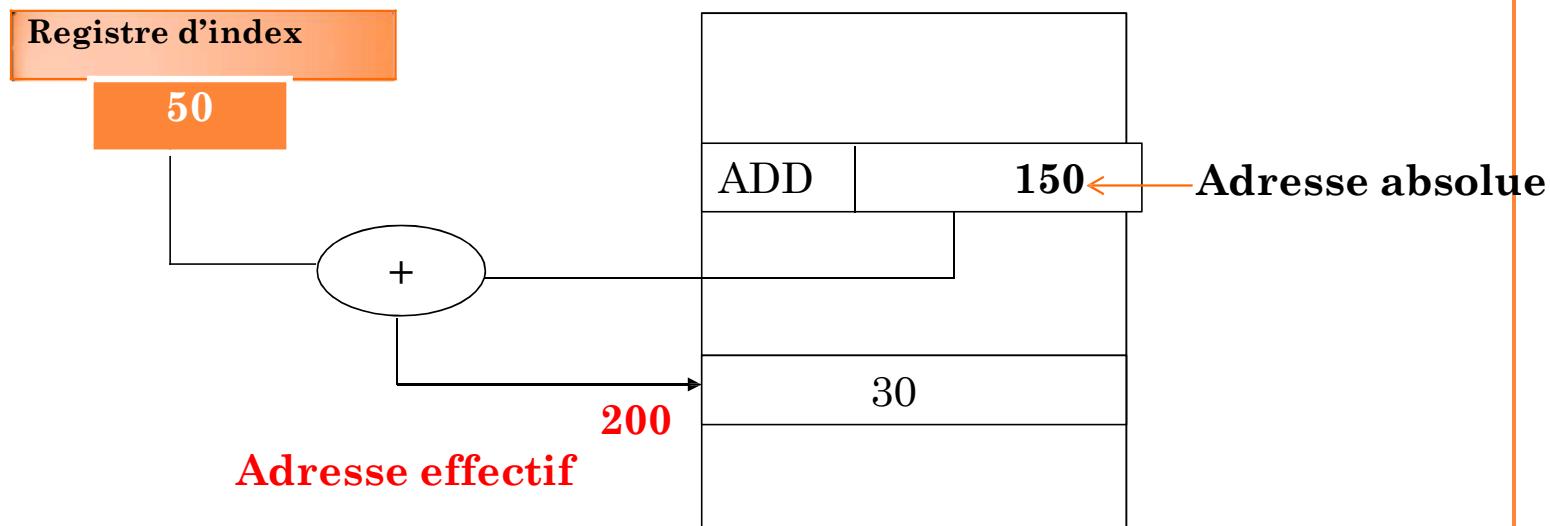
$ACC \leftarrow ACC + 40$



ADRESSAGE INDEXÉ JEU D'INSTRUCTIONS

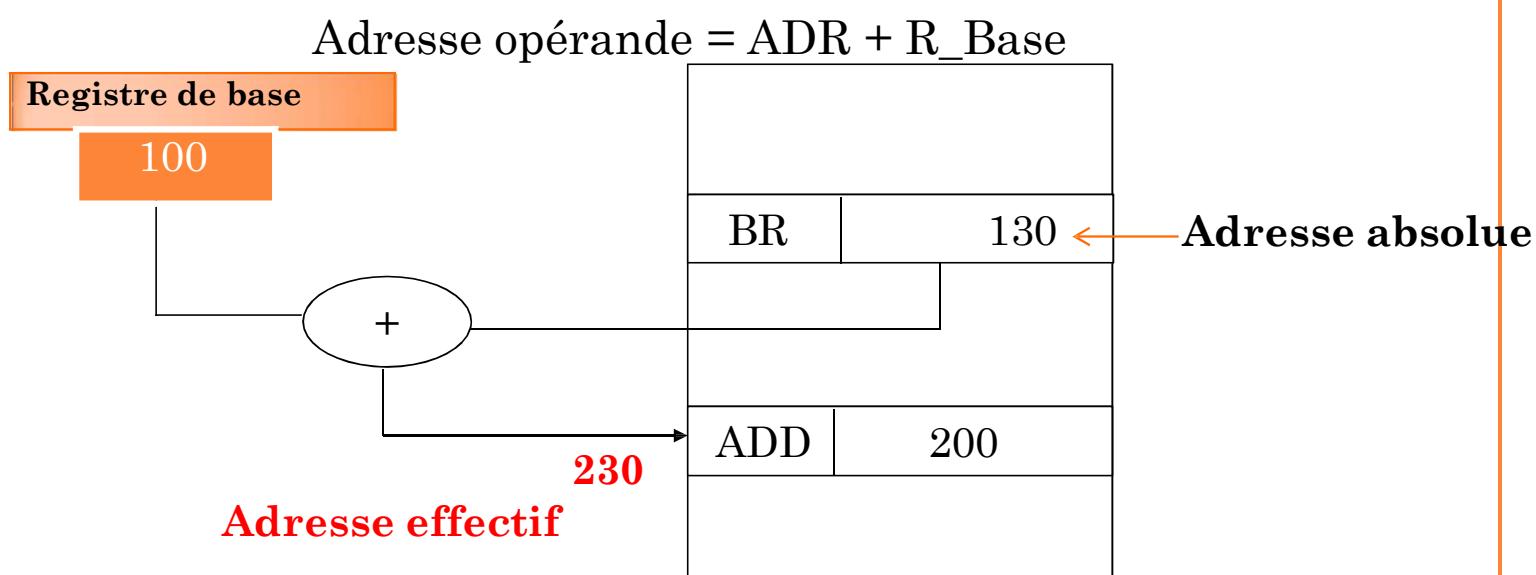
- Le champs opérande contient l'**adresse absolue de l'opérande**. L'adresse effectif de l'opérande est relatif à une zone mémoire. L'adresse de cette zone se trouve dans un registre spécial appelé **registre index**.
- Adresse opérande = ADR + R_Index

Code opération	adresse absolue de l'opérande
----------------	-------------------------------



ADRESSAGE RELATIF JEU D'INSTRUCTIONS

- Le champs opérande contient l'**adresse absolue de l'opérande**. L'adresse effectif de l'opérande est relatif à une zone mémoire. L'adresse de cette zone se trouve dans un registre spécial appelé **registre de base**.
- Ce mode d'adressage est utilisée pour les instructions de branchement.



Etapes d'exécution d'une instruction dans le microprocesseur

Etape 1: Recherche de l'instruction à traiter

1. Le PC contient l'adresse d'instruction qui sera placée sur le bus d'adresse
2. Le contenu de la case mémoire sélectionné est disponible sur le bus de données
3. L'instruction est stockée dans le registre RI

Etapes d'exécution d'une instruction dans le microprocesseur

Etape 2: Décodage de l'instruction et recherche de l'opérande

4. L'instruction est décodée sur plusieurs mots, le premier mot est le code opératoire qui définit l'opération
5. L'unité de commande transforme l'instruction à l'unité de traitement
6. Si l'instruction nécessite une donnée de la mémoire, l'unité de commande récupère sa valeur sur le bus de données. L'identification de l'adresse a travers le bus d'adresse.
7. Récupération de la valeur par le bus de données
8. l'opérand est stocké dans un registre (UAL)

Etapes d'exécution d'une instruction dans le microprocesseur

Etape 3: Exécution de l'instruction

9. Faire le calcul par UAL
10. Les drapeaux, les indicateurs ou les flags sont positionnées dans le registre d'état.
11. L'unité de commande positionne le PC pour l'instruction suivante

DÉROULEMENT DE L'INSTRUCTION D'ADDITION EN MODE DIRECT

ÉTAPES D'EXÉCUTION D'UNE INSTRUCTION

Phase 1 : Charger l'instruction

- $\text{RAM} \leftarrow \text{CO}$
- Lecture
- $\text{RI} \leftarrow \text{RIM}$

Phase 2: Décoder l'instruction

Phase 3: Charger l'opérande

- $\text{RAM} \leftarrow \text{ADR}$
- Lecture
- $\text{RD} \leftarrow \text{RIM}$

Phase 4: Exécuter l'instruction

- $\text{ACC} \leftarrow \text{ACC} + \text{RD}$

Phase 5: Passer à l'instruction suivante

- $\text{CO} \leftarrow \text{CO} + 1$

III. Les mémoires

Mémoires

- Vecteur dont chaque composante est accessible par une adresse.
- Les données stockées peuvent être consultées et utilisées chaque fois que nécessaire par le CPU pour le traitement nécessaire.
- Cette unité de mémoire généralement désignée comme zone de stockage primaire
- L'unité de mesure pour la mémoire est l'octet.

Octet est l'espace nécessaire pour stocker 8 bits c.-à-d. espace pour stocker n'importe quel caractère (A-Z, a-z, 0-9, @, !, #, ...)

Types de mémoires

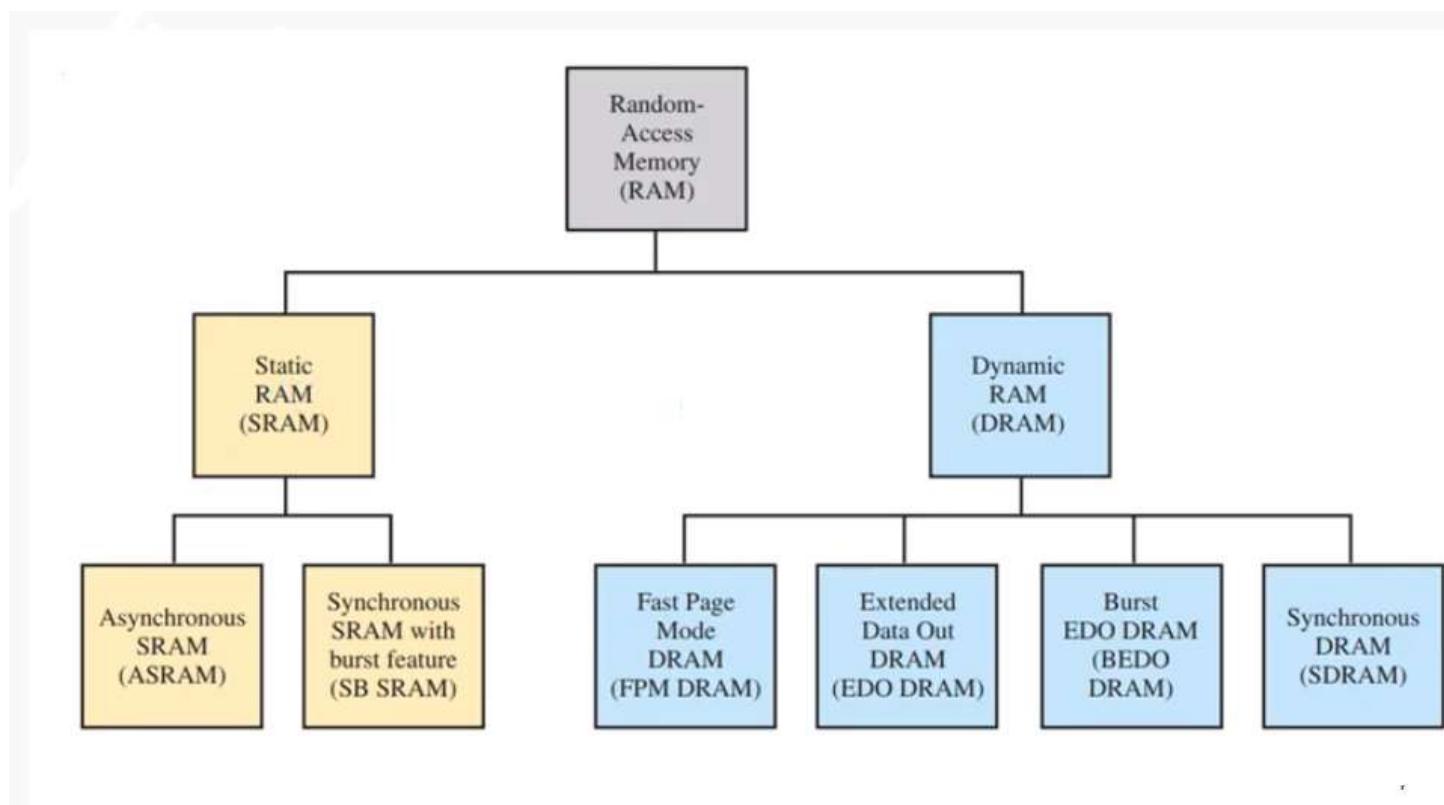
Il y a donc trois types de mémoires :

- La mémoire vive :
 - ✓ Mémoire rapide (appelée génériquement la RAM)
 - ✓ "Petite" capacité
 - ✓ Perd son information lorsqu'on coupe le courant
 - ✓ exemple : SDRAM, SIMM, DIMM, DDRAM, etc.
- La mémoire morte:
 - ✓ Mémoire rapide (appelée génériquement la ROM)
 - ✓ *Mémoire non volatile*
 - ✓ Elle conserve les données nécessaires au démarrage de l'ordinateur
 - ✓ Exemple : ROM, EPROM, EEPROM
- La mémoire de masse :
 - ✓ Mémoire "*lente*" mais grande capacité
 - ✓ N'a pas besoin de courant pour garder l'information
 - ✓ exemple : Disquette, Disque Dur, Clé USB, CD-ROM, DVD

RAM

- Nous savons qu'il existe deux types distincts de mémoire : les mémoires vives et les mémoires mortes.
- Une mémoire vive sert au stockage temporaire de données. Elle doit avoir un temps de cycle très court pour ne pas ralentir le microprocesseur. Les mémoires vives sont en général volatiles : elles perdent leurs informations en cas de coupure d'alimentation. (Certaines d'entre elles, ayant une faible consommation, peuvent être rendues non volatiles par l'adjonction d'une batterie.) Il existe deux grandes familles de mémoires RAM (Random Acces Memory : mémoire à accès aléatoire) :
 - 1 Les RAM statiques ;
 - 2 Les RAM dynamiques.

La famille RAM



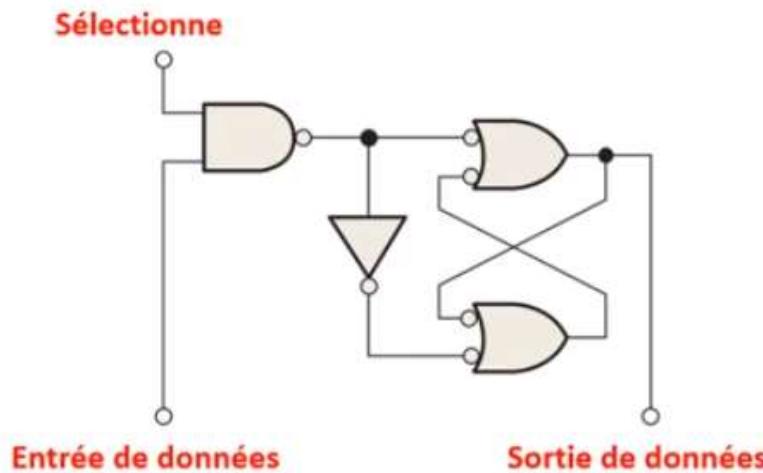
RAM statique

- Le bit mémoire d'une RAM statique (SRAM) est composé d'une bascule (composant électronique élémentaire). Chaque bascule contient entre quatre et six transistors.
- SRAM est un type de mémoire vive utilisant des bascules pour mémoriser les données. Mais contrairement à la mémoire dynamique, elle n'a pas besoin de rafraîchir périodiquement son contenu.
- SRAM est plus onéreuse et moins dense, mais beaucoup moins énergivore et plus rapide que la mémoire dynamique.



RAM Statique

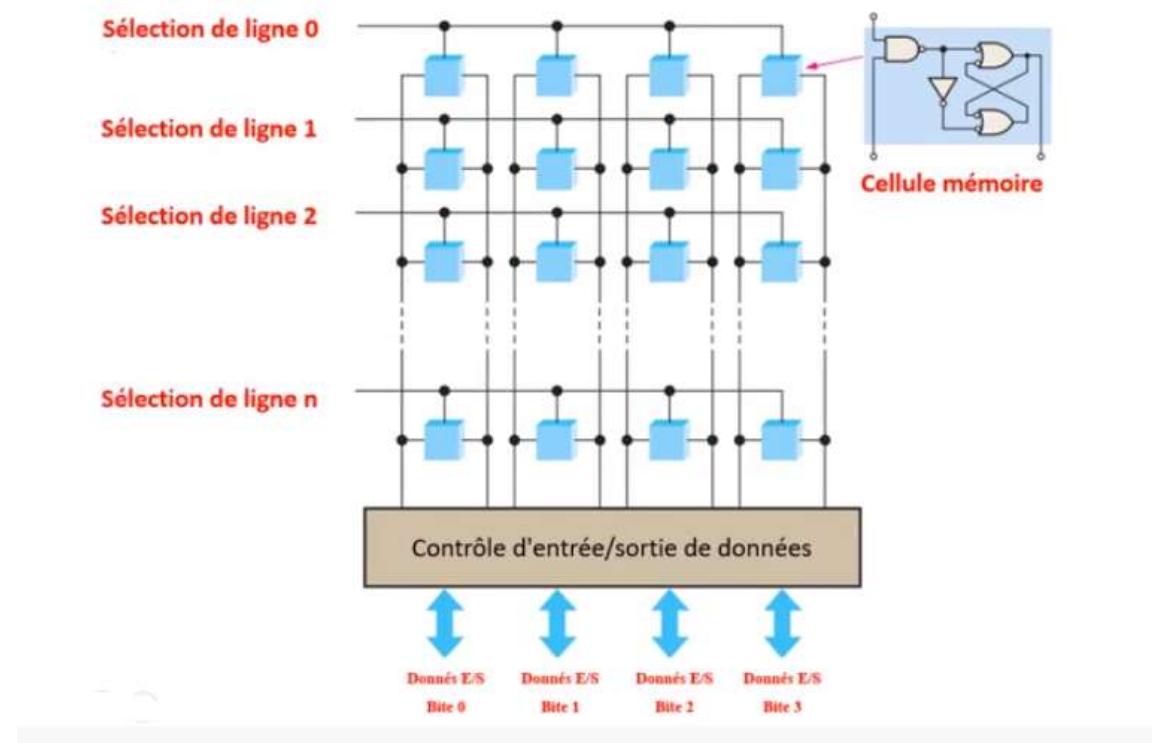
Toutes les SRAM sont caractérisées par des cellules de mémoire à bascule. Tant que le courant continu est appliqué à une cellule de mémoire statique, elle peut conserver un état 1 ou 0 indéfiniment. Si l'alimentation est coupée, le bit de données stocké est perdu.



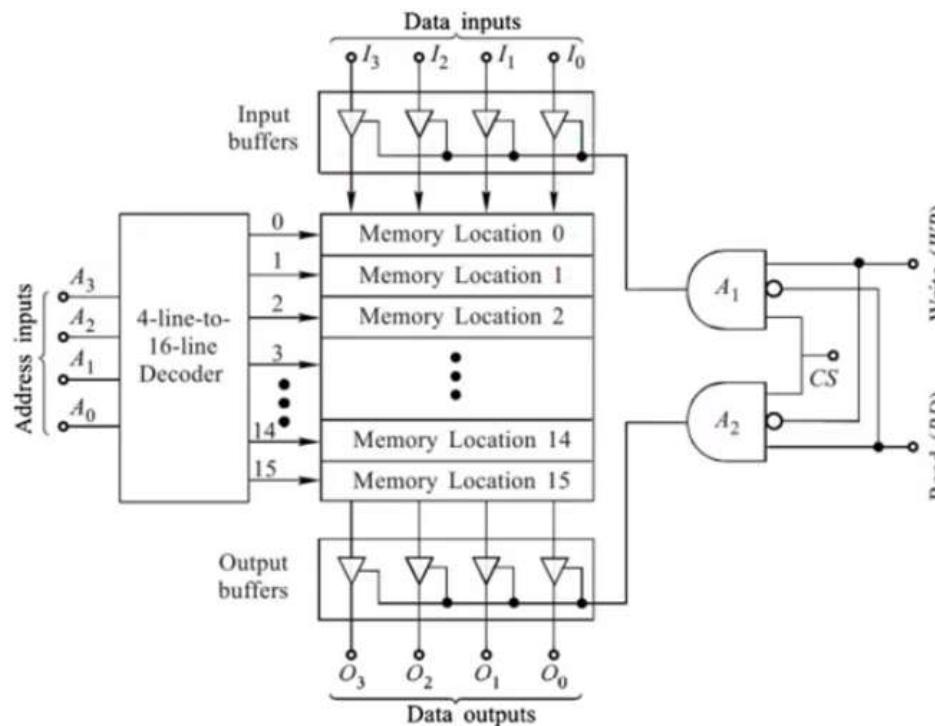
RAM dynamique

- Dans les RAM dynamiques (DRAM), l'information est mémorisée sous la forme d'une charge électrique stockée dans un condensateur.
- **Avantages** : Cette technique permet une plus grande densité d'intégration, car un point mémoire nécessite environ quatre fois moins de transistors que dans une mémoire statique. Sa consommation s'en retrouve donc aussi très réduite ;
- **Inconvénient** : La présence de courants de fuite dans le condensateur contribue à sa décharge. Ainsi, l'information est perdue si on ne la régénère pas périodiquement (charge du condensateur). Les RAM dynamiques doivent donc être rafraîchies régulièrement pour entretenir la mémorisation : il s'agit de lire l'information et de la recharger.

Organisation SRAM asynchrone de base



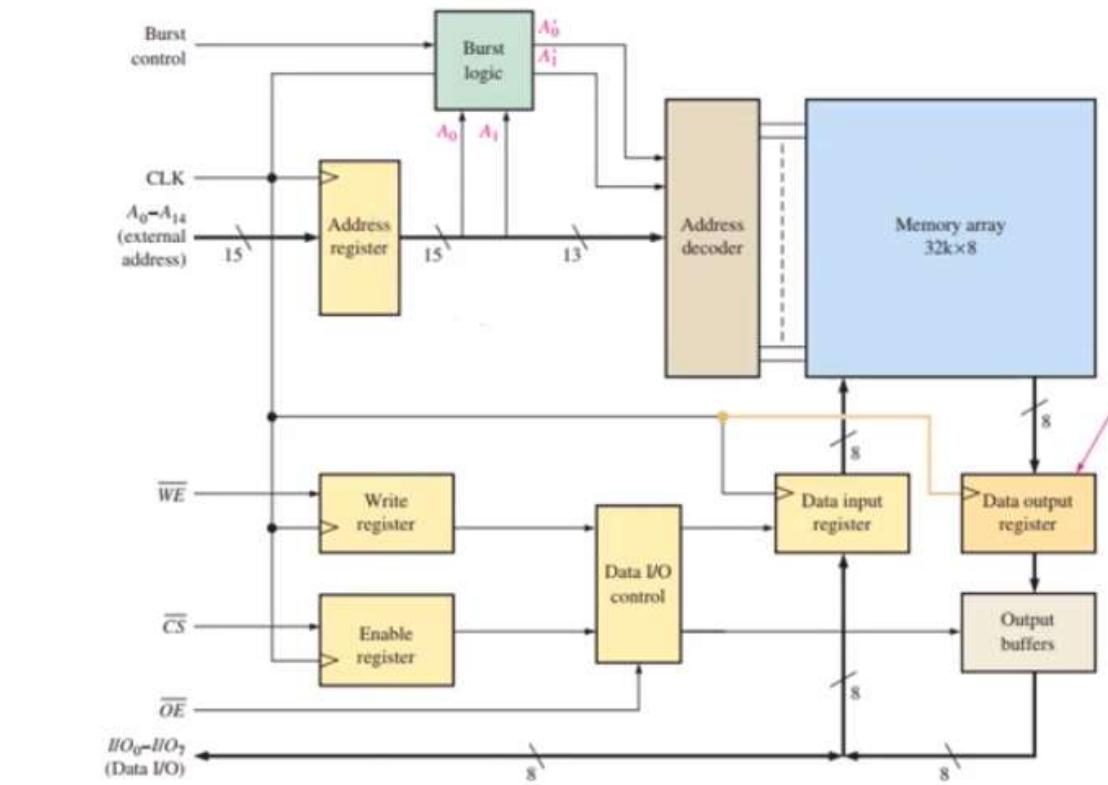
Bus de contrôle



SRAM synchrone

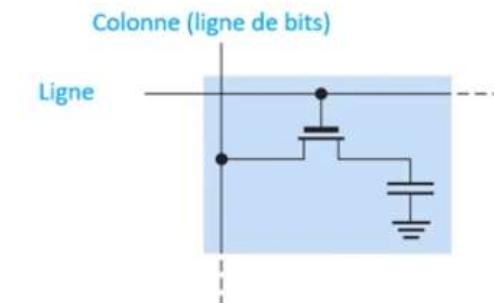
Contrairement à la SRAM asynchrone, une SRAM synchrone est synchronisée avec l'horloge système. Par exemple, dans un système informatique, la SRAM synchrone fonctionne avec le même signal d'horloge qui fait fonctionner le microprocesseur de sorte que le microprocesseur et la mémoire sont synchronisés pour un fonctionnement plus rapide.

Un schéma fonctionnel de base d'une SRAM synchrone

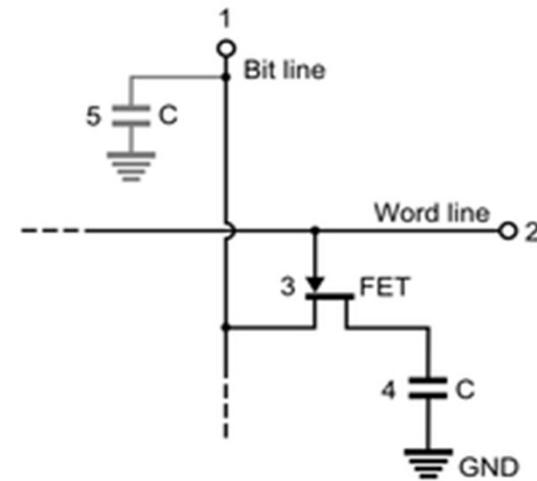


RAM Dynamique DRAM

Les cellules de mémoire dynamique stockent un bit de données dans un petit condensateur plutôt que dans un verrou. L'avantage de ce type de cellule est qu'elle est très simple, permettant ainsi de construire de très grands plans mémoires sur une puce à moindre coût par bit. L'inconvénient est que le condensateur de stockage ne peut pas conserver sa charge sur une période de temps prolongée et perdra le bit de données stocké à moins que sa charge ne soit rafraîchie périodiquement. Le rafraîchissement nécessite des circuits de mémoire supplémentaires et complique le fonctionnement de la DRAM. La figure montre une cellule DRAM typique constituée d'un seul transistor MOS (MOSFET) et d'un condensateur.

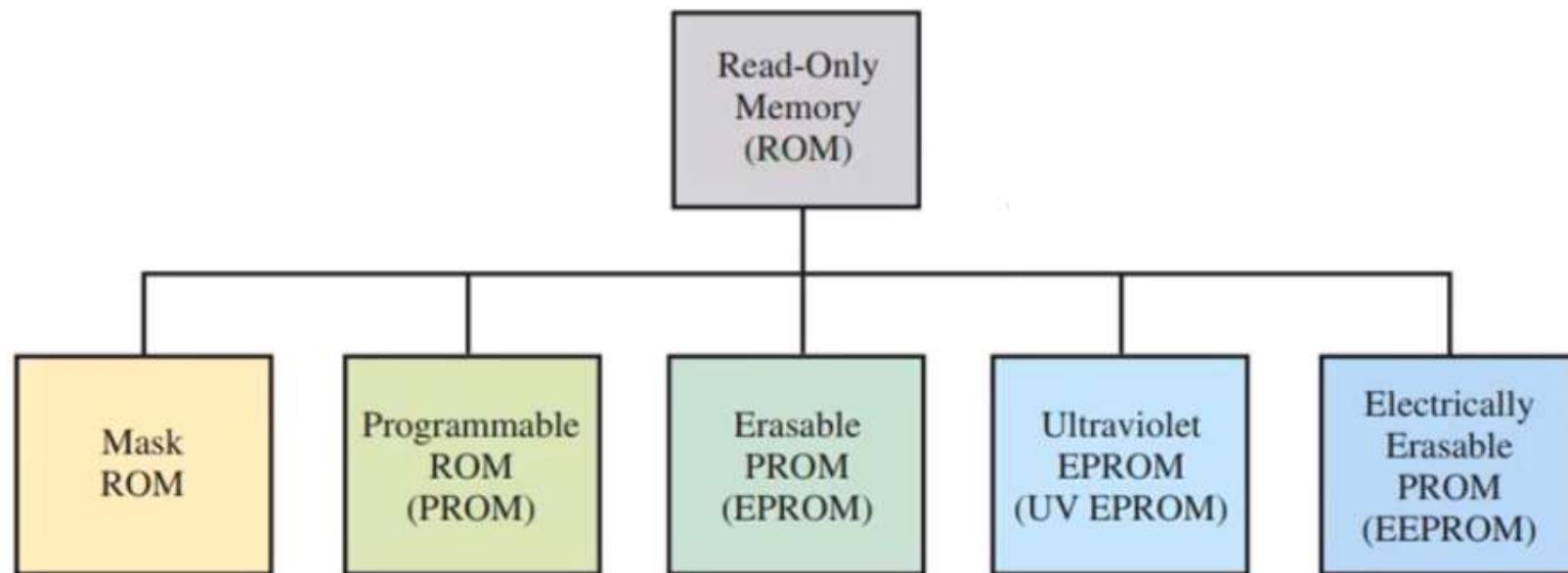


RAM dynamique



Cellule DRAM

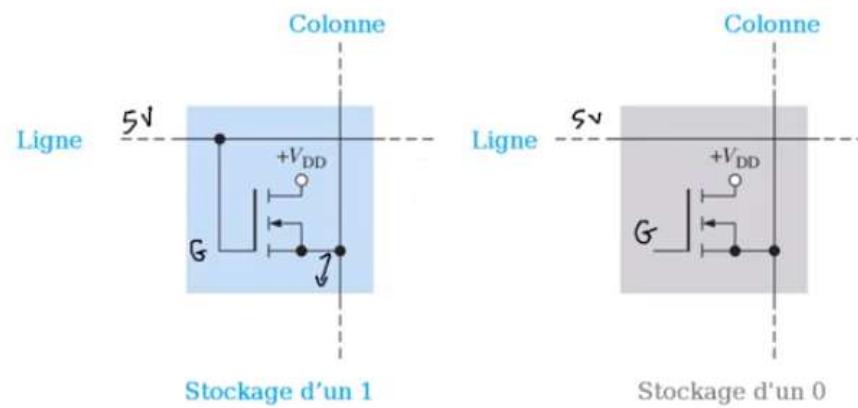
La famille ROM



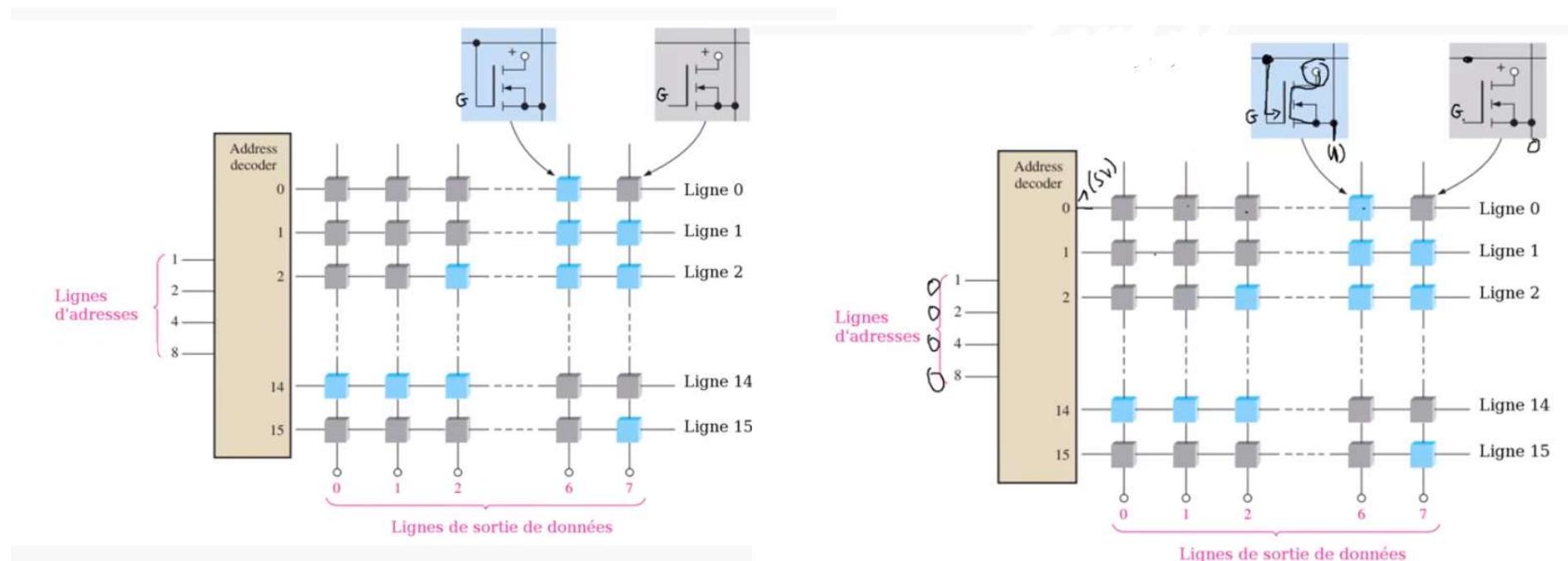
ROM

La ROM de masque est généralement appelée simplement ROM. Il est programmé en permanence pendant le processus de fabrication pour fournir des fonctions standard largement utilisées, telles que des conversions populaires, ou pour fournir des fonctions spécifiées par l'utilisateur. Une fois la mémoire programmée, elle ne peut plus être modifiée. La plupart des ROM IC utilisent la présence ou l'absence d'une connexion de transistor à une jonction rangée/colonne pour représenter un 1 ou un 0.

cellules ROM.



Une représentation d'un tableau de ROM 16 * 8 bits.

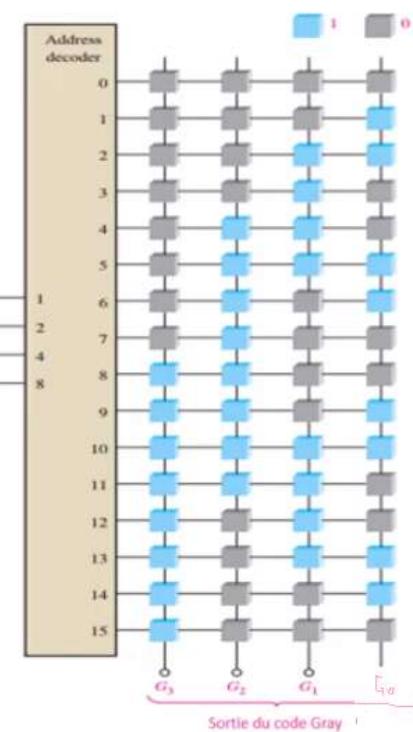


Exemple

Représentation d'une ROM programmée comme un convertisseur de code binaire vers Gray.

Binary				Gray			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Code binaire appliqu  aux lignes d'entr e d'adresse



ROM

- Pour certaines applications, il est nécessaire de pouvoir conserver des informations de façon permanente même lorsque l'alimentation électrique est interrompue. On utilise alors des mémoires mortes ou mémoires à lecture seule (ROM : Read Only Memory). Ces mémoires sont non volatiles. Ces mémoires, contrairement aux RAM, ne peuvent être que lue. L'inscription en mémoire des données restent possible mais est appelée programmation. Suivant le type de ROM, la méthode de programmation changera. Il existe donc plusieurs types de ROM :
 1. ROM ;
 2. PROM ;
 3. EPROM ;
 4. EEPROM ;
 5. FLASH EPROM.

EPROM

- La mémoire EPROM (de l'anglais Erasable Programmable Read-Only Memory signifiant mémoire morte reprogrammable) est un type de mémoire dite morte dont le code peut être modifié dans certaines conditions.



PROM

- Une mémoire morte programmable (en anglais PROM pour Programmable Read Only Memory, Field Programmable Read Only Memory, FPROM) ou One Time Programmable OTP) est un type de mémoire morte qui peut être programmée une seule fois.
- La caractéristique de la mémoire PROM est qu'elle ne peut être programmée qu'une fois. Une fois programmée, elle devient une mémoire morte (ROM) au sens littéral du terme.
- Une mémoire morte est une mémoire utilisée pour enregistrer des informations qui ne doivent pas être perdues lorsque l'appareil, qui les contient, n'est plus alimenté en électricité.

EEPROM

- La mémoire EEPROM (*Electrically-Erasable Programmable Read-Only Memory* ou mémoire morte effaçable électriquement et programmable) (aussi appelée E2PROM ou E²PROM) est un type de mémoire morte.
- Le contenu de la mémoire EEPROM peut être facilement effacé et réécrit à l'aide d'un courant électrique.



Mémoire flash

- La **mémoire flash** est une mémoire de masse à semi-conducteurs réinscriptible, c'est-à-dire une mémoire possédant les caractéristiques d'une mémoire vive mais dont les données ne disparaissent pas lors d'une mise hors tension. La mémoire flash stocke dans des cellules de mémoire les bits de données qui sont conservées lorsque l'alimentation électrique est coupée.

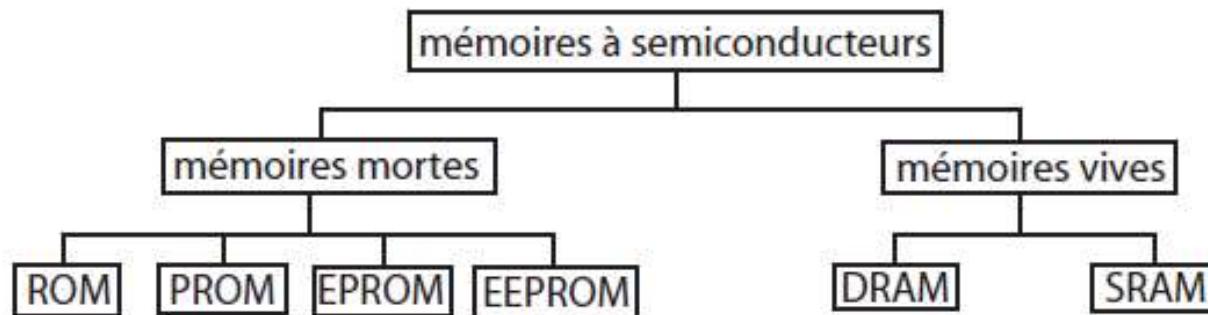


Classification des mémoires (suite)

• Mémoires vives :

- **SRAM** : Static Random Access Memory. Mémoire statique à accès aléatoire, à base de bascules à semiconducteurs 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.

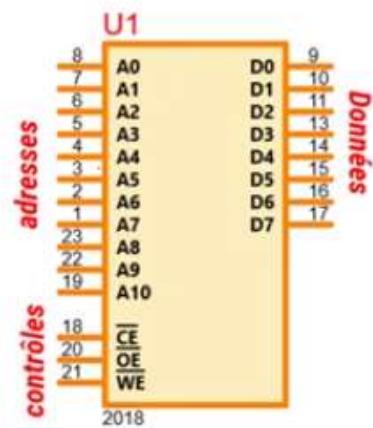
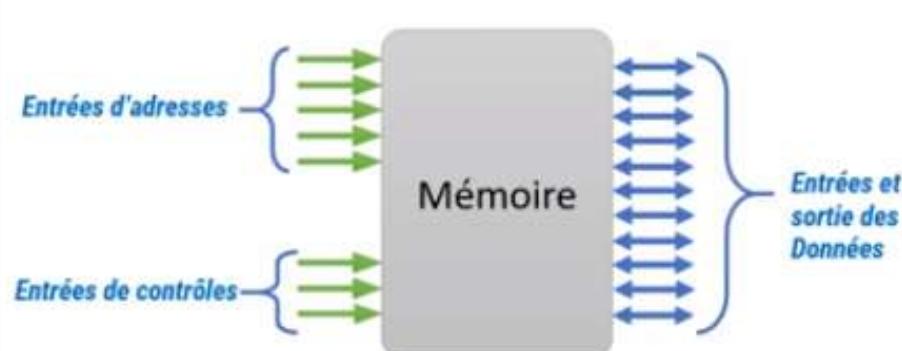
Inconvénient : nécessite un rafraîchissement périodique à cause du courant de fuite des condensateurs. Application : réalisation de la mémoire vive des ordinateurs (barettes mémoire SIMM : Single In-line Memory module).



Classification des mémoires (suite)

- Il y a des échanges continus entre la RAM et les mémoires de masse
- Par exemple :
 - Au démarrage de l'ordinateur on transfert des programmes de la mémoire de masse vers la mémoire vive
 - Lorsque vous enregistrez un fichier (par exemple de votre traitement de texte), vous transférez des données (ce que vous avez tape) de la mémoire vive vers la mémoire de masse

Les mémoires à semi-conducteurs sont des dispositifs de stockage d'informations numériques. Ils sont utilisés dans tous les appareils, y compris les microprocesseurs et également dans la mise en œuvre de circuits logiques programmables.



Mémoires (semi-conducteur)



Mémoires Mortes

(les mémoires non volatiles)

ROM

Mémoires Vives

(les mémoires volatiles)

RAM

– **les mémoires non volatiles** : ces mémoires conservent des informations même en l'absence d'alimentation électrique.

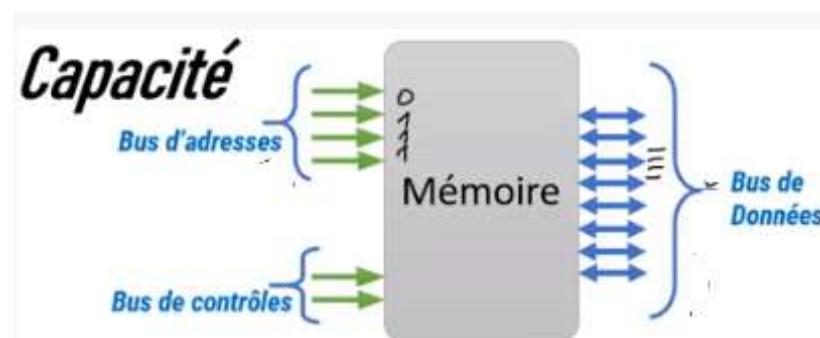


– **mémoire volatile** : ce sont des mémoires dont l'intégrité des informations n'est garantie que si elles sont alimentées électriquement. Ils sont lisibles et inscriptibles.



Adresse mémoire et capacité

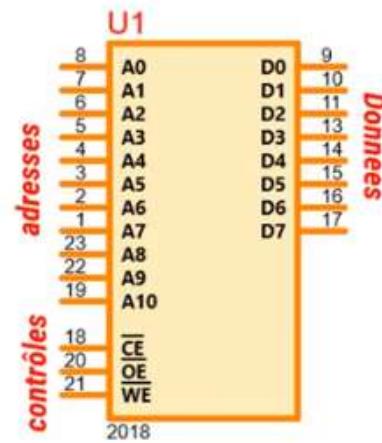
- Capacité



Chaque élément de stockage dans une mémoire peut conserver un 1 ou un 0 et est appelé une cellule. Les mémoires sont constituées de matrices de cellules,

$$2^4 \times 8 = 128 \text{ bits}$$

Exemple

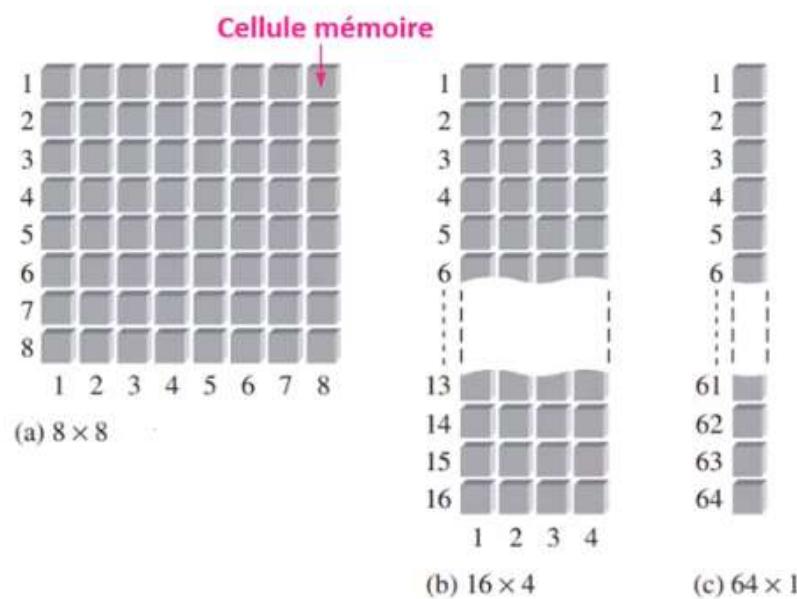


$$2^{11} \times 8 = 16384 \text{ bits}$$

$$2^{11} \times 8 = 128K\text{bits bits}$$

$$2^{11} \times 8 = 16K\text{octet bits}$$

Exemple



1 OCTET = 8 BITS

1 BIT peut valoir 0 ou 1

La norme internationale
(dans la vie courante, pour avoir un ordre d'idée)

1 Ko = 1 000 o
1 Mo = 1 000 Ko
1 Go = 1 000 Mo
1 To = 1 000 Go

la "norme" informatique
(Pour réaliser des calculs)

1 Ko = 1 024 o
1 Mo = 1 024 Ko
1 Go = 1 024 Mo
1 To = 1 024 Go

Opérations de base sur la mémoire

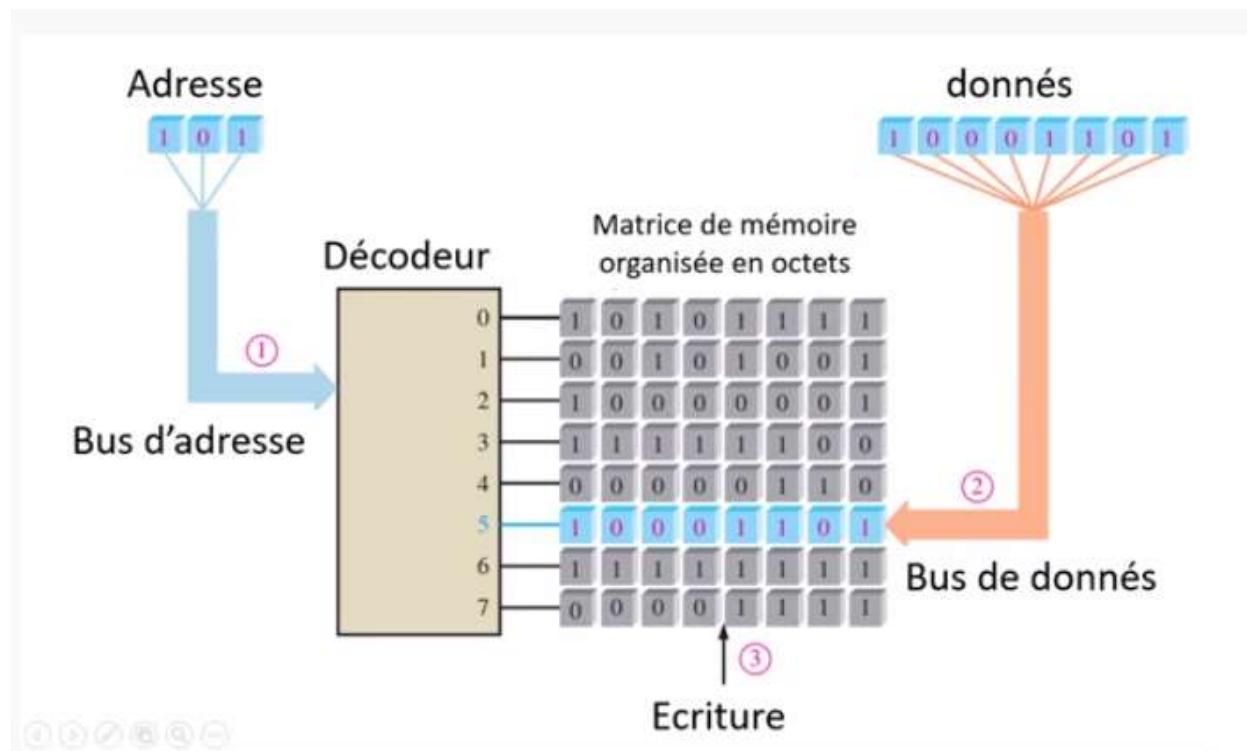
Adressage

L'adressage est le processus d'accès à un emplacement spécifié dans la mémoire. Puisqu'une mémoire stocke des données binaires, les données doivent être placées dans la mémoire et les données doivent être copiées à partir de la mémoire en cas de besoin.

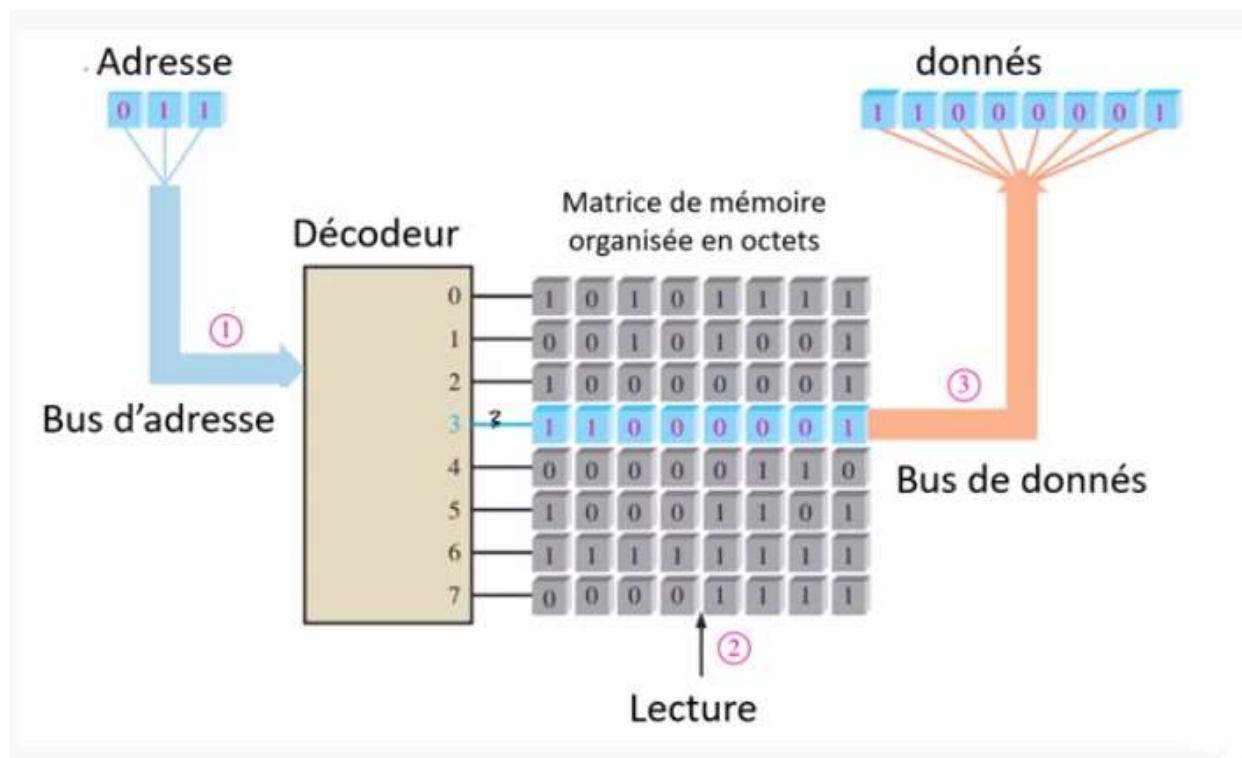
Écriture et lecture

L'opération d'écriture place les données à une adresse spécifiée dans la mémoire et l'opération de lecture copie les données à partir d'une adresse spécifiée dans la mémoire.

Ecriture dans la mémoire



Lecture dans la mémoire



Temps de cycle

Une opération de lecture ou d'écriture s'effectue avec les étapes suivantes :

- la sélection de l'adresse ;
- le choix entre lecture et écriture (niveau appliqué sur R/W) ;
- la sélection du circuit (niveau appliqué sur CS) ;
- la lecture ou l'écriture de la donnée.

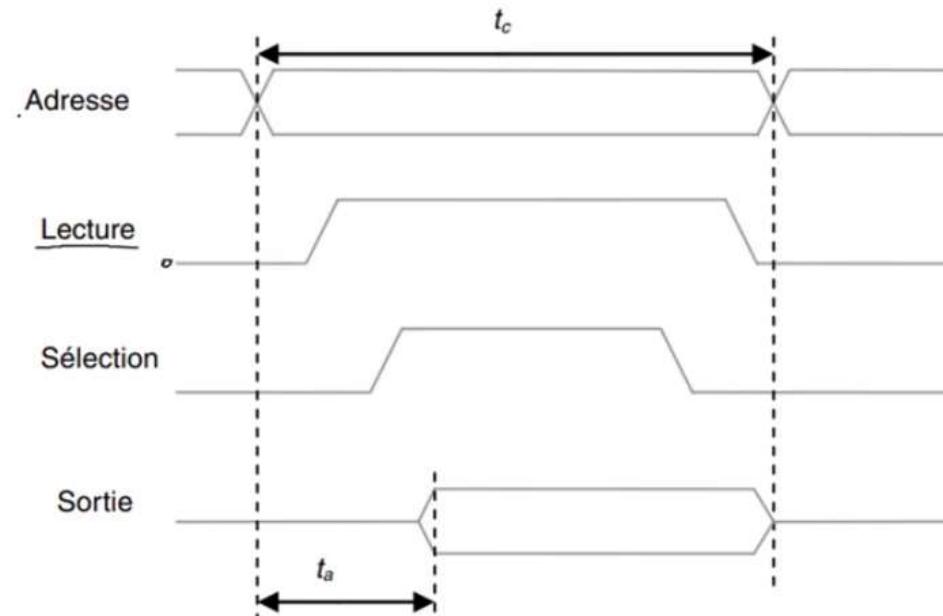
L'ensemble de ces opérations forment un cycle de lecture ou d'écriture .

Les chronogrammes font apparaître le temps d'accès t_a et le temps de cycle t_c .

Chronogrammes de lecture/écriture en mémoire

Une caractéristique importante des mémoires est leur **temps d'accès** : c'est le temps qui s'écoule entre l'instant où l'adresse de la case mémoire est présentée sur le bus d'adresses et celui où la mémoire place la donnée demandée sur le bus de données. Ce temps varie entre 50 ns (mémoires rapides) et 300 ns (mémoires lentes).

Chronogramme de lecture en mémoire :

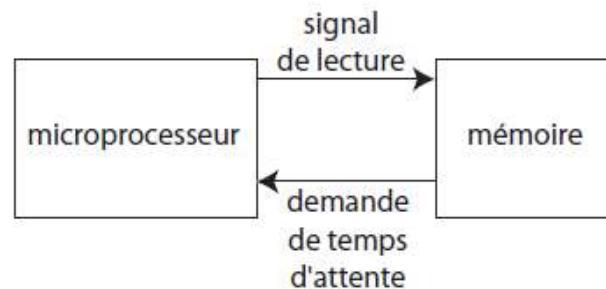


Connexion de plusieurs boitiers mémoire sur le bus d'un microprocesseur

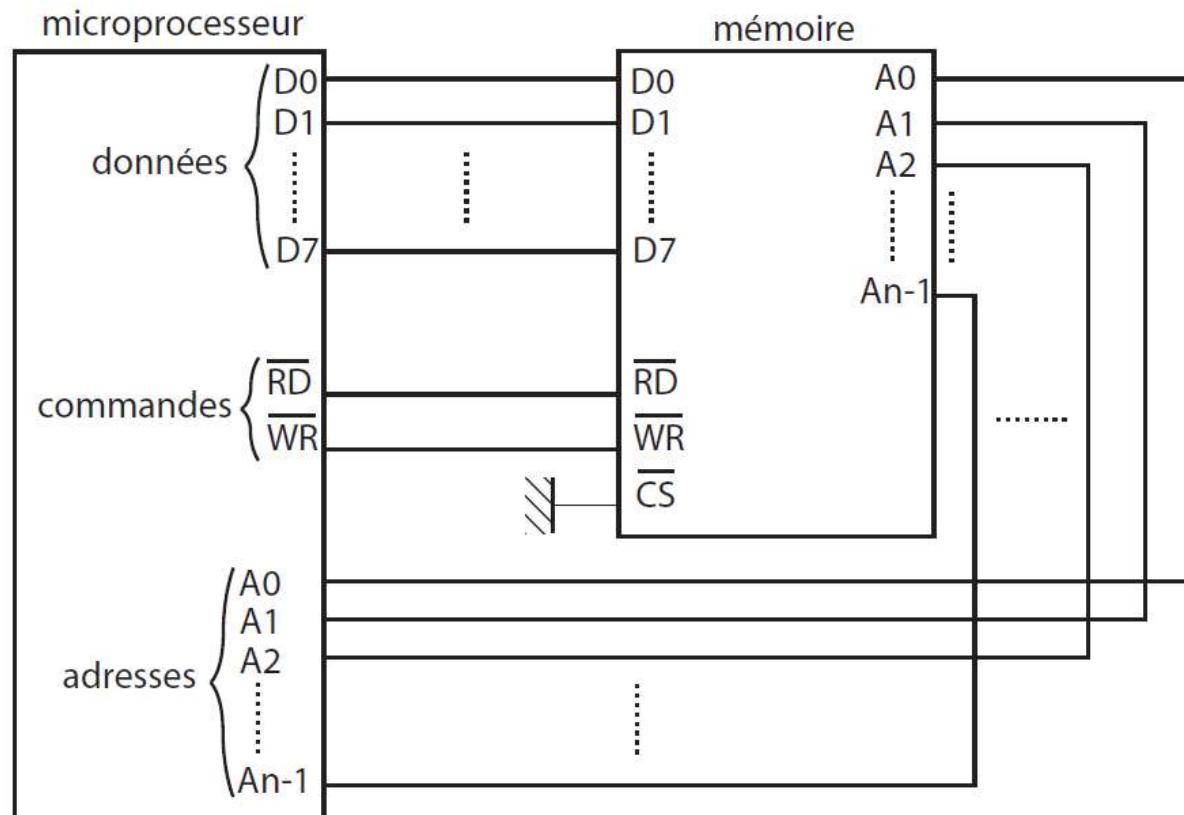
Les boitiers mémoire possèdent une broche notée CS : Chip Select. Lorsque cette broche est active (état bas), le circuit peut être lu ou écrit. Lorsqu'elle est inactive(état haut), le circuit est exclu du service : ses broches de données D0 `a D7 passent `a l'état de haute impédance : tout se passe comme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

Chronogrammes de lecture/écriture en mémoire(suite)

Remarque : si le temps d'accès d'une mémoire est supérieur à une période d'horloge (mémoire lente), le microprocesseur peut accorder à la mémoire un temps supplémentaire (une ou plusieurs périodes d'horloge), à la demande de celle-ci. Ce temps supplémentaire est appelé **temps d'attente** (wait time : TW) :

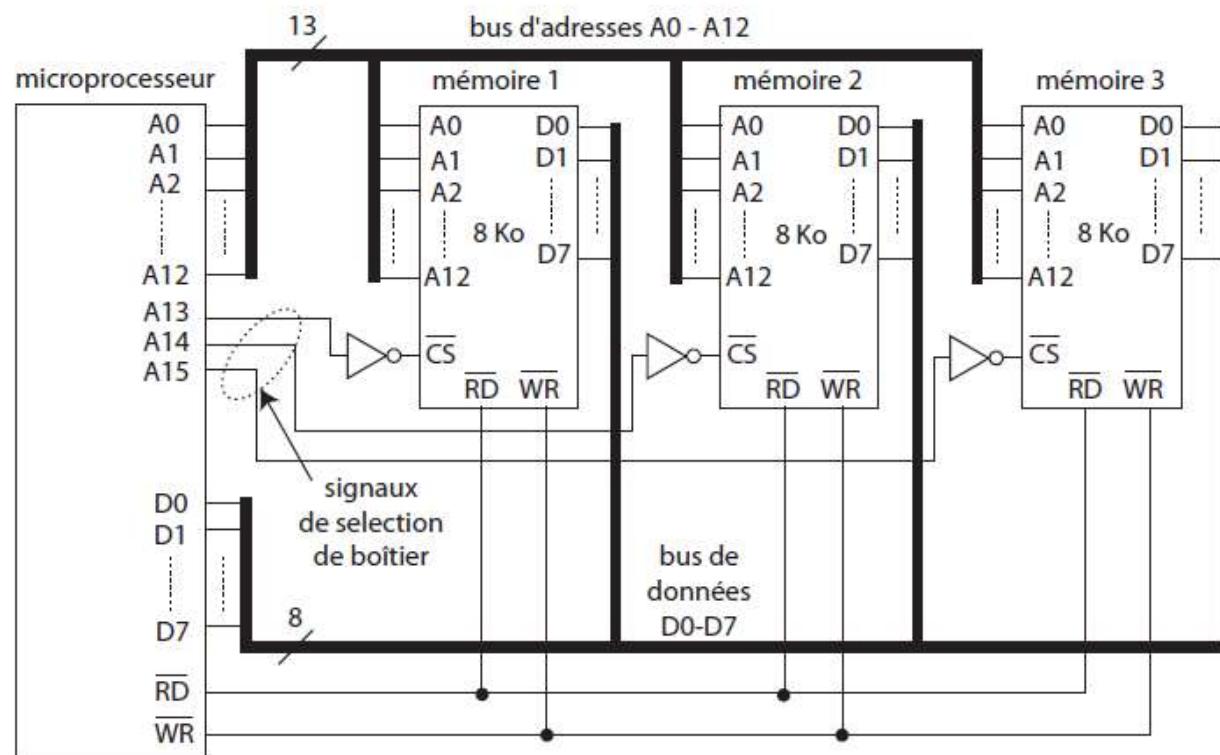


Interfaçage microprocesseur/mémoire



Connexion de plusieurs boîtiers mémoire sur le bus d'un microprocesseur

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



Connexion de plusieurs boîtiers mémoire sur le bus d'un microprocesseur(suite)

Dans un même boîtier, une case mémoire est désignée par les bits d'adresses A0 à A12 :

A12	A11	...	A1	A0	à	A12	A11	...	A1	A0
0	0	...	0	0		1	1	...	1	1
0000H										1FFFFH

Pour atteindre la mémoire n°1, il faut mettre à 1 le bit A13 et à 0 les bits A14 et A15.
La plage d'adresses occupée par cette mémoire est donc :

A15	A14	A13	A12	...	A0	à	A15	A14	A13	A12	...	A0	
0	0	1	0	...	0		0	0	1	1	...	1	
2000H													3FFFFH

De même, pour la mémoire n°2, on doit avoir A13 = 0, A14 = 1 et A15 = 0 d'où la plage d'adresses occupée cette mémoire :

A15	A14	A13	A12	...	A0	à	A15	A14	A13	A12	...	A0	
0	1	0	0	...	0		0	1	0	1	...	1	
4000H													5FFFFH

Pour la mémoire n°3, on doit avoir A13 = 0, A14 = 0 et A15 = 1 d'où la plage d'adresses occupée cette mémoire :

A15	A14	A13	A12	...	A0	à	A15	A14	A13	A12	...	A0	
1	0	0	0	...	0		1	0	0	1	...	1	
8000H													9FFFFH

Décodage d'adresses

Les trois bits A13, A14 et A15 utilisés précédemment fournissent en fait 8 combinaisons, de 000 à 111, d'où la possibilité de connecter jusqu'à 8 boîtiers mémoire de 8 Ko sur le bus. La mémoire totale implantée devient donc de $8 \times 8 \text{ Ko} = 64 \text{ Ko}$: valeur maximale possible avec 16 bits d'adresses.

Pour cela, il faut utiliser un **circuit de décodage d'adresses**, dans ce cas : un **décodeur 3 vers 8**.

