Chapitre II :

Principaux composants d'un ordinateur

1. Introduction
2. Composants d'un ordinateur
3. Processeur (Unité Arithmétique et Logique)
4. Bus
5. Registres
6. Mémoire
7. Mémoire interne
8. Mémoire cache
9. Hiérarchie de mémoires
10. Conclusion

2.1. Introduction

Ce deuxième chapitre s'intéresse aux éléments fondamentaux d'un ordinateur et leurs fonctionnements. Nous présenterons les principaux modules constituant l'architecture d'un ordinateur type. Nous expliquerons la fonctionnalité de chacun de ces modules et de leurs relations fonctionnelles dans l'ordinateur. Il s'agit ici uniquement de présenter de manière globale le fonctionnement de l'ordinateur.

En premier lieu, nous parlerons de la carte mère et ses caractéristiques, de l'unité de calcul (Unité Arithmétique et Logique qui se trouve dans le processeur) ainsi que des composants de transport d'informations (les registres pour le processeur et les bus pour le reste des éléments). Ensuite, nous parlerons d'un composant indispensable de l'ordinateur 'la mémoire', qui se résume souvent à une simple fonction de stockage. On distingue plusieurs catégories de mémoires différenciées par leurs caractéristiques (adressage, performances, accès…) : Mémoire interne et Mémoire cache.

Un **ordinateur** est une machine électronique capable de résoudre et traiter des problèmes en appliquant des instructions probablement définies. Donc il permet :

- D'**acquérir** des informations.
- De **conserver** des informations.
- D'**effectuer** des traitements sur les informations.
- De **restituer** des informations.

Concernant l'**organisation de base d'un ordinateur** (fig. 3), il doit posséder les unités fonctionnelles suivantes :

- **Unité de traitement** (*Processeur*) : cerveau de l'ordinateur, supervise les autres unités et effectue les traitements (exécution et calcule).
- **Unités de stockage** (*Mémoire*) : lieu de stockage des informations (programmes et données).
- **Unités d'entrées et de sorties** (*Périphériques*) : ce sont les unités qui sont destinées à recueillir les informations en entrée et à les restituer en sortie.
- **Bus de communication** assurent les connections entre les différentes unités.

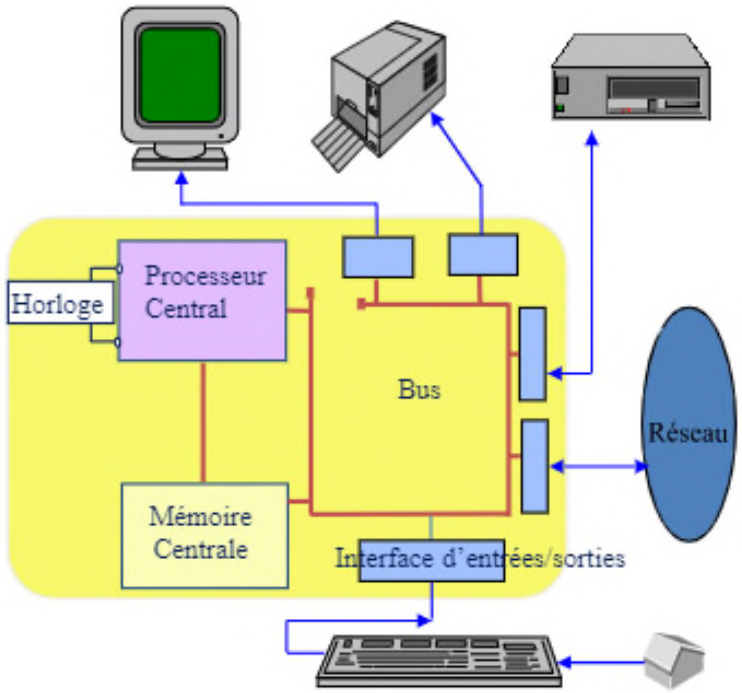


Figure 3 : La structure d'un ordinateur général.

2.2. Composants de l'ordinateur

Un ordinateur est un ensemble de **composants électroniques modulaires**, c'est-à-dire des composants pouvant être remplacés par d'autres composants ayant éventuellement des caractéristiques différentes, capables de faire fonctionner des programmes informatiques.

La mise en œuvre de ces systèmes s'appuie sur deux modes de réalisation distincts :

- ***Le matériel*** (**hardware** mot signifiant quincaillerie) correspond à l'aspect concret ou physique de l'ordinateur : unité centrale, mémoire, organes d'entrées-sorties, etc…
- ***Le logiciel*** (**software** mot fabriqué pour les besoins de la cause en remplaçant hard 'dur' par soft 'mou') désigne au contraire tout ce qui n'est pas matériel est qui correspond à un ensemble d'instructions, appelé programme, qui sont contenues dans les différentes mémoires du système d'un ordinateur et qui définissent les actions effectuées par le matériel.

Les composants matériels de l'ordinateur sont architecturés autour d'une carte principale comportant quelques circuits intégrés et beaucoup de composants électroniques tels que *condensateurs, résistances, etc...* Tous ces composants sont soudés sur la carte et sont reliés par les connexions du circuit imprimé et par un grand nombre de connecteurs : cette carte est appelée « **carte mère ».**

2.2.1 Présentation de la carte mère

L'élément constitutif principal et essentiels de l'ordinateur est la ***carte* mère** (en anglais « **Mainboard** » ou « **Motherboard** », parfois abrégé en « **Mobo** »).

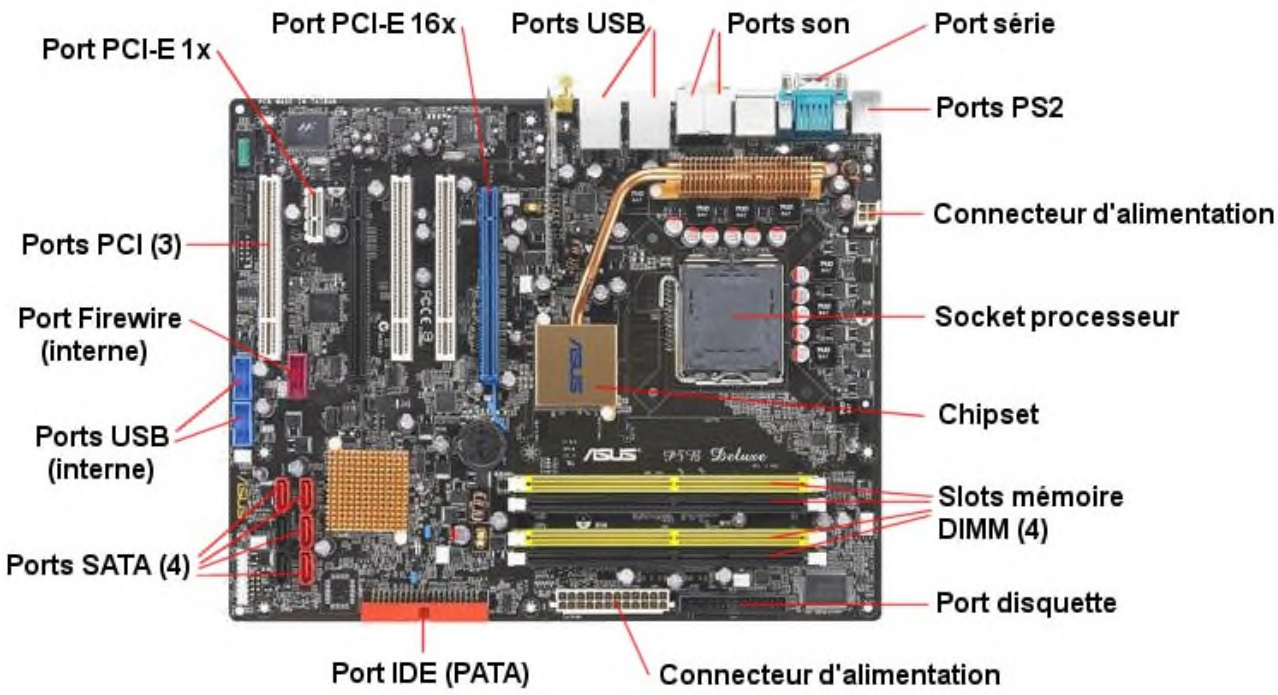


Figure 4 : Le modèle de carte mère.

La **carte mère** (fig. 4 https://lacartemere.wordpress.com/) est la plus grande carte électronique prenant la forme d'un circuit imprimé. C'est le système nerveux de l'ordinateur car elle assemble et met en relation tous les composants matériels. Elle permet à tous ses composants de fonctionner ensemble efficacement car elle assure la connexion physique des différents composants (processeur, mémoire, carte d'entrées/sorties, ...) par l'intermédiaire de différents bus (adresses, données et commande). La qualité de la carte mère est vitale puisque la performance de l'ordinateur dépend énormément d'elle.

2.2.2 Caractéristiques d'une carte mère

Il existe plusieurs façons de caractériser une carte mère, notamment selon les caractéristiques suivantes :

a. **Le facteur d'encombrement *(ou facteur de forme, en anglais form factor)* :** on désigne par ce terme la géométrie, les dimensions, l'agencement et les caractéristiques électriques de la carte mère. Il existe différents formats de cartes mères, comme par exemple : en 1995 ***ATX (****Advanced Technology eXtended****)***, en 2005 ***BTX (****Balanced Technology eXtended,****)***, en 2009 ***ITX (****Information Technology eXtended****)***, … *etc.*

b. **Le chipset : *(traduisez jeu de composants ou jeu de circuits)* :** c'est une interface d'entrée/sortie. Elle est constituée par un jeu de plusieurs composants chargés de gérer la communication entre le microprocesseur et les périphériques. C'est le lien entre les différents bus de la carte mère.

c. **Le bios *(Basic Input Output Service)* :** c'est un programme responsable de la gestion du matériel (clavier, écran, disque dur, liaisons séries et parallèles, etc..). Il est sauvegardé dans une mémoire morte (ROM de type EEPROM) et agit comme une interface entre le système d'exploitation et le matériel.

d. **Le type de support :** On distingue deux catégories de supports :

1. ***Sockets :*** un **socket** (en anglais) est le nom du connecteur destiné au processeur. Il s'agit d'un connecteur de forme carré possédant un grand nombre de petits connecteurs sur lequel le processeur vient directement s'enficher.
2. ***Slots :*** un **slot** (en anglais) est une fente rectangulaire dans laquelle on insère un composant. Selon le type de composant accueilli, on peut utiliser d'autres mots pour designer des slots :
   - Un port d'extension ou un connecteur d'extension pour enficher une ***carte d'extension***
   - Un support pour enficher une barrette de ***mémoire vive***
   - Un slot pour enficher un ***processeur***, à ne pas confondre avec un socket car certains processeurs conditionnés sous forme de cartouche.

e. **Les ports de connexion :** ils permettent de connecter des périphériques sur les différents bus de la carte mère. Il existe deux sortes de connecteurs (ou ports) :

1. ***Les connecteurs internes :*** Il existe des connecteurs internes pour connecter des cartes d'extension (***PCI*** 'Peripheral Component Interconnect', ***ISA*** 'Industry Standard Architecture', ***AGP*** 'Accelerated Graphics Port') ou des périphériques de stockage de masse (***IDE*** aussi appelé PATA 'Parallel ATA', ***SCSI*** 'Small Computer System Interface', ***SATA*** 'Serial ATA').
2. ***Les connecteurs externes*** (aussi appelé I/O Panel (Input/Output Panel) en anglais) : Il existe des connecteurs externes pour connecter d'autres périphériques externes à l'ordinateur : ***USB*** 'Universal Serial Bus', ***RJ45*** **'**Registered Jack**'**, ***VGA*** **'**Video Graphics Array**'**, ***DVI*** **'**Digital Visual Interface**'**, ***HDMI*** **'**High Definition Multimedia Interface**'**, ***DisplayPort***, ***audio analogiques***, ***audio numériques***, ***Firewire.***

Remarque :

- Il existe à autres éléments embarqués dans la carte mère (intégrés sur son circuit imprimé) comme l'*horloge* et la *pile du CMOS*, le *bus système* et les *bus d'extension*.
- Les cartes mères récentes embarquent généralement un certain nombre de périphériques multimédia et réseaux pouvant être désactivés : *carte réseau intégrée, carte graphique intégrée, carte son intégrée, contrôleurs de disques durs évolués.*
- Les bus de connexions filaires tendent à être remplacés par des systèmes de communications sans fils. A l'heure actuelle, il existe :
  - ***Bluetooth*** qui va servir à connecter des périphériques nécessitant des bandes passantes faibles (clavier, souris, etc…).
  - ***WIFI*** (WIreless FIdelity Network) qui permet de connecter des ordinateurs en réseau.

2.3. Processeur

Le **processeur** (**CPU**, pour *Central Processing Unit*, soit *Unité Centrale de Traitement*) est le cerveau de l'ordinateur. Il permet les échanges de données entre les différents composants (disque dur, mémoire RAM, Carte graphique, …) et de manipuler des informations numériques, c'est-à-dire des informations codées sous forme binaire et d'exécuter les instructions stockées en mémoire. Sa puissance est exprimée en Hertz.

Electroniquement, le processeur est une puce (circuit intégré complexe) d'environ 4cm de côté et quelques millimètres d'épaisseur en silicium regroupant quelques centaines de millions de transistors, qui chauffe beaucoup car il est très sollicité. Au-dessus du radiateur, un ventilateur va se charger d'évacuer cette chaleur (fig. 5 lien : https://cours-informatique-gratuit.fr/cours/processeur-et-carte-mere/ et https://blogue.bestbuy.ca/ordinateurs-portables-et-tablettes/ordinateurs/tout-ce-que-vous-voulez-savoir-sur-le-processeur-de-votre-ordinateur-portable).

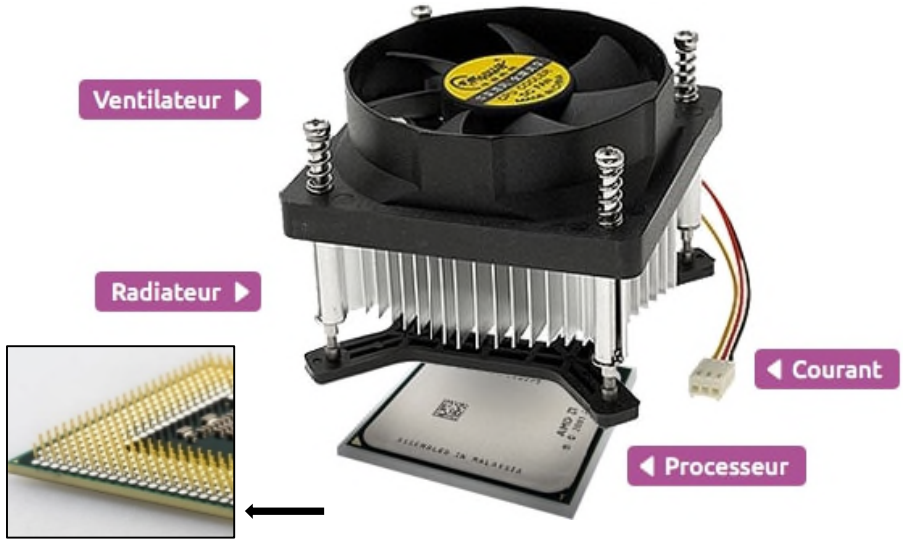


Figure 5 : Le processeur.

2.3.1 Unités d'un processeur

Le processeur est constitué d'un ensemble d'unités fonctionnelles reliées entre elles (la figure 6 ci-dessous présente son architecture générale). Les rôles des principaux éléments d'un microprocesseur sont les suivants :

1. Une **unité d'instruction** (ou ***unité de commande***, en anglais *control unit*), qui contrôle toutes les composantes et qui lit les données arrivantes, les décode puis les envoie à l'unité d'exécution. L'unité d'instruction est notamment constituée des éléments suivants :
    a. **Séquenceur** (ou ***bloc logique de commande***) chargé de synchroniser l'exécution des instructions au rythme d'une horloge. Il est ainsi chargé de l'envoi des signaux de commande.
    b. **Compteur ordinal** contenant l'adresse de l'instruction en cours.
    c. **Registre d'instruction** contenant l'instruction à exécuter.
    d. **Décodeur d'instruction** identifie l'instruction à exécuter qui se trouve dans le registre RI, puis d'indiquer au séquenceur la nature de cette instruction afin que ce dernier puisse déterminer la séquence des actions à réaliser.

2. Une **unité d'exécution** (ou ***unité de traitement***), qui accomplit les tâches que lui a donné l'unité d'instruction. L'unité d'exécution est notamment composée des éléments suivants :
3. L'**unité arithmétique et logique** (notée **UAL** ou en anglais *ALU* pour *Arithmetical and Logical Unit*) pour le traitement des données.
   a. L'**unité de virgule flottante** (notée **FPU**, pour *Floating Point Unit*), qui accomplit les calculs complexes non entiers que ne peut réaliser l'unité arithmétique et logique.
   b. Le **registre d'état**.
   c. Le **registre accumulateur**.
4. Une **unité de gestion des bus** (ou ***unité d'entrées-sorties***), qui gère les flux d'informations entrant et sortant, en interface avec la mémoire vive du système.

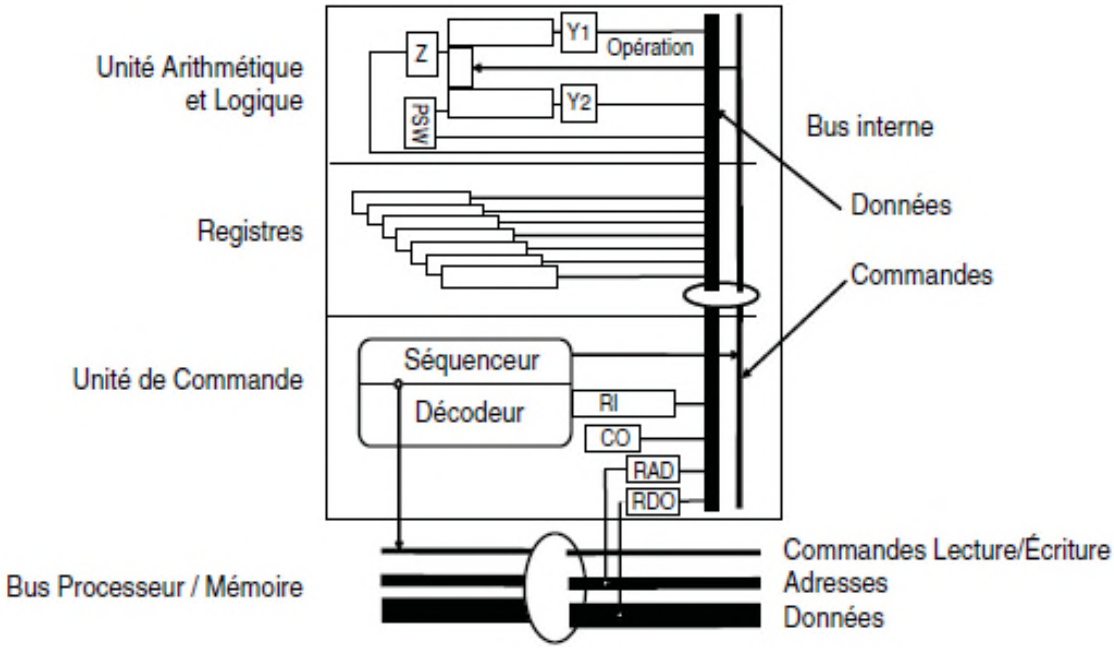


Figure 6 : L'architecture générale d'un processeur

2.3.2 Unité Arithmétique et Logique

C'est le cœur du processeur, l'**UAL** (l'abrégé de l'unité arithmétique et logique) est chargé de l'exécution de tous les calculs que peut réaliser le microprocesseur

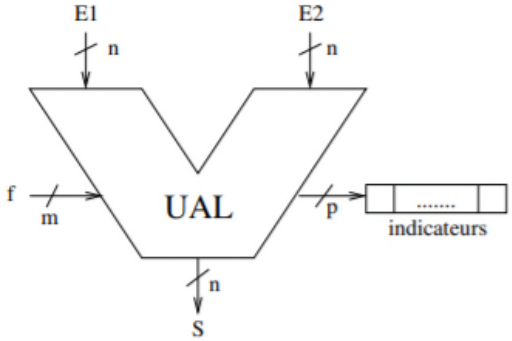


Figure 7 : L'unité arithmétique et logique.

C'est un circuit combinatoire (fig. 7) qui produit un résultat (S) sur n bits en fonction des données présentes sur ses entrées (E1 et E2) et de la fonction à réaliser (f) et met à jour les indicateurs.

➢ **Fonctions de l'UAL :**

L'UAL permet de réaliser différents types d'opérations sur des données de la forme S= f (E1, E2) :

- ➢ Des opérations arithmétiques : additions, soustractions, ...
- ➢ Des opérations logiques : ou, et, ou exclusif, ...
- ➢ Des décalages et rotations.

Elle met par ailleurs à jour **des indicateurs d'états** (ou drapeaux ou flag) en fonction du résultat de l'opération effectuée :

- **S** (Signe) : le bit de signe du résultat de la dernière opération arithmétique.
- **Z** (Zéro) : indicateur mis à 1 si le résultat de l'opération est 0.
- **N** (Négatif) : indicateur mis à 1 pour un résultat négatif (bit le plus à gauche égal à 1).
- **C** (Carry-out) : mis à 1 en cas de retenue ou débordement en contexte non signé.
- **V** (Overflow) : mis à 1 en cas de débordement en contexte signé.
- …

Exemple : Dans notre exemple, l'UAL possède deux registres d'entrée (E1 et E2) et un registre de sortie (S). Pour faire une addition :

– la première donnée est placée dans E1 via le bus interne de données.

– la seconde donnée est placée dans E2 via le bus interne de données.

– la commande d'addition est délivrée au circuit d'addition via le bus interne de commandes.

– le résultat est placé dans le registre S.

2.4. BUS

Un bus est un ensemble de fils (conducteurs électriques) qui assure la transmission des informations binaires entre les éléments de l'ordinateur. Il y a plusieurs bus spécialisés en fonction des types de périphériques concernés et de la nature des informations transportées : adresses, commandes ou données.

2.4.1. Caractéristiques d'un bus

Un bus est caractérisé par :

a. **Sa largeur :** un bus est caractérisé par le volume d'informations qui peuvent être envoyées en parallèle (exprimé en bits) correspond au nombre de lignes physiques sur lesquelles les données sont envoyées de manière simultanée. Ainsi la **largeur** désigne le nombre de bits qu'un bus peut transmettre simultanément.

1 fil transmet un bit, 1bus à n fils = bus n bits

Exemple : une nappe de 32 fils permet ainsi de transmettre 32 bits en parallèle.

b. **Sa vitesse :** est le nombre de paquets de données envoyés ou reçus par seconde. Elle est également définie par sa **fréquence** (exprimée en Hertz).
On parle de **cycle** pour désigner chaque envoi ou réception de données. Un cycle mémoire assure le transfert d'un mot mémoire :

Cycle mémoire (s)=1 / fréquence

c. **Son débit :** Le **débit** maximal du bus (ou le **taux de transfert** maximal) est la quantité de donnée qu'il peut transfert par unité de temps, en multipliant sa largeur par sa fréquence.

Débit (octets/s) = (nombre de transferts par seconde * largeur) / 8

Bande passante (en Mo/s) = largeur bus (en octets) * fréquence (en Hz)

Exercice : Un bus de 8 bits, cadencé à une fréquence de 100 MHz. Calculer le taux de transfert.

Solution :

Le bus possède donc un taux de transfert égal à :

Taux de transfert = largeur bus * fréquence

$= 8 * 100.10^6 = 8.\ 10^8$ bits/s $= 10^8$ octets/s $= 10^5$ K octets/s = **10^2 M octets/s**

2.4.2. Différents types de bus

Selon la nature de l'information à transporter, on retrouve trois types de bus d'information (fig. 8) en parallèle dans un système de traitement programmé de l'information:

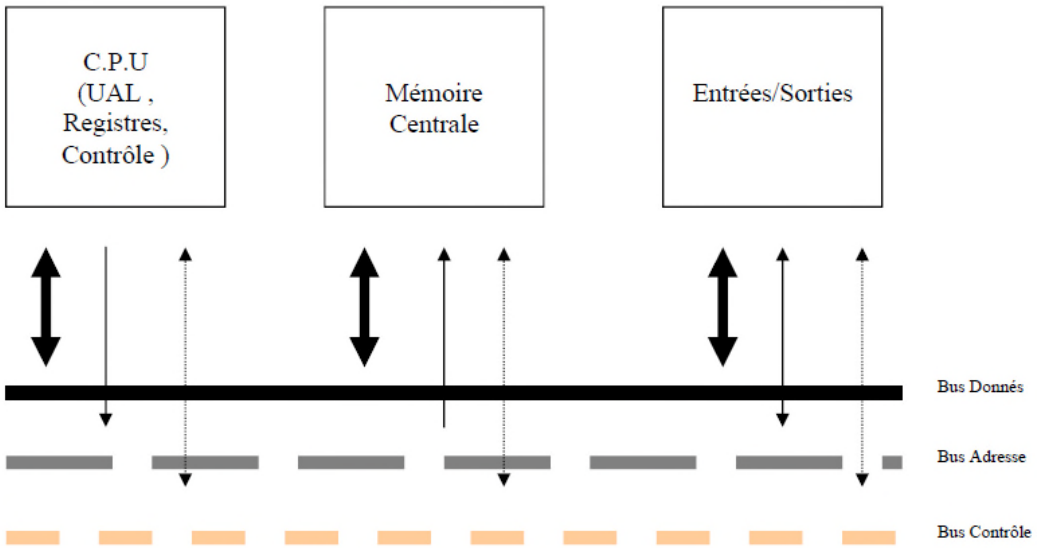


Figure 8 : Les différents types de bus.

a. **Bus de données** : c'est un bidirectionnel, il assure le transfert des informations (opérations et données) entre le microprocesseur et son environnement, et inversement. Son nombre de lignes est égal à la capacité de traitement du microprocesseur.

b. **Bus d'adresses (bus d'adressage ou mémoire) :** c'est un bus unidirectionnel, il permet la sélection des informations à traiter dans un espace mémoire (ou espace adressable) selon la demande du processeur pour lire ou écrire une donnée. Il peut avoir **2^n** emplacements, avec **n** = nombre de conducteurs du bus d'adresses.

c. **Bus de commande (bus de contrôle)** : c'est un bidirectionnel, constitué par quelques conducteurs qui assurent la synchronisation des flux d'informations. Il transporte les signaux de contrôle (lecture ou écriture mémoire, opération d'entrées/ sorties, …), dont les éléments sont disponibles sur les bus donnés ou adresses.

2.4.3. Types de bus de données

Il existe deux grands types de bus de données (fig. 9 lien : https://www.commentcamarche.net/contents/770-port-serie-et-port-parallele) selon le type de transmission :

a. **Les bus séries :** ils permettent des transmissions sur de grandes distances. Ils utilisent une seule voie de communication sur laquelle les bits sont envoyés les uns à la suite des autres. Exemples : USB, SATA.
b. **Les bus parallèles :** sur un bus parallèle plusieurs bits sont transmis simultanément. Ils sont utilisés sur des distances courtes par exemple ; pour relier le processeur à la mémoire. Exemple : PATA.

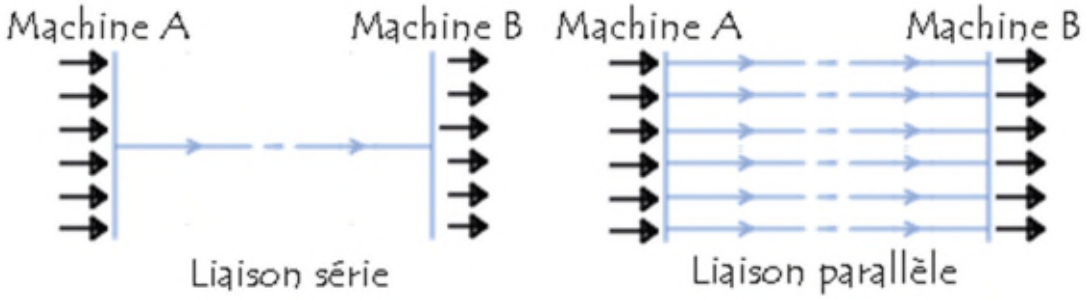


Figure 9 : Les types de bus de données.

2.4.4. Principaux bus

On distingue généralement sur un ordinateur deux principaux bus :

a. **Bus système** (*bus interne, en anglais internal bus ou front-side bus, noté FSB*) : permet au processeur de communiquer avec la mémoire centrale du système (mémoire vive ou RAM) comme les bus d'adresse et de données.
b. **Bus d'extension** (*bus d'entrée/sortie*) : permet aux divers composants liés à la carte-mère (USB, série, parallèle, cartes branchées sur les connecteurs PCI, disques durs, lecteurs et graveurs de CD-ROM, etc.) de communiquer entre eux. Il permet aussi l'ajout de nouveaux périphériques grâce aux connecteurs d'extension (appelés slots) qui lui y sont raccordés.

Remarque :

La performance d'un bus est conditionnée par sa capacité de transport simultané (16, 32, 64 bits …) et par l'électronique qui le pilote (**le chipset**).

2.5. Registres

Lorsque le processeur exécute des instructions en cours de traitement, les données sont temporairement stockées dans de petites mémoires (rapides de 8, 16, 32 ou 64 bits) que l'on appelle *registres*. Suivant le type de processeur le nombre global de registres peut varier d'une dizaine à plusieurs centaines.

Il existe deux types de registres :

a. **Registres visibles par l'utilisateur (manipulable par le programmeur) :** un registre utilisateur est un registre référençable pour les instructions exécutées par le processeur. On trouve différentes catégories :
   - **Données :** ne peuvent pas être employées pour le calcul d'adresses.
   - **Adresses :** souvent dévolues à un mode d'adressage particulier (contenant des valeurs de base ou d'index).
   - **Conditions (flags)** : constitués d'une suite de bits indépendants dont chacun est positionné en fonction du résultat d'une opération.
   - **Autres :** n'ont pas de fonction spécifique.

Les registres de l'UAL (unité arithmétique et logique), qui sont accessibles au programmeur, contrairement aux registres de l'UCC (unité de contrôle et commande). On dénombre :

- **Registre accumulateur (ACC),** stockant les résultats des opérations arithmétiques et logiques des données en cours de traitement.
- **Registres arithmétiques** : destinés pour les opérations arithmétiques (+, -, *, /, complément à 1, ...) ou logiques (NOT, AND, OR, XOR), l'accumulateur (ACC) pour stocker le résultat, ….
- **Registres d'index :** pour stocker l'index d'un tableau de données et ainsi calculer des adresses dans ce tableau.
- **Registre pointeur :** d'une pile ou de son sommet.
- **Registres généraux :** pour diverses opérations, exemple stocker des résultats intermédiaires.
- **Registres spécialisés :** destinés pour certaines opérations comme les registres de décalages, registres des opérations arithmétiques à virgule flottante, …etc

b. **Registres de contrôle et des statuts (non visible par le programmeur) :** utilisés par l'unité de commandes pour contrôler l'activité du processeur et par des programmes du système d'exploitation pour contrôler l'exécution des programmes. Quatre registres sont essentiels à l'exécution d'une instruction. Ils sont utilisés pour l'échange avec la mémoire principale :
   - **Le compteur ordinale** (CO ou PC, pour Program Counter) : contient l'adresse de la prochaine instruction à exécuter.
   - **Le registre d'instruction** (RI ou IR, pour Instruction Register) : contient l'instruction en cours de traitement.
   - **Le registre d'adresse mémoire** (MAR, pour Memory Adress Register) : contient une adresse mémoire et il est directement connecté au bus d'adresse.
   - **Le registre tampon mémoire** (MBR, pour Memory Buffy Register) **:** contient un mot de données à écrire en mémoire ou un mot lu récemment. Il est directement connecté au bus de données et il fait le lien avec les registres visibles par l'utilisateur.

Comme registre de statut, le **registre d'état** (PSW, pour Processor Status Word) contient des informations de statut. Il permet de stocker des indicateurs sur l'état du système (retenue, dépassement, etc.) et qui dépend du résultat donné par l'UAL.

2.6. Mémoire

Un ordinateur a deux caractéristiques essentielles qui sont **la vitesse** à laquelle il peut traiter un grand nombre d'informations et **la capacité de mémoriser ces informations**.

On appelle « **mémoire** » tout dispositif capable d'**enregistrer**, de **conserver** aussi longtemps que possible et de les **restituer** à la demande. Il existe deux types de mémoire dans un système informatique :

- **La mémoire centrale** (ou interne) permettant de mémoriser temporairement les données et les programmes lors de l'exécution des applications. Elle est très rapide, physiquement peu encombrante mais coûteuse. C'est *la mémoire de travail de l'ordinateur*.
- **La mémoire de masse** (ou auxiliaire, externe) permettant de stocker des informations à long terme, y compris lors de l'arrêt de l'ordinateur. Elle est plus lente, assez encombrante physiquement, mais meilleur marché. C'est *la mémoire de sauvegarde des informations*.

2.6.1. Caractéristiques d'une mémoire

Les principales caractéristiques d'une mémoire sont les suivantes :

1- **La capacité** (la taille) : représentant le volume global d'informations (en bits et aussi souvent en octet.) que la mémoire peut stocker.
2- **Le format des données** : correspondant au nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du **mot** mémorisable.
3- **Le temps d'accès** : correspondant à l'intervalle de temps entre la demande de lecture/écriture (en mémoire) et la disponibilité sur le bus de donnée T_a.
4- **Le temps de cycle** : représentant l'intervalle de temps minimum entre deux accès successifs de lecture ou d'écriture T_c. On a $T_a < T_c$ à cause des opérations de synchronisation, de rafraîchissement, de stabilisation des signaux, … etc. On $T_c = T_a +$ **temps de rafraîchissement mémoire.**
5- **Le débit** (vitesse de transfert ou bande passante) : définissant le volume d'informations échangées (lues ou écrites) par unité de temps (seconde), exprimé en bits par seconde : **Débit = n / T_c** et **n** est le nombre de bits transférés par cycle.
6- **La non volatilité :** caractérisant l'aptitude d'une mémoire à conserver les données lorsqu'elle n'est plus alimentée électriquement et volatile dans le cas contraire.

2.6.2. Modes d'accès

Le mode d'accès à une mémoire (fig. 10) dépendant surtout de l'utilisation que l'on veut en faire :

a. Direct ou Aléatoire :

- La recherche s'effectue via une adresse Mémoire à accès aléatoire, il s'agit du mode le plus employé.
- Le temps d'accès est identique car chaque mot mémoire est associé à une adresse unique.
- Les opérations associées à ce mode d'accès : *lecture(adr), écriture (adr, donnée).*

- Il est utilisé par : les mémoires qui composent la mémoire principale (mémoire vive) et quelques mémoires caches.

b. Associatif *(mémoire adressable par le contenu) :*

- La recherche s'effectue en parallèle sur toutes les cases mémoires via une clé et non via un index numérique. Donc un mot est retrouvé par une partie de son contenu.
- Le temps d'accès est constant.
- Les opérations associées à ce mode d'accès : *écriture (clé, donnée), lecture(clé), existe (clé), retirer(clé).*
- Il est employé principalement par les mémoires caches.

c. Semi direct ou Semi séquentiel *(Intermédiaire entre séquentiel et direct) :*

- Accès direct à un bloc de données ou cylindre (contenant la donnée recherchée) via son adresse unique puis déplacement séquentiel jusqu'à la donnée recherchée.
- Le temps d'accès est variable.
- Les opérations associées à ce mode d'accès : *lecture (bloc, déplacement), écriture (bloc, déplacement, donnée).*
- Il est employé par les disques (durs ou souple).

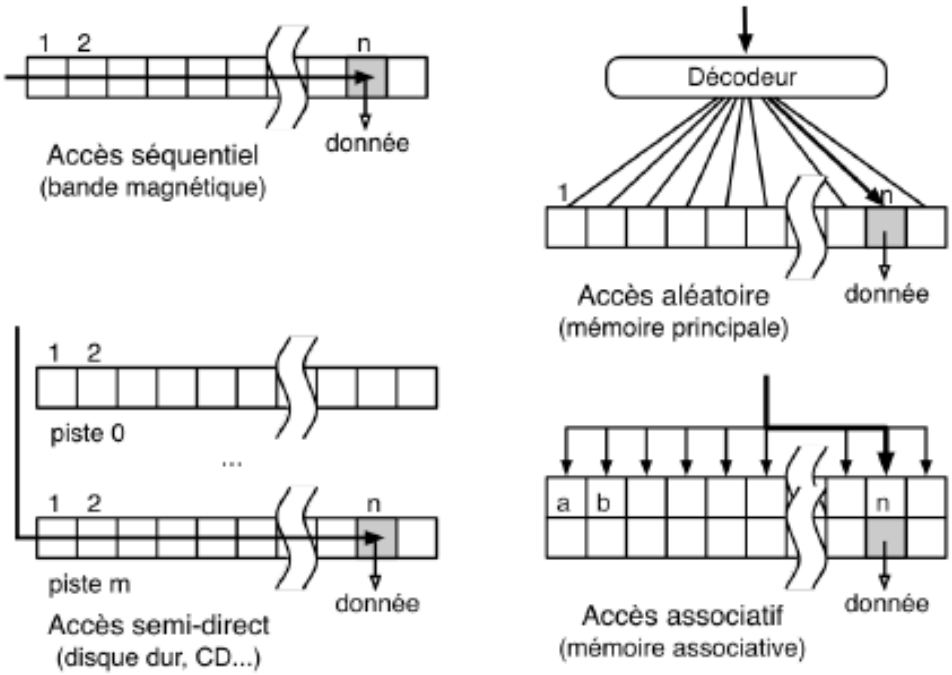


Figure 10 : Les différentes méthodes d'accès.

Remarque :

- L'accès direct est similaire à l'accès à une case d'un tableau. On accède directement à n'importe quelle case (information) directement par son indice (adresse).
- Pour un disque magnétique, l'accès à la piste est direct, puis l'accès au secteur est séquentiel. Donc c'est un accès semi-séquentiel : combinaison des accès direct et séquentiel.
- Il y a aussi un autre accès qui est **l'accès séquentiel**. C'est l'accès le plus lent il est similaire à l'accès d'une information dans une liste chainée. Pour accéder à une information, il faut parcourir toutes les informations qui la précède exemple : bandes magnétiques (K7 vidéo). Le temps d'accès est variable selon la position de l'information recherchée.

2.7. Mémoire interne

La mémoire centrale (MC) représente **l'espace de travail** de l'ordinateur car 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).

Les mémoires composant la mémoire principale sont des mémoires à base de *semi-conducteurs*, employant un mode d'accès aléatoire. Elles sont de deux types : volatiles ou non. Voici un schéma qui résume les différents types de mémoires :

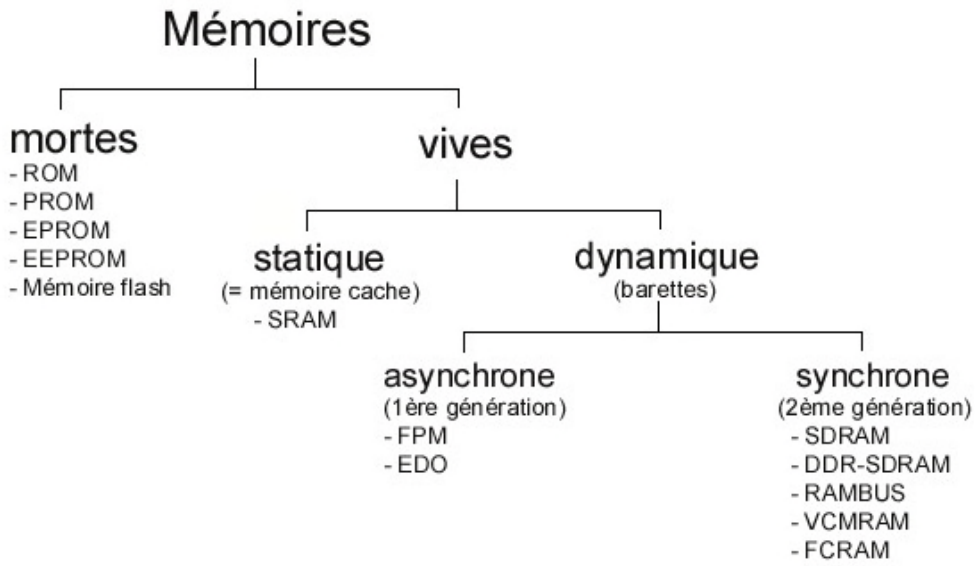


Figure 11 : Les différents types de mémoires semi-conducteurs.

2.7.1. Organisation d'une mémoire centrale

Cette mémoire est constituée de **circuits élémentaires** nommés ***bits*** (*binary digit*). Il s'agit de circuits électroniques qui présentent deux états stables codés sous la forme d'un **0** ou d'un **1**. De par sa structure la mémoire centrale permet donc de **coder les informations** sur la base d'un alphabet **binaire** et toute information stockée en mémoire centrale est représentée sous la forme d'une suite de digits binaires.

Pour stocker l'information la mémoire est **découpée** en cellules mémoires : ***les mots mémoires***. Donc une mémoire peut être représentée comme une *armoire* de rangement constituée de différents *tiroirs* où chaque tiroir représente alors une *case mémoire* (mots mémoires) qui peut contenir un seul élément (exemple fig. 12).

Chaque mot est constitué par un certain nombre de bits qui définissent sa taille. On peut ainsi trouver des mots de 1 bit, 4 bits (*quartet*) ou encore 8 bits (*octet* ou *byte*), 16 bits voire 32 ou 64 bits. Chaque mot est repéré dans la mémoire par une ***adresse***, un numéro qui identifie le mot mémoire. Ainsi un mot est un contenant accessible par son adresse et la suite de digits binaires composant le mot représente le contenu ou valeur de l'information (Données ou instruction).

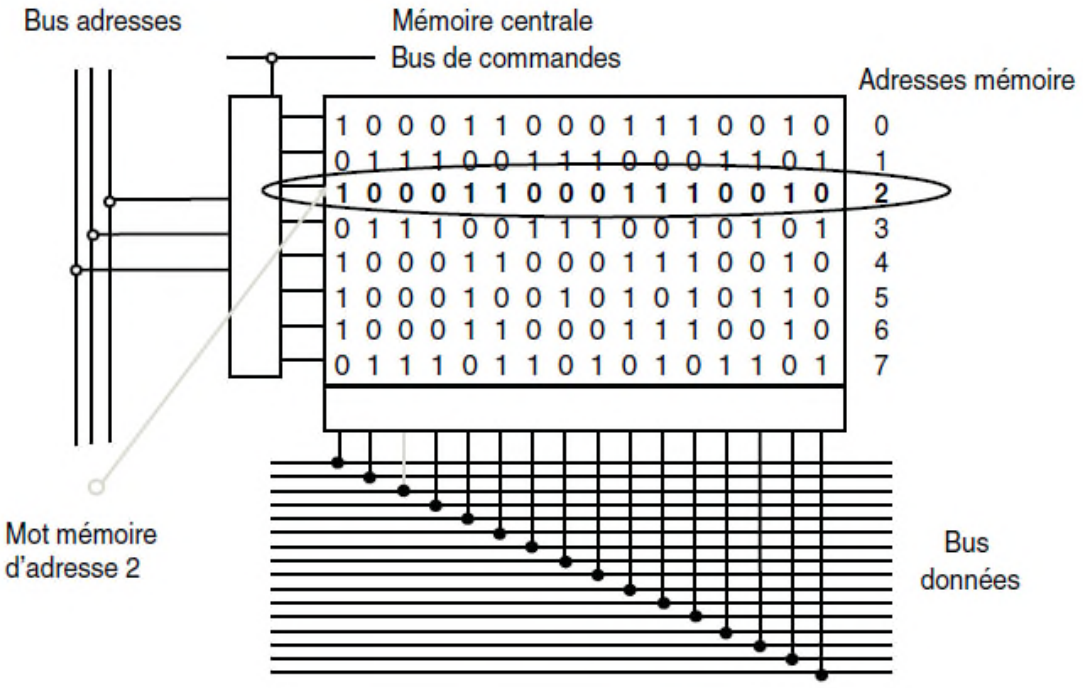


Figure 12 : L'organisation d'une mémoire centrale.

Sachant que :

- La capacité de stockage de la mémoire est définie comme étant le nombre de mots constituant.
- Avec une adresse de ***n*** bits il est possible de référencer au plus $\mathbf{2^n}$ cases mémoire
- Chaque case est remplie par un mot de données (sa longueur ***m*** est toujours une puissance de **2**).

Capacité = 2^n Mots mémoire = 2^n * m Bits

- Le nombre de fils d'adresses d'un boîtier mémoire définit donc le nombre de cases mémoire que comprend le boîtier.

Nombre de mots = $2^{\text{nombre de lignes d'adresses}}$

- Le nombre de fils de données définit la taille des données que l'on peut sauvegarder dans chaque case mémoire.

Taille du mot (en bits) = nombre lignes de données

<u>Exercice1</u> : (de la figure 12)

Notre mémoire a une capacité de 8 mots de 16 bits chacun. On exprime également cette capacité en nombre d'octets ou de bits.

Solution :

Capacité d'une mémoire = Nombre de mots * Taille du mot

Notre mémoire a donc une capacité de (8*2 octets) **16 octets** ou de (8*16 bits) **128 bits.**

Exercice 2 :

Dans une mémoire la taille du bus d'adresses K=16 et la taille du bus de données N=8. Calculer la capacité de cette mémoire ?

Solution :

Capacité d'une mémoire = Nombre de mots * Taille du mot

On a **Tailla de bus d'adresse = Nombre de lignes d'adresses** donc **Nombre de mots =** $2^{\text{nombre de lignes d'adresses}}$

Et aussi **Tailla de bus de données = Taille du mot**

Alors Capacité=2^{16} mots de 8 bits ➔ Capacité = $2^{16} * 2^{3} = 2^{19}$ **bits** = 2^{16} **octets** = 2^{13} **K octets**

Remarque :

- Un mot de **n** bits peu avoir 2^{n} combinaisons différentes.
- La capacité est exprimée aussi en octet (ou byte) ou en mot de 8, 16 ou 32 bits. On utilise des puissances de deux, avec les unités suivantes :

Kilo 1K = 2^{10} , Méga 1M = 2^{20}, Giga 1G = 2^{30}, Téra 1T = 2^{40}, Péta 1P = 2^{50}…

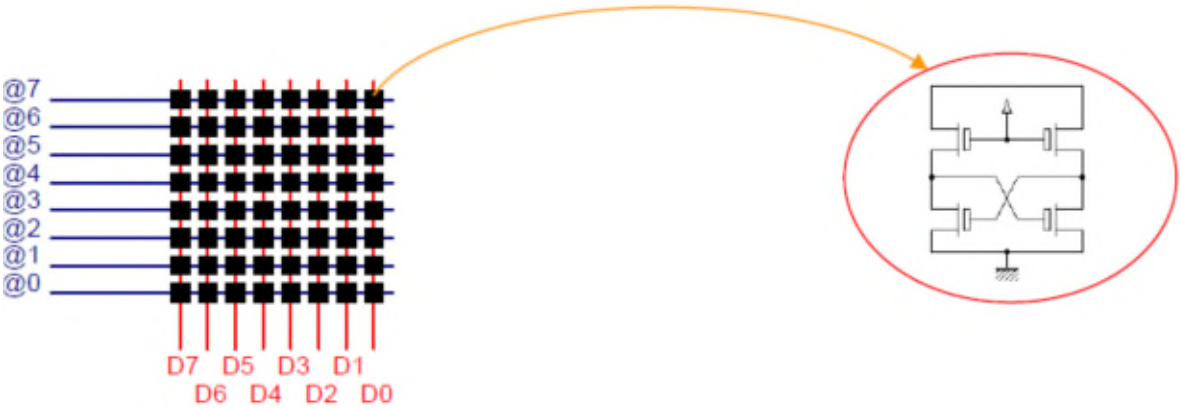
2.7.2. Mémoire vive

Une mémoire vive ou **RAM** *(Random Acces Memory*, la traduction est *Mémoire à accès aléatoire*). Son contenu est modifiable car elle sert au stockage temporaire des données et des programmes nécessaires fonctionnement du matériel. Elle doit avoir un temps de cycle très court pour ne pas ralentir le microprocesseur. Les mémoires vives sont en général volatiles car elles perdent leurs informations en cas de coupure d'alimentation.

Il existe deux grands types de mémoires RAM :

a. Les mémoires statiques

- Dans la mémoire vive statique ou SRAM (Static Random Access Memory), la cellule de base est constituée par une **bascule** (généralement bascule D) de transistors (1bit = 4 transistors = 2 portes NOR ou 6 transistors).

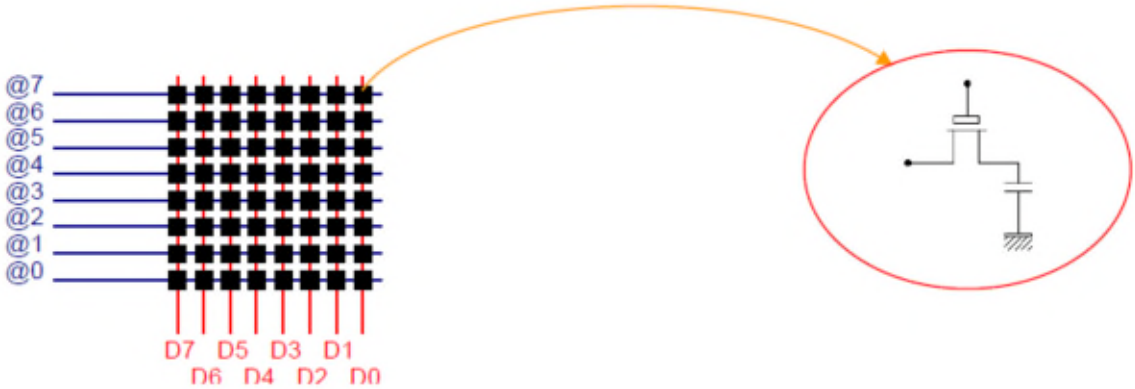


- Elle ne nécessite quasiment pas de rafraichissement.
- Le terme statique, fait référence au fonctionnement interne de la bascule.

- Dans la mesure où ce rafraichissement à un coût en temps, cela explique pourquoi ce type de mémoire est très rapide, entre 6 et 15 ns, mais assez chère.
- Elle est plus coûteuse qu'une DRAM et utilisée essentiellement pour des mémoires de faibles capacités comme dans la mémoire cache pour les microprocesseurs.
- Elle est un type de mémoire informatique spéciale utilisée dans certaines applications de recherche à très haute vitesse. Elle est aussi connue sous le nom de mémoire adressable.

b. Les mémoires dynamiques

- Dans la mémoire vive dynamique ou DRAM (Dynamic Random Access Memory), la cellule de base est constituée par un **condensateur** et un transistor (1 bit = 1 transistor + 1 condensateur) et le condensateur est utilisé pour stocker l'information.

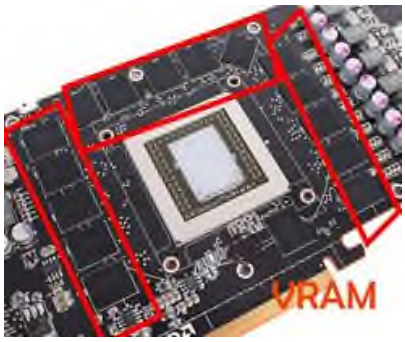


- Mémoire électronique à réalisation très simple mais le problème c'est que les condensateurs ont le défaut de se décharger (perdre lentement sa charge) et ils doivent être rechargés fréquemment (rafraichissement).
- Durant ces temps de rechargement, la mémoire ne peut être ni lue, ni écrite, ralentissant donc son fonctionnement (d'où le terme de Dynamique).
- Peu coûteuse elle est principalement utilisée pour la mémoire centrale de l'ordinateur.

Il y a aussi :

- **SDRAM** (Synchrone DRAM) : est une mémoire dynamique DRAM qui fonctionne à la vitesse du bus mémoire, elle est donc synchrone avec le bus (processeur) (lien image : https://pc4you.pro/composants-memoire-ram/724-sdram-pc100-64mb-hyundai-barrette-memoire-ram-0000000000000.html).

- **DDR SDRAM** (**D**ouble **D**ata **R**ate SDRAM) : est une SDRAM à double taux de transfert pouvant expédier et recevoir des données deux fois par cycle d'horloge au lieu d'une seule fois.



- **VRAM** (**V**ideo RAM) **:** elle a 2 ports pour pouvoir être accédée simultanément en lecture et en écriture (lien image : https://forums.tomshardware.com/threads/installed-a-kraken-g10-on-my-gtx-980-do-i-need-to-get-vrm-heatsinks.2780287/).
- **DIMM** (Dual In-line Memory Module) : groupe de puces RAM fonctionnant en 64 bits et généralement montées sur un circuit imprimé de forme rectangulaire, appelé barrette, que l'on installe sur la carte mère d'un ordinateur.
- **SIMM** (Single In-line Memory Module): idem à DIMM mais en 32 bits.
- **Mémoire flash** : est une mémoire RAM basée sur une technologie EEPROM. Le temps d'écriture est similaire à celui d'un disque dur (ex. mémoire d'appareils photos, téléphone, USB (flash disk), MemoryStick, ...). (lien image : https://www.macfix.fr/recuperation-donnees/r%C3%A9cup%C3%A9ration-de-donn%C3%A9es-cl%C3%A9-usb-ou-m%C3%A9moire-flash-detail)

Remarque :

Les performances des mémoires s'améliorent régulièrement. Le secteur d'activité est très innovant, le lecteur retiendra que les mémoires les plus rapides sont les plus chères et que pour les comparer en ce domaine, il faut utiliser un indicateur qui se nomme le cycle mémoire.

2.7.3. Mémoire morte

Les mémoires mortes ou mémoires à lecture seule (ROM : Read Only Memory) sont utilisées pour stocker des informations permanentes (programmes systèmes, microprogrammation). Ces mémoires, contrairement aux RAM, ne peuvent être que lue (l'exécution des programmes) et les conservent en permanence même hors alimentation électrique (c.à.d. non volatile). Suivant le type de ROM, la méthode de programmation changera. Il existe donc plusieurs types de ROM :

a. **ROM :** information stockée au moment de la conception du circuit.
b. **PROM :** (Programmable ROM) mémoire programmable une seule fois et elle est réalisée à partir d'un programmateur spécifique.
c. **EPROM *ou* *UV-EPROM :*** L'EPROM *(*Erasable Programmable ROM*)* mémoire (re)programmable et effaçable par ultraviolet (lien image : https://www.reichelt.com/ch/fr/eprom-uv-c-mos-c-dil-42-2-mx8-1-mx16-100-ns-27c160-100-p40037.html).

d. **EEPROM :** (Electically EPROM) mémoire (re)programmable et effaçable électriquement.

e. **FLASH EPROM :** La mémoire Flash est programmable et effaçable électriquement comme les EEPROM. Exemple : appareil photo numérique - lecteur MP3 (lien image : http://www.industrialautomation-products.com/sale-11136421-6es7952-1as00-0aa0-siemens-memory-card-ram-s7-400-flash-memory-card.html).

2.7.4. Structure physique d'une mémoire centrale

Concernant la **structure physique d'une mémoire centrale** (fig. 13), elle contient les composants suivants :

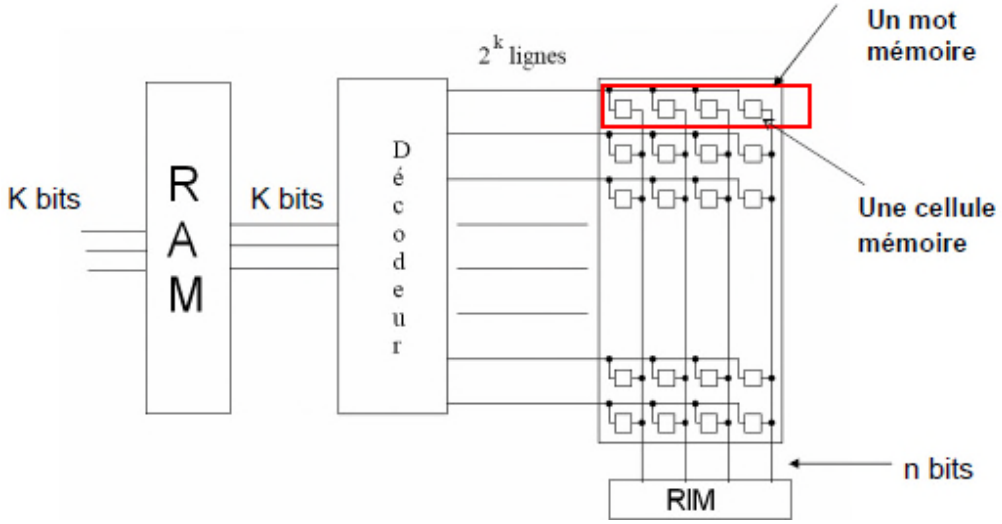


Figure 13 : La structure physique d'une mémoire centrale.

•**RAM** (Registre d'adresse Mémoire) : ce registre stock l'adresse du mot à lire ou à écrire.

•**RIM** (Registre d'information mémoire) : stock l'information lu à partir de la mémoire ou l'information à écrire dans la mémoire.

•**Décodeur** : permet de sélectionner un mot mémoire.

•**R/W** : commande de lecture/écriture, cette commande permet de lire ou d'écrire dans la mémoire (si R/W=1 alors lecture sinon écriture)

•**Bus d'adresses** de taille **k** bits

•**Bus de données** de taille **n** bits

Pour le principe de **sélection d'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, donc cette sortie va nous permettre de sélectionner un seul mot mémoire.

2.7.5. Lecture et écriture de l'information

Les seules opérations possibles sur la mémoire sont :

- **Ecriture dans un emplacement** *(récupérer ou restituer)* : le processeur donne une valeur et une adresse et la mémoire range la valeur à l'emplacement indiqué par l'adresse.
- **Lecture d'un emplacement** *(enregistrer ou modifier)* : le processeur demande à la mémoire la valeur contenue à l'emplacement dont il indique l'adresse. Le contenu de l'emplacement lu reste inchangé.

- **Algorithme de lecture**

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

1. L'unité centrale commence par charger dans le registre RAM l'adresse mémoire du mot à lire.
2. Elle lance la commande de lecture à destination de la mémoire (R/W=1)
3. L'information est disponible dans le registre RIM au bout d'un certain temps (temps d'accès) où l'unité centrale peut alors le récupérer.

- **Algorithme d'écriture**

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

1. L'unité centrale commence par placer dans le RAM l'adresse du mot où se fera l'écriture.
2. Elle place dans le RIM l'information à écrire.
3. L'unité centrale lance la commande d'écriture pour transférer le contenu du RIM dans la mémoire centrale.

Remarque :

- Il y a écriture lorsqu'on enregistre des informations en mémoire et lecture lorsqu'on récupère des informations précédemment enregistrées.
- Dans l'étape N°1, puisque les 2 opérations (lecture et écriture) sont indépendantes et qu'elles utilisent des bus différents, alors elles peuvent être effectuées en parallèle (gain de temps).

2.8. Mémoire cache

La mémoire cache ou ***antémémoire*** (fig. 14) est une mémoire très rapide d'accès pour le microprocesseur. On la réalise à partir de cellule SRAM de taille réduite (à cause du coût) car SRAM est 8 à 16 fois plus rapide que DRAM mais 4 à 8 fois plus volumineuse. Elle agit comme un tampon entre le processeur et la mémoire principale. Sa capacité mémoire est donc très inférieur à celle de la mémoire principale.

Elle est utilisée pour maintenir les parties de données et programmes qui sont le plus fréquemment utilisés par les CPU. Les parties de données et les programmes sont transférés du disque vers la mémoire cache par le système d'exploitation. Les données stockées dans une mémoire cache pourraient être les résultats d'un calcul plus tôt, ou les doublons de données stockées ailleurs.

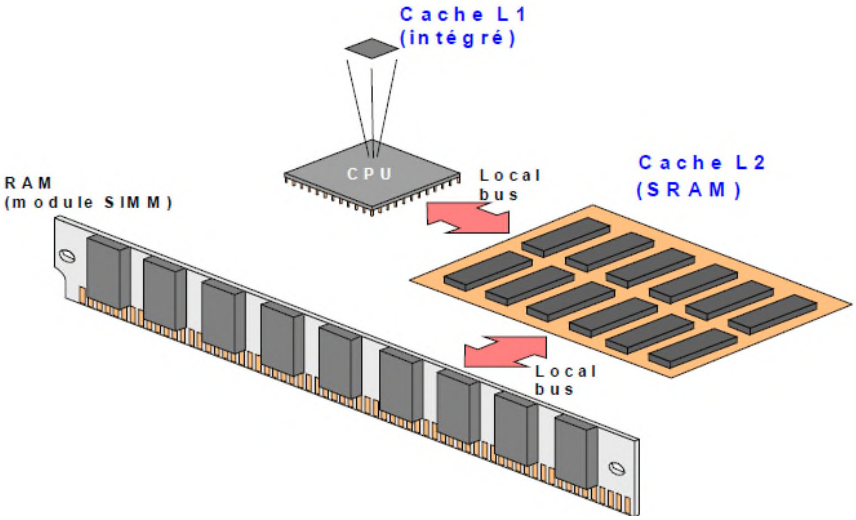


Figure 14 : Exemple de mémoire cache à deux niveaux.

Au départ cette mémoire était intégrée en dehors du microprocesseur mais elle fait maintenant partie intégrante du microprocesseur et se décline même sur plusieurs niveaux.

2.8.1. Principe

Le principe de cache est très simple (fig. 15), puisque le cache contient des copies des informations (instructions ou données) qui sont en mémoire centrale, le microprocesseur n'a pas conscience de sa présence et lui envoie toutes ses requêtes comme s'il agissait de la mémoire principale :

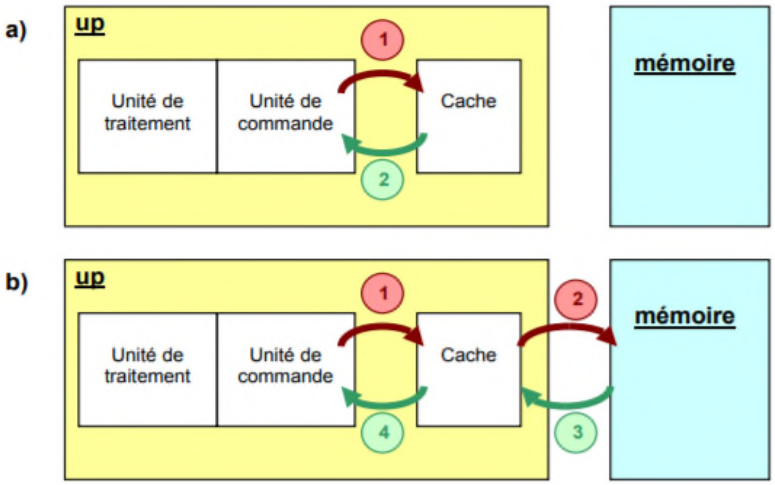


Figure 15 : Le principe de la mémoire cache.

- Soit la donnée ou l'instruction requise est présente dans le cache et elle est alors envoyée directement au microprocesseur. On parle de **succès de cache (a) (**en anglais **Hit).**
- Soit la donnée ou l'instruction n'est pas dans le cache et le contrôleur de cache envoie alors une requête à la mémoire principale. Une fois l'information récupérée, il la renvoie au microprocesseur tout en la stockant dans le cache. On parle de **défaut de cache (b) (**en anglais **Miss).**

Remarque :

Le cache mémoire n'apporte un gain de performance que dans le premier cas. Sa performance est donc entièrement liée à son taux de succès. Il est courant de rencontrer des taux de succès moyens de l'ordre de 80 à 90%.

2.8.2. Fonctionnement

Actuellement le cache des micro-processeurs récents sur le marché est composé de deux niveaux de mémoires de type SRAM la plus semblable à celle des registres : le *cache de niveau un* est noté **L1** et le *cache de niveau deux* est noté **L2**.

Sachant que la mémoire ***cache de niveau L1*** est dans le processeur (cache interne/ on-chip), *unifié :* contient instructions et données (ex. : Intel 486), mais actuellement au moins *2 caches* : 1 cache de données et 1 cache d'instructions (ex. Pentium : 2 caches L1 de 8 ko, Pentium III : 2 caches L1 de 16 ko, actuellement cache L1 : 128 ko). Avantage des caches séparés : les opérations mémoires sur des instructions et données indépendantes peuvent être simultanées.

Pour la mémoire ***cache de niveau L2***, elle est à côté du processeur (cache externe / out-chip), généralement de 256 ko. Il y a aussi le ***cache de niveau trois L3*** à l'extérieur comme **L2**, rarement utilisé (ex : Intel Core i7).

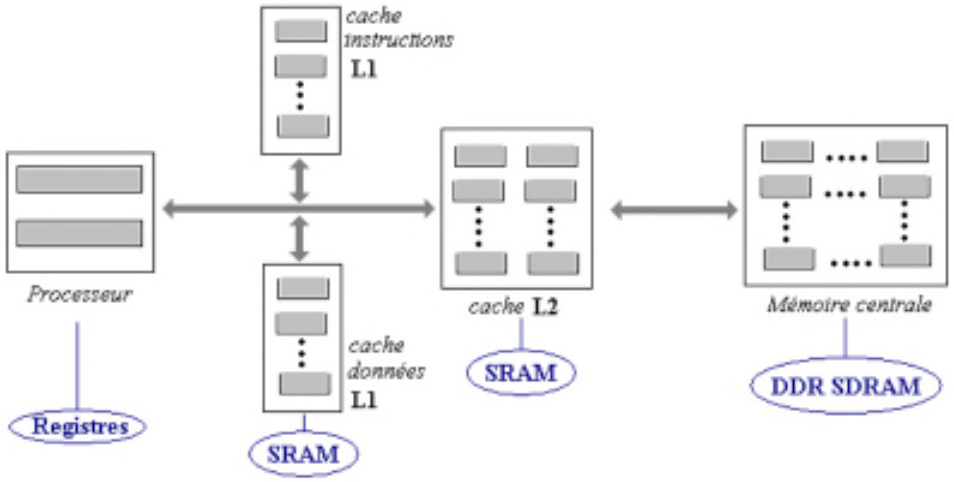


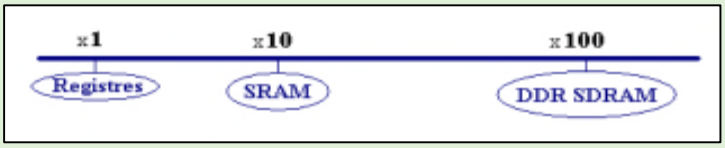
Figure 16 : Le fonctionnement de la mémoire cache.

Le fonctionnement est le suivant (fig. 16) :

- **Si** un étage du *processeur cherche une donnée,* elle va être d'abord **recherchée** dans le cache de donnée **L1** et **rapatriée** dans un *registre* adéquat, **sinon si** la donnée **n'est pas présente** dans le **cache L1**, elle sera **recherchée** dans le **cache L2**.
- **Si** la *donnée* est **présente** dans **L2**, elle est alors **rapatriée** dans un *registre* adéquat et **recopiée** dans le **bloc de donnée** du **cache L1**. Il en va de même lorsque la *donnée* **n'est pas présente** dans le **cache L2**, elle est alors **rapatriée** depuis la **mémoire centrale** dans le *registre* adéquat et **recopiée** dans le **cache L2.**

Remarque :

Le facteur d'échelle (d'un coefficient de multiplication des temps d'accès à une information) relatif entre les différents composants mémoires du processeur et de la mémoire centrale.



Les registres, mémoires les plus rapides se voient affecter la valeur de référence 1. L'accès par le processeur à une information située dans la DDR SDRAM de la mémoire centrale est 100 fois plus lente qu'un accès à une information contenue dans un registre.

2.8.3. Gestion de la mémoire cache

a. Définitions

- **Ligne :** est le plus petit élément de données qui peut être transféré entre la mémoire cache et la mémoire de niveau supérieur.
- **Mot :** est le plus petit élément de données qui peut être transféré entre le processeur et la mémoire.

b. Localité

Le principe de localité affirme que les informations auxquelles va accéder le processeur ont une forte probabilité d'être localisées dans une ***fenêtre spatiale*** et une ***fenêtre temporelle***.

- **Localité spatiale :** indique que l'accès à une instruction située à une adresse X va probablement être suivie d'un accès à une zone toute proche de X
  Exemple : tableaux, structures.
  La localité spatiale, suggère de copier des blocs de mots dans le cache plutôt que des mots isolés.
- **Localité temporelle** : indique que l'accès à une zone mémoire à un instant donné a de fortes chances de se reproduire dans la suite du programme.
  Exemple : structures itératives.
  La localité temporelle suggère de conserver pendant quelque temps dans le cache les informations auxquelles on vient d'accéder.

c. Nombre de cache et localisation

Actuellement, la norme est à l'utilisation de multiple caches, organisés en niveau (level).

- Un cache peut être situé sur la même puce que le processeur (**on-chip/internal cache**).
- Ou n'être accessible que via un bus externe au processeur (**external cache**).

L'utilisation d'un cache interne permet d'augmenter les performances de laisser le bus externe disponible.

Une organisation typique est : un cache interne (de niveau 1) et un cache externe (de niveau 2). Le cache de niveau 2 doit être de 10 à 100 fois plus grand que le/les caches de niveau 1 pour être intéressant.

d. Correspondance cache et mémoire (le mapping)

La taille du cache est beaucoup plus petite que la taille de la mémoire. Il faut définir une stratégie de copie des blocs de données dans le cache. Cette méthode s'appelle le mapping. Trois stratégies sont possibles :

- **Correspondance directe (direct mapped cache) :** le bloc **n** de la mémoire principale peut se retrouver seulement dans le bloc **m = (n modulo sb)** de la mémoire cache, sachant que **sb** est la taille en nombre de blocs de la mémoire cache (fig. 17).

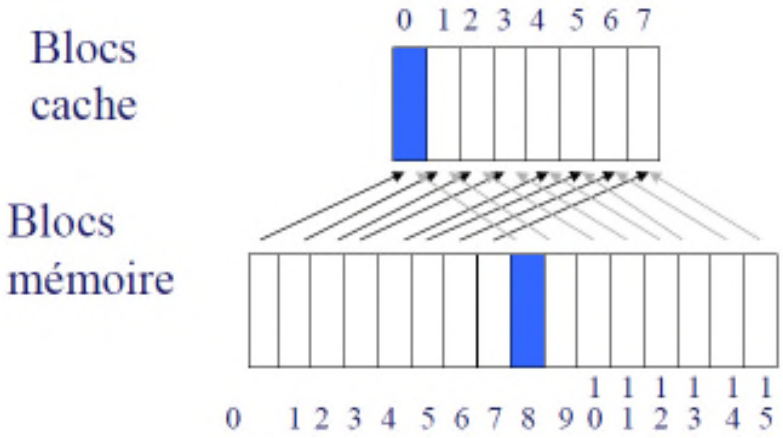


Figure 17 : La correspondance cache directe.

- **Correspondance totalement associatif (fully associative cache)** : chaque bloc mémoire peut être placé dans n'importe quel bloc du cache (fig. 18).

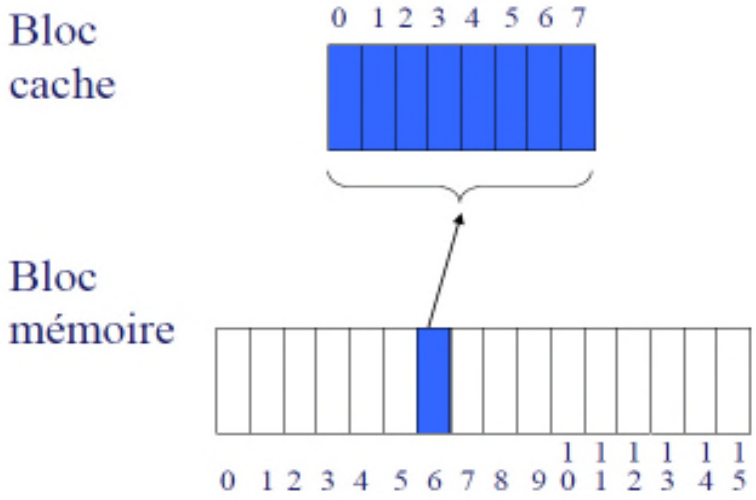


Figure 18 : Correspondance cache totalement associatif.

- **Correspondance associative par ensemble (set associative cache)** : séparation de la mémoire cache en groupes de blocs et associativité complète dans un groupe, c.à.d. le bloc **n** de la mémoire principale peut se retrouver dans n'importe quel bloc du groupe **g = (n modulo sg**) de la mémoire cache, sachant que **sg** est le nombre total de groupes de blocs dans la mémoire cache (fig. 19).

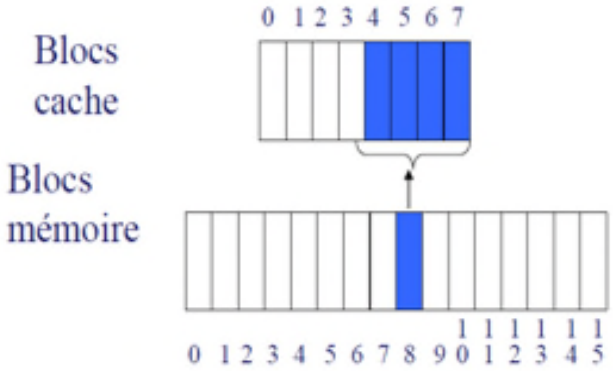


Figure 19 : La correspondance cache associative par ensemble.

Remarque :

La différence entre les correspondances cache et mémoire

Fonction de correspondance	*Avantages*	*Désavantages*
Placement direct	- Simple, - Peu - couteux - Bon choix pour les caches larges	- Très restrictive
Cache totalement associatif	- Meilleur taux de succès - Bon choix pour les caches petits	- Très couteux - Demande matériel - Demande une étiquette plus large
Cache associatif par ensemble de bloc	- Compromis - Préféré souvent	- Demande une étiquette large

De nos jours, la grande majorité des caches sont à correspondance directe ou a correspondance associative par ensemble de 2 ou 4 blocs.

Exercice : *Pentium 4 Prescott* ayant les caractéristiques de mémoire cache suivantes :

- L1 (données) : 16 Kbits ; lignes de 64bits ; associative par ensembles de 8
- L2 : 1 Mbits ; lignes de 128bits ; associative par ensembles de 8

1- Combien y-a-t-il de lignes dans cette mémoire cache ?

2- Combien y-a-t-il de blocs associatifs dans cette mémoire cache ?

Solution :

1- Combien y-a-t-il de lignes dans cette mémoire cache ?

Nombre de lignes L1 = Taille cache / Taille de la ligne = 16 Kbits / 64 bits

$= 2^4 * 2^{10} / 2^6 =$ **2^8 Lignes**

Nombre de lignes L2 = Taille cache / Taille de la ligne = 1 Mbits / 128 bits

$= 2^{20} / 2^7 =$ **2^{13} Lignes**

2- Combien y-a-t-il de blocs associatifs dans cette mémoire cache ?

Nombre de blocs L1= Nombre de lignes/ Nombre de lignes par bloc = $2^8 / 8 =$ **2^5 blocs**

Nombre de blocs L2= Nombre de lignes/ Nombre de lignes par bloc = $2^{13} / 8 =$ **2^{10} blocs**

e. Accès à un bloc du cache

Les adresses mémoires peuvent être construites en fonction de la correspondance entre mémoire principale et cache (fig. 20). Dans ce cas, l'adresse mémoire d'un mot contient des informations sur sa présence dans un bloc et sa présence éventuelle dans le cache. Elle se décompose en deux parties :

- Un **numéro de bloc**, qui se décompose en

  - un *index*, correspondant à l'emplacement de e bloc dans le cache

  - une *étiquette* permettant d'identifier le bloc mémoire correspondant au bloc placé dans le cache

- Un **déplacement** dans le bloc (le numéro du mot dans le bloc).

Ainsi, une table d'étiquette est maintenue, ce qui donne pour chaque bloc du cache l'étiquette du bloc mémoire placé dans ce bloc, ou le fait qu'aucun bloc mémoire n'a été copié dans ce bloc.

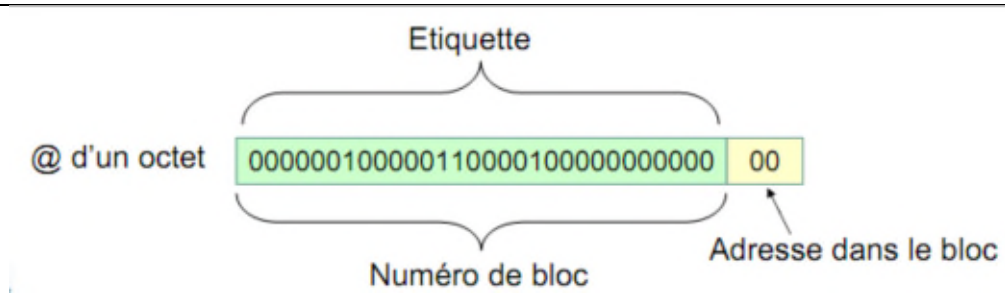
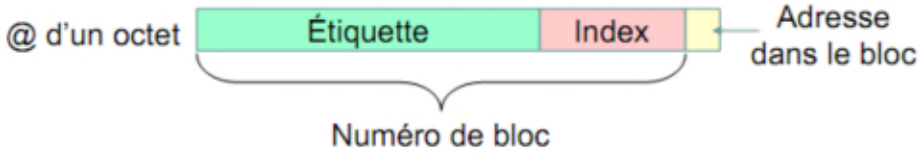
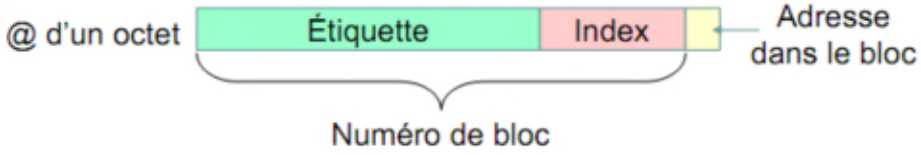
Cache totalement associatif	 Le numéro de bloc est utilisé comme étiquette.
Placement direct	 Le champ « numéro de bloc » est scindé en deux parties : l'étiquette et l'index.
Cache associatif par ensemble	 - Le choix d'un ensemble est associatif. - Cependant, chaque ensemble est géré comme dans le cache à correspondance direct.

Figure 20 : Format d'adresse mémoire cache.

f. Algorithme de remplacement

Si le cache est plein et que le processeur a besoin d'un bloc qui n'est pas dans le cache, il faut remplacer un des blocs du cache. Diverses stratégies sont employées, principalement :

- Choisir un bloc candidat de manière aléatoire (Random)
- Choisir le plus ancien bloc du cache (FIFO pour First In First Out)
- Choisir le bloc le moins récemment utilisé (LRU pour Least Recently Used) :
- Choisir le bloc le moins fréquemment utilisé (LFU pour Least Frequently Used)

Remarque :

- Dans un cache à accès direct le problème ne se pose évidemment pas. En revanche dans les caches associatifs, ou associatifs par ensemble, une stratégie doit être mise en œuvre.
- Les stratégies concernant l'utilisation (LFU, LRU) sont les plus efficaces, vient ensuite la stratégie aléatoire.
- Les stratégies aléatoires et FIFO sont plus faciles à implanter.

g. Interaction avec la mémoire centrale (lecture / écriture)

La lecture est l'opération la plus courante dans les caches. Toutes les instructions sont lues et la plupart d'entre elles ne provoquent pas d'écriture.

Les ***politiques de lecture*** lors d'un *échec* dans le cache sont :

- **Lecture immédiate (**en anglais **Read Through)** : la lecture se fait directement de la mémoire cache vers le CPU.
- **Lecture non immédiate (**en anglais **No Read Through)** : la lecture se fait de la mémoire centrale vers le cache puis du cache vers le CPU.

Deux ***politiques d'écriture*** sont employées pour traiter le cas d'un *succès* dans le cache :

- **Ecriture immédiate ou simultanée (**en anglais **Write Through)** : l'information est écrite dans le cache et dans la mémoire centrale.
- **Ecriture remplacement ou réécriture (**en anglais **Write Back)** : l'information est écrite uniquement dans le cache. Elle est écrite dans la mémoire centrale seulement lors d'un remplacement. Un bit, appelé **dirty bit**, indique pour chaque voie s'il est nécessaire de mettre à jour la voie en mémoire centrale.

Il y a également deux politiques lors d'une écriture sur une information non présente dans le cache :

- **Ecriture allouée (**en anglais **Write Allocate)** : l'information est d'abord chargée dans le cache puis modifiée.
- **Ecriture non allouée (**en anglais **No Write Allocate)** : l'information est directement modifiée dans la mémoire centrale et n'est pas chargée dans le cache.

Résume des politiques de lecture et d'écriture :

<table>
<tr><th colspan="2">En cas d'échec d'écriture</th><th colspan="2">Ecriture dans le cache</th></tr>
<tr><th colspan="2"></th><th>Oui</th><th>Non</th></tr>
<tr><th rowspan="3">Ecriture dans la mémoire</th><th>Oui</th><td rowspan="2">Ecriture immédiate + Ecriture allouée</td><td>Ecriture immédiate + Ecriture non allouée</td></tr>
<tr><td>Réécriture + Ecriture non allouée</td></tr>
<tr><th>Non</th><td>Réécriture + Ecriture allouée</td><td></td></tr>
</table>

h. Performance

On peut évaluer la performance d'une mémoire utilisant un cache par le calcul du temps d'accès mémoire moyen :

Temps D'accès Mémoire Moyen = Temps D'accès Succès + Taux D'échec * Pénalité D'échec

Temps D'accès Succès = Temps D'accès A Une Donnée Résidant Dans Le Cache

Taux D'échec = Nombre De Défauts De Cache / Nombre D'accès Cache Ou = 1 - Taux De Succès

Taux De Succès = Nombre De Succès / Nombre D'accès Cache

Sachant que : **Temps D'accès Succès << Pénalité D'échec**

Exercice :

A partir des performances du tableau ci-dessous, calculer le temps d'exécution moyen d'une instruction pour chaque niveau sachant que la durée d'un cycle horloge est **T.**

Niveau	Temps d'accès succès (ns)	Taux de succès (ns)	Pénalité d'échec (Cycles)	Taille
Cache L1	3	80%	5	128 Ko
Cache L2	5	90%	10	512 Ko

Solution :

On a le **Temps d'accès mémoire moyen = temps d'accès succès + taux d'échec * pénalité d'échec**

Et **taux d'échec = 1 - taux de succès**

Donc **temps d'accès mémoire moyen Cache L1** = 3 + (1-80%)*5= **4T**

Et **temps d'accès mémoire moyen Cache L2** = 5 + (1-90%)*10= **6T**

2.8.4. Avantages et inconvénients

a. Avantages de la mémoire cache

- Elle est très rapide d'accès plus que la mémoire principale.
- Elle consomme moins de temps d'accès par rapport à .la mémoire.
- Elle stocke le programme qui peut être exécuté dans un temps court.
- Elle stocke les données pour une utilisation temporaire.

b. Inconvénients de la mémoire cache

- Elle a une capacité limitée.
- Elle est très coûteuse.

2.9. Hiérarchie de mémoires

Les différents éléments de la mémoire d'un ordinateur sont ordonnés en fonction des critères : temps d'accès, capacité et coût par bit (fig. 21).

1- Les éléments de mémoire situés dans l'unité centrale de traitement (CPU) sont les **registres** qui sont caractérisés par une grande vitesse et servent principalement au stockage d'opérandes et des résultats intermédiaires.
2- La **mémoire cache (**ou **l'antémémoire) :** est une mémoire rapide de faible capacité (par rapport à la mémoire centrale). Utilisée comme mémoire tampon entre le CPU et la mémoire centrale. Cette mémoire permet au CPU de faire moins d'accès à la mémoire centrale et ainsi de gagner du temps.
3- La **mémoire centrale :** est l'organe principal de rangement des informations utilisées par le CPU. Pour exécuter un programme, il faut le charger (instructions + données) en mémoire centrale. Cette mémoire est une mémoire à semi-conducteur, mais son temps d'accès est beaucoup plus grand que celui des registres et du cache.
4- La **mémoire d'appui :** sert de mémoire intermédiaire entre la mémoire centrale et les mémoires auxiliaires. Elle est présente dans les ordinateurs les plus évolués et permet d'augmenter la vitesse d'échange des informations entre ces deux niveaux.
5- La **mémoire de masse (**ou **mémoire auxiliaire) :** est une mémoire périphérique de grande capacité et de coût relativement faible, utilisée pour le stockage permanent des informations. Elle est utilisée pour le stockage, la sauvegarde ou l'archivage à long terme des informations. Elle utilise pour celà des supports magnétiques (disques, cartouches, bandes), magnéto-optiques(disques) ou optiques (disques optiques).

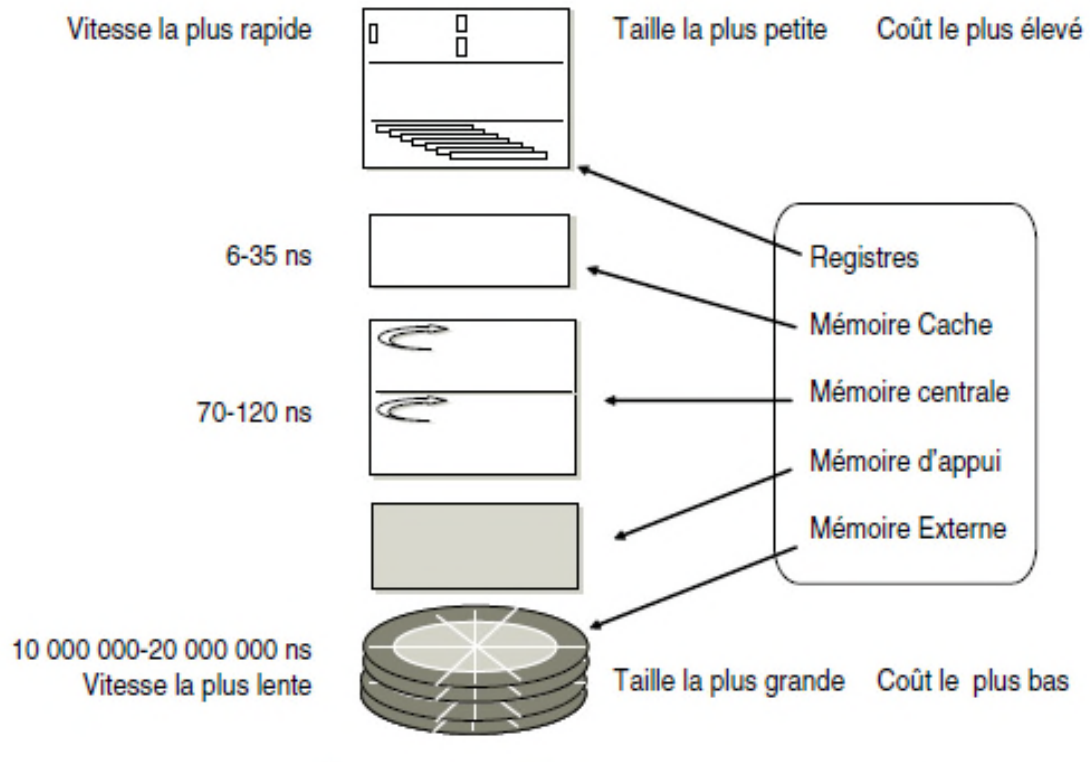


Figure 21: La hiérarchie de mémoires.

Exercice : Classez les mémoires suivantes par taille, par rapidité : CD-ROM, Registre d'Instruction, Disques durs, ROM, Cache L1, USB, Cache L2.
Solution :
Par taille : RI < L1 < L2 < ROM < CD <USB < DD.
Par vitesse : RI > L1 > L2 > ROM > DD > USB > CD.