Chapitre 2 : Principaux composants d'un ordinateur

1- Introduction aux principaux composants d'un ordinateur
2- Le processeur
3- L'UAL
4- Les bus
5- Les registres
6- La mémoire interne : mémoire RAM (SRAM et DRAM), ROM, temps d'accès, latence,...
7- La mémoire cache : utilité et principe, algorithmes de gestion du cache (notions de base)
8- Hiérarchie des mémoires
9- Conclusion

1. Introduction aux principaux composants d'un ordinateur

Un ordinateur est une machine programmable universelle de traitement de l'information.

Pour accomplir sa fonction, il doit pouvoir :

- **Acquérir** de l'information de l'extérieur
- **Stocker** en son sein ces informations
- **Combiner** entre elles les informations à sa disposition
- **Restituer** ces informations à l'extérieur

L'ordinateur doit donc posséder :

- **Une ou plusieurs unités de stockage**, pour mémoriser le programme en cours d'exécution ainsi que les données qu'il manipule.
- **Une unité de traitement** permettant l'exécution des instructions du programme et des calculs sur les données qu'elles spécifient.
- **Différents dispositifs 'périphériques'** servant à interagir avec l'extérieur : clavier, écran, souris, carte graphique, carte réseau, etc.

Les constituants de l'ordinateur sont reliés par un ou plusieurs bus, ensembles de fils parallèles servant à la transmission des adresses, des données, et des signaux de contrôle.

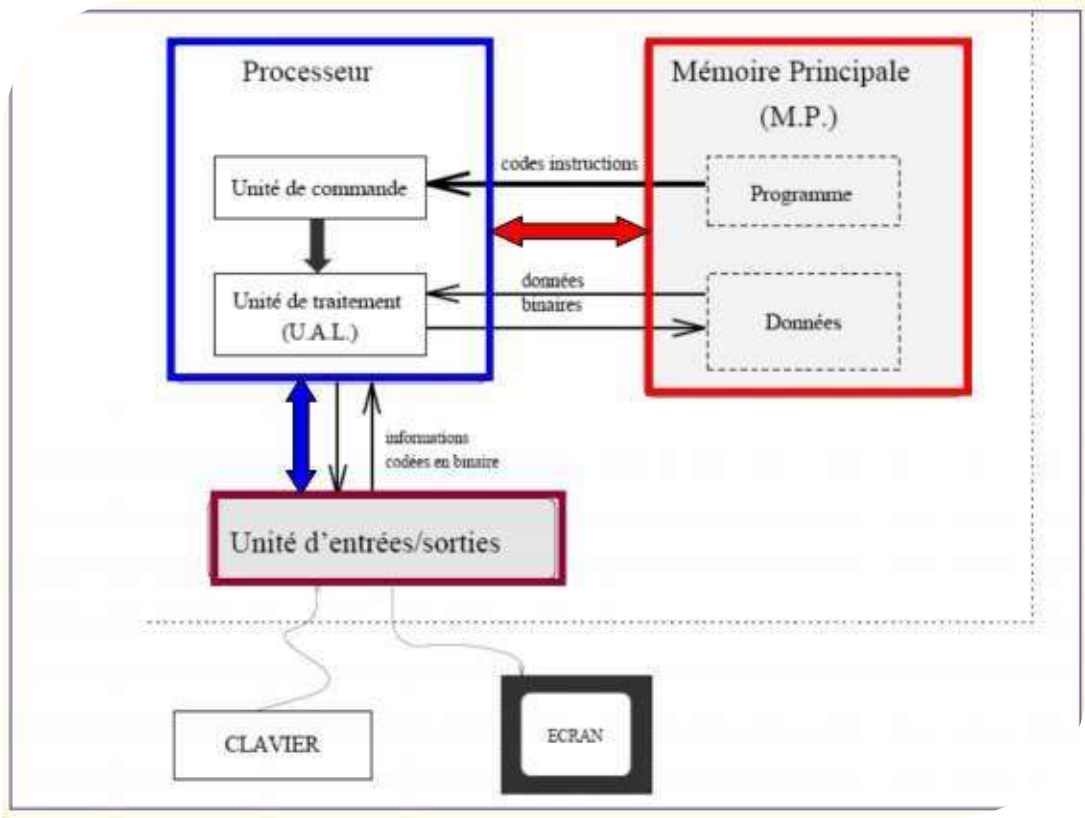


Figure 6. L'architecture de base d'un ordinateur

1.3. Les composants de l'ordinateur

Le traitement de l'information par l'ordinateur nécessite la coopération de plusieurs éléments. Au niveau de l'ordinateur, on distingue deux parties :

- *Une partie matérielle (Hardware) :* elle renvoie à la construction physique de la machine.
- *Une partie logicielle (Software) :* elle est constituée de l'ensemble des programmes pouvant être un programme d'application ou un programme de pilotage ou de base.

Les composants physique (ou matérielle) de l'ordinateur est composée en général : d'une unité centrale et de différents périphériques.

Dans le boîtier (unité centrale) est monté une *carte-mère* où sont implantés les lignes du *bus* et les principaux circuits électroniques : processeur, chipset, RAM *(Mémoire vive ; en anglais Random Acces Memory)*, ROM *(Mémoire Morte ; en anglais Read Only Memory),* connecteurs et câble de liaisons, les cartes contrôleurs et sorties des différents périphériques.

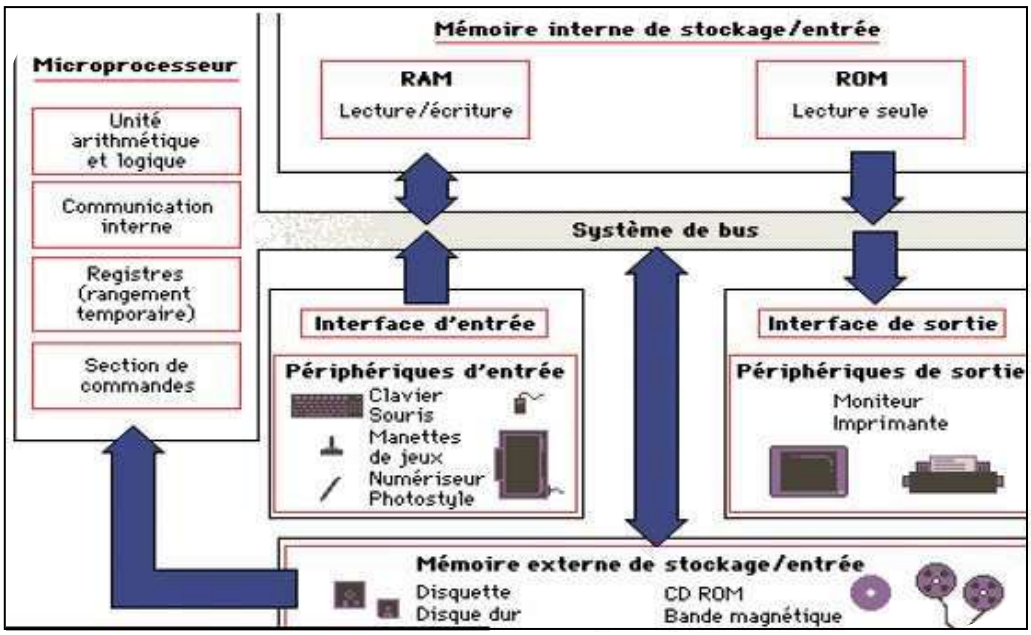


Figure 7. Structure d'un ordinateur

1.4. La carte mère

La carte mère *(en anglais « **Mainboard** » ou « **Motherboard** »)* est le socle permettant la connexion de l'ensemble des éléments essentiels de l'ordinateur. La carte mère est une carte maîtresse, prenant la forme d'un grand circuit imprimé possédant notamment des connecteurs pour les cartes d'extension, les barrettes de mémoires, le processeur, etc.

a. Les caractéristiques d'une carte mère

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

- ***Le facteur d'encombrement ou facteur de forme :*** définit la géométrie, les dimensions, l'agencement et les caractéristiques électriques de la carte mère.
- ***Le chipset :*** circuit électronique chargé de coordonner les échanges de données entre les divers composants de l'ordinateur (processeur, mémoire...).
- ***Le type de support de processeur :*** On distingue deux catégories de supports :
  - **Slot** (en français fente) : il s'agit d'un connecteur rectangulaire dans lequel on enfiche le processeur verticalement.
  - **Socket** (en français embase) : il s'agit d'un connecteur carré possédant un grand nombre de petits connecteurs sur lequel le processeur vient directement s'enficher.
- ***Les connecteurs de mémoire vive*** **(RAM) :** Les connecteurs d'extension sont des réceptacles dans lesquels il est possible d'insérer des cartes d'extension, c'est-à-dire

des cartes offrant de nouvelles fonctionnalités ou de meilleures performances à l'ordinateur.

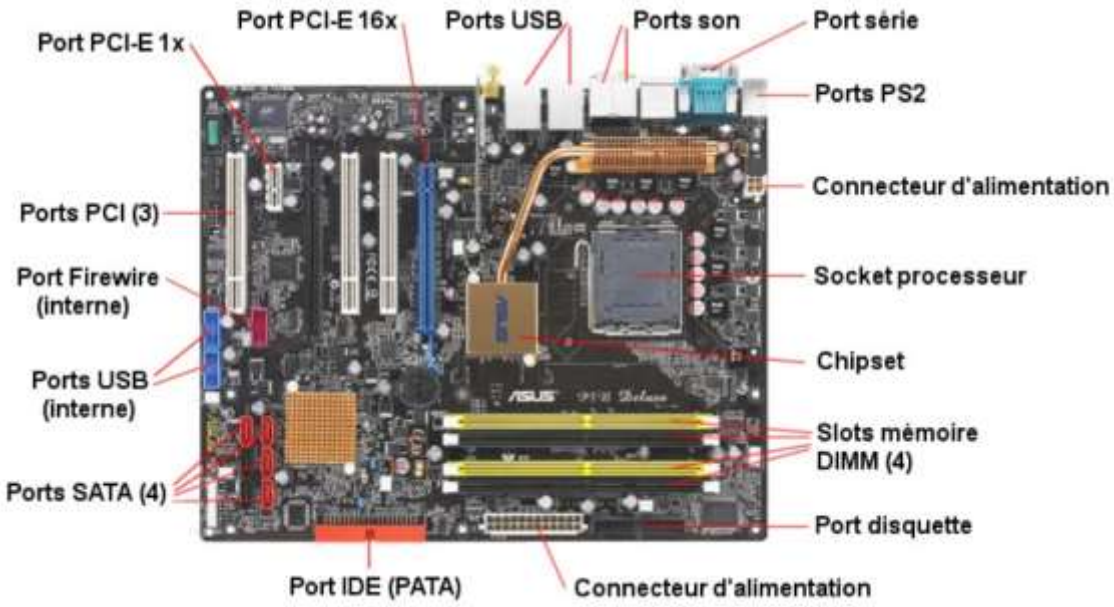
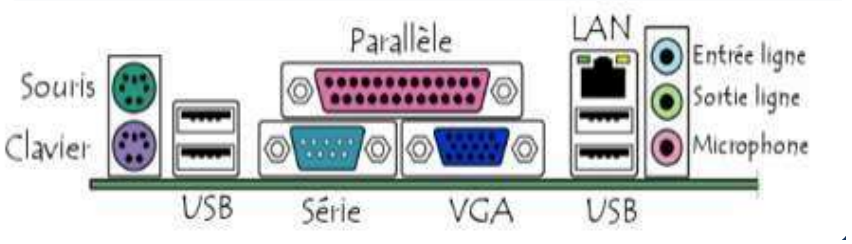


Figure 8. Principaux composants d'une carte mère

b. Les types de connecteurs

Il existe plusieurs sortes de connecteurs :

- ***Les connecteurs d'entrée-sortie*** : La carte mère possède un certain nombre de connecteurs d'entrées-sorties regroupés sur le « panneau arrière ».



Vue arrière du PC - branchement des périphériques

La plupart des cartes mères proposent les connecteurs suivants :

- **Port série**, permettant de connecter de vieux périphériques ;

- **Port parallèle**, permettant notamment de connecter de vieilles imprimantes ;
- **Ports USB** « Universal Serial Bus » (1.1, bas débit, ou 2.0, haut débit), permettant de connecter des périphériques plus récents ;
- **Connecteur RJ45** « Registered Jack » (appelés LAN ou port Ethernet) permettant de connecter l'ordinateur à un réseau. Il correspond à une carte réseau intégrée à la carte mère.
- **Connecteur VGA** « Video Graphics Array » (appelé SUB-D15), permettant de connecter un écran. Ce connecteur correspond à la carte graphique intégrée.
- **Prises audio** (entrée Line-In, sortie Line-Out et microphone), permettant de connecter des enceintes acoustiques ou une chaîne hi-fi, ainsi qu'un microphone. Ce connecteur correspond à la carte son intégrée

2. Le processeur

2.1. Définition

Le **processeur** (noté **CPU**, pour Central Processing Unit) est un circuit électronique cadencé au rythme d'une horloge interne, grâce à un cristal de quartz qui, soumis à un courant électrique, envoie des impulsions, appelées « **top** ».

2.2. Le rôle du processeur

Le **processeur** (**CPU**, pour Central Processing Unit, soit Unité Centrale de Traitement) est le cerveau de l'ordinateur. Il permet de manipuler des informations numériques, c'est-à-dire des informations codées sous forme binaire, et d'exécuter les instructions stockées en mémoire.

2.3. La structure du processeur

Le processeur est constitué d'un ensemble d'unités fonctionnelles reliées entre elles. Les rôles des principaux éléments d'un processeur sont les suivants :

1) Une **unité d'instruction** (ou unité de commande, en anglais control unit) 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** identifier 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 données l'unité d'instruction. L'unité d'exécution est notamment composée des éléments suivants :

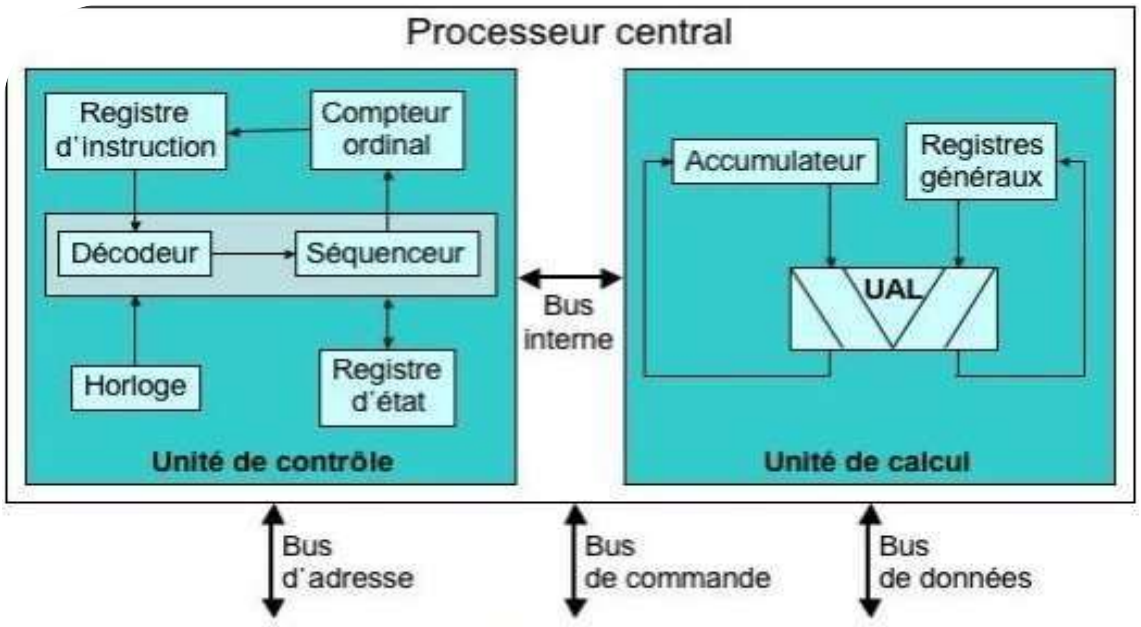
3) L'**unité arithmétique et logique** (notée **UAL** ou en anglais ALU pour Arithmetical and Logical Unit) pour le traitement des données.

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 ;



Architecture simplifiée d'un processeur central

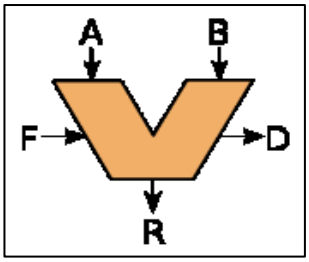
3. L'Unité Arithmétique et Logique (UAL)

3.1 Définition

L'unité arithmétique et logique, abrégée UAL, est l'organe de l'ordinateur chargé d'effectuer les calculs. Elle est incluse dans le microprocesseur.

Elle effectue les opérations spécifiées par les instructions et exécute les calculs arithmétiques (ex : addition) et logique (ex : comparaison).

C'est un circuit combinatoire qui produit un résultat (R) sur n bits fonction des données présentes sur ses entrées (A et B) et de la fonction à réaliser (F) et met à jour des drapeaux.



Une unité arithmétique et logique à deux entrées

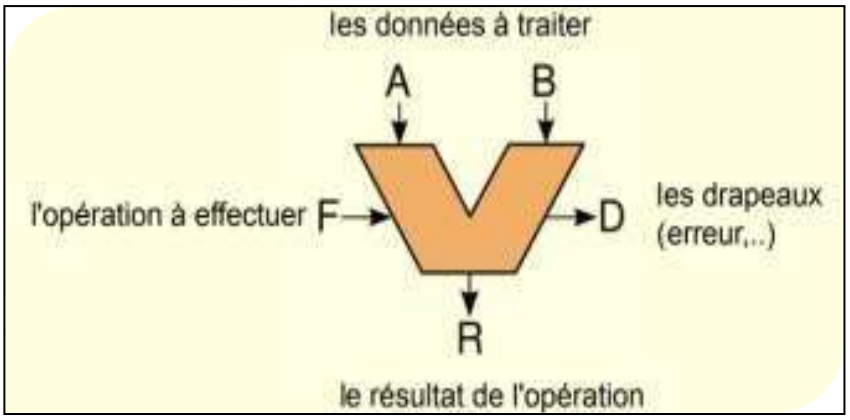
3.2 Fonctionnement de l'UAL

L'**UAL** permet de réaliser différents types d'opérations sur des données de la forme R=F(A,B) :

- *Des opérations arithmétiques* : additions, soustractions, ...
- *Des opérations logiques* : ou, et, ou exclusif, ...

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

- **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). o **C** (Carry-out) : mis à 1 en cas de retenue ou débordement en contexte non signé. o **V** (Overflow): mis à 1 en cas de débordement en contexte signé.



Fonctionnement de l'UAL

4. Les bus

En informatique, le mot bus désigne l'ensemble des liaisons électrique (nappes, pistes de circuits imprimés, etc.) utilisés par plusieurs éléments matériels afin de communiquer

entre eux. Si cette liaison relie deux éléments seulement, elle est appelée *port matériel* (port série, port parallèle, etc.).

Un bus est un ensemble de fils permettant de lier et faire communiquer les composants d'un ordinateur afin d'assurer la transmission du même type d'information (données, adresses ou commandes).

4.1. Caractéristiques d'un bus Un bus est caractérisé par :

a) **Sa largeur :** un bus est caractérisé par le volume d'informations transmises simultanément (exprimé en bits). La **largeur** désigne le nombre de bits qu'un bus peut transmettre simultanément.

1 fil transmet **un bit, 1bus à n fils = 1 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.

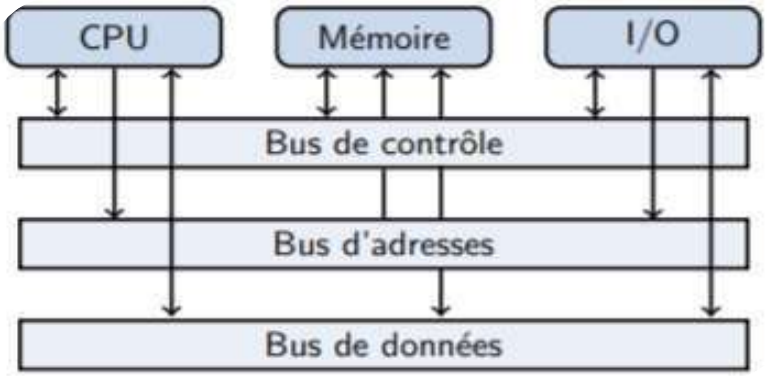
Débit (octets/s) = (Nombre de transferts par seconde x Largeur) / 8

Bande passante (en Mo/s) = Largeur bus (en octets) x Fréquence (en MHz)

4.2. Différents types de bus

On peut distinguer trois types de bus véhiculant des informations en parallèle dans un système de traitement programmé de l'information :

- **Un bus de données** : bidirectionnel qui assure le transfert des informations entre le microprocesseur et son environnement, et inversement. Son nombre de lignes est égal à la capacité de traitement du microprocesseur.
- **Un bus d'adresses :** unidirectionnel qui permet la sélection des informations à traiter dans un espace mémoire (ou espace adressable) qui peut avoir 2^n emplacements, avec **n** = nombre de conducteurs du bus d'adresses. L'espace mémoire adressable dépend de la largeur du bus d'adresses.
- **Un bus de commande** : constitué par quelques conducteurs qui assurent la synchronisation des flux d'informations sur les bus des données et des adresses.

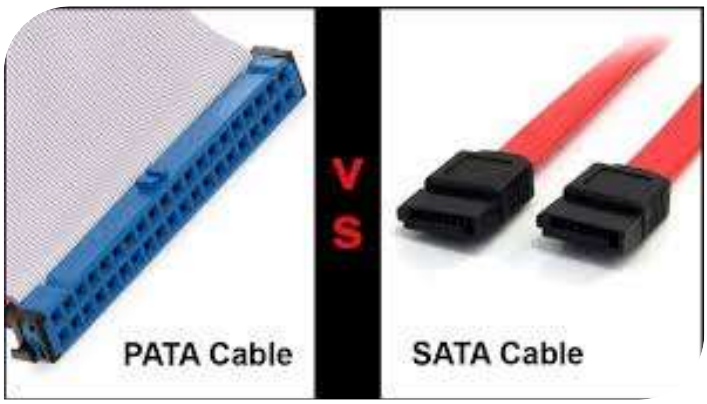


Différents types de bus selon la nature de l'information à transporter

4.3. Types de bus de données

Il existe deux grands types de bus de données selon le type de transmission :

- **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.
- **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 ; la mémoire.
  Exemple : Bus PATA.



Câble PATA et câble SATA

4.4. Les principaux bus

On distingue généralement sur un ordinateur deux principaux bus :

- **Le bus système** (*bus interne, en anglais internal bus ou front-side bus, noté FSB*). Le bus système permet au processeur de communiquer avec la mémoire centrale du système (mémoire vive ou RAM).
- **Le 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 raccordées.

5. Les registres

5.1. Définition

Un registre est un emplacement de mémoire interne à un processeur. Les registres se situent au sommet de la hiérarchie mémoire : il s'agit de la mémoire la plus rapide d'un ordinateur, mais dont le coût de fabrication est le plus élevé, car la place dans un microprocesseur est limitée.

Chaque registre peut stocker une valeur entière distincte, bornée par la taille des registres (nombre de bits)

5.2. Les principaux registres

Certains registres sont spécialisés, comme :

- **Le compteur ordinal (« Program Counter » CO ou PC)** qui stocke l'adresse de la prochaine instruction à exécuter
- **Le registre d'instruction (« Instruction Register » RI ou IR),** qui stocke l'instruction en cours d'exécution
- **L'accumulateur (Acc)**, registre résultat de l'UAL, etc.

5.3. Les registres de l'UAL

Il y a aussi **les registres de l'UAL** ; qui sont accessibles au programmeur, contrairement aux registres de l'UCC. On dénombre :

- **Registres arithmétiques** : destinés pour les opérations arithmétiques (+, -, *, /, complément à1, ...) 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 d'état** (PSW, Processor Status Word) **:** permettant de stocker des indicateurs sur l'état du système (retenue, dépassement, etc.) ;
- **Registre pointeur :** d'une pile ou de son sommet.
- **Registres généraux :** pour diverses opérations, ex., stocker des résultats intermédiaires
- **Registres spécialisés :** destinés pour certaines opérations comme les registres de décalages, registres des opérations arithmétiques à virgule flottante, …etc.

6. La mémoire interne : mémoire RAM (SRAM et DRAM), ROM, temps d'accès, latence,...

6.1. La 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 de contenir, de conserver et de restituer sans les modifier de grandes quantités d'information (instructions + données).

6.2. Type de mémoire

Il existe deux types de mémoire dans un système informatique :

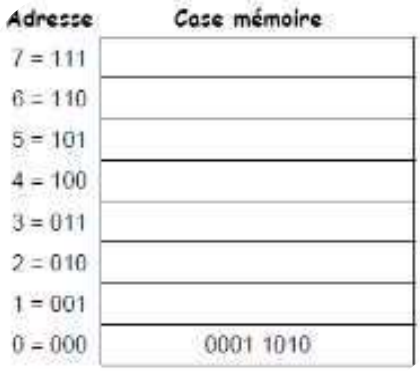
- **La mémoire centrale** qui est très rapide, physiquement peu encombrante mais coûteuse, c'est la mémoire de travail de l'ordinateur,
- **La mémoire de masse** ou mémoire auxiliaire, qui est plus lente, assez encombrante physiquement, mais meilleur marché, c'est la mémoire de « sauvegarde » des informations.

6.3. Caractéristiques d'une mémoire

- **La capacité :** c'est le nombre total de bits que contient la mémoire. Elle s'exprime aussi souvent en octet.
- **Le format des données :** c'est le nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du mot mémorisable.
- **Le temps d'accès :** c'est le temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture en mémoire et l'instant où la première information est disponible sur le bus de données.
- **Le temps de cycle :** il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture.
- **Le débit :** c'est le nombre maximum d'information lues ou écrites par seconde.
- **Volatilité :** elle caractérise la permanence des informations dans la mémoire. L'information stockée est volatile si elle risque d'être altérée par un défaut d'alimentation électrique et non volatile dans le cas contraire.

6.4. Organisation d'une mémoire

Une mémoire peut être représentée comme une armoire de rangement constituée de différents tiroirs. Chaque tiroir représente alors une case mémoire qui peut contenir un seul élément : des **données**. Le nombre de cases mémoires pouvant être très élevé, il est alors nécessaire de pouvoir les identifier par un numéro. Ce numéro est appelé **adresse**. Chaque donnée devient alors accessible grâce à son adresse.



Organisation de la mémoire

- Avec une adresse de ***n* bits** il est possible de référencer au plus 2^n cases mémoire. Chaque case est remplie par un mot de données (sa longueur ***m*** est toujours une puissance de 2). Le nombre de fils d'adresses d'un boîtier mémoire définit donc le nombre de cases mémoire que comprend le boîtier. Le nombre de fils de données définit la taille des données que l'on peut sauvegarder dans chaque case mémoire.

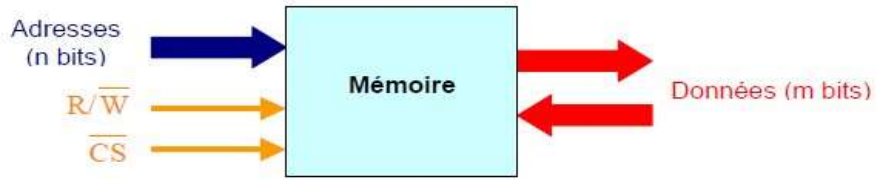
Capacité (en bits) = 2nombre de lignes d'adresse x nombre de lignes de données

Nombre de Mots = Capacité / taille du mot

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

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

- En plus du bus d'adresses et du bus de données, un boîtier mémoire comprend une entrée de commande qui permet de définir le type d'action que l'on effectue avec la mémoire (lecture/écriture) et une entrée de sélection qui permet de mettre les entrées/sorties du boîtier en haute impédance.
- On peut donc schématiser un circuit mémoire par la figure suivante où l'on peut distinguer :



- les entrées d'adresses
- les entrées de données
- les sorties de données
- les entrées de commandes :
  - une entrée de sélection de lecture ou d'écriture. (R/$\overline{W}$)
  - une entrée de sélection du circuit. ($\overline{CS}$)

Circuit mémoire

- Une opération de lecture ou d'écriture de la mémoire suit toujours le même cycle :
  1) sélection de l'adresse
  2) choix de l'opération à effectuer (R/$\overline{W}$)
  3) sélection de la mémoire ($\overline{CS} = 0$)
  4) lecture ou écriture la donnée

6.5. Classification des mémoires

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

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

6.6. Types d'accès à la mémoire

Le mode d'accès à une mémoire dépend surtout de l'utilisation qu'on veut en faire et il existe trois types :

- **Par le contenu** : mémoire adressable par le contenu (ex. mémoire cache). La recherche s'effectue en parallèle sur toutes les cases mémoires via une clé et non via un index numérique. Le temps d'accès est constant.
  - Les opérations associées à ce mode d'accès : écriture (clé, donnée) ; lecture (clé) ; existe (clé) ; retirer (clé)
- **Aléatoire (ex., pour la mémoire vive)** : via une adresse Mémoire à accès aléatoire [Random Access Memory (RAM)] : 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)
- **Direct ou semi séquentiel (ex. les disques durs, CDs, ...)** : accès à un bloc de données ou cylindre (contenant la donnée recherchée) via son adresse 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)

6.7. La mémoire interne (centrale ou principale)

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

- C'est l'organe principal de **rangement** des informations utilisées par le processeur.
- Dans une machine (ordinateur / calculateur) pour **exécuter** un programme il faut le charger (copier) dans la mémoire centrale.
- Le **temps d'accès** à la mémoire centrale et **sa capacité** sont deux éléments qui influent sur le **temps d'exécution** d'un programme (performance d'une machine).
- 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.

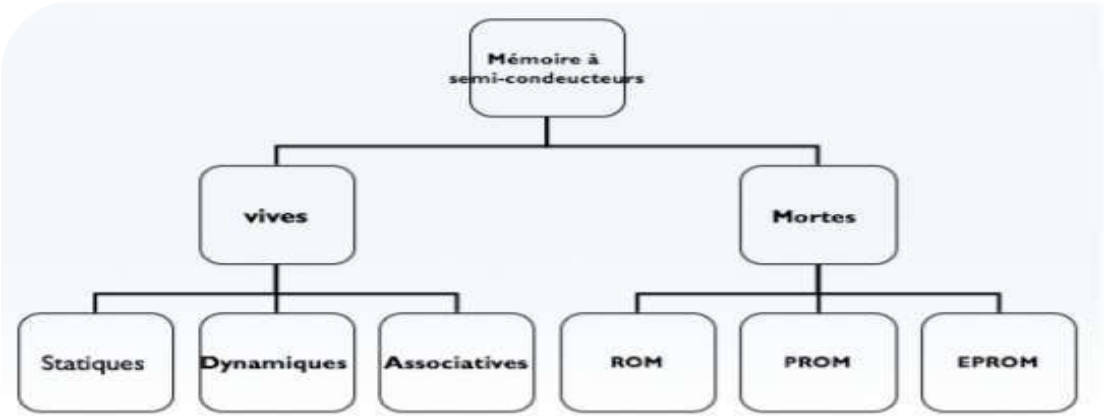


Schéma des différents types de mémoire

Les mémoires vives (RAM)

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) :

a) **Les RAM statiques (SRAM) :** Le bit mémoire d'une RAM statique (SRAM) est composé d'une bascule. Chaque bascule contient entre 4 et 6 transistors.

b) **Les RAM dynamiques (DRAM) :** Dans les RAM dynamiques (DRAM), l'information est mémorisée sous la forme d'une charge électrique stockée dans un condensateur (capacité grille substrat d'un transistor MOS).

a) Les mémoires statiques

Dans la mémoire vive statique ou SRAM (Static Random Access Memory), la cellule de base est constituée par une bascule de transistors.

Le terme de statique fait référence à leur fonctionnement interne. Elles ne nécessitent quasiment pas de rafraichissement.

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.

On utilisera donc essentiellement pour des mémoires de faible capacité comme dans la mémoire cache pour les microprocesseurs.

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
- Son inconvénient réside dans les courants de fuite des pico-condensateurs : l'information disparaît à moins que la charge des condensateurs ne soit rafraîchie avec une période de quelques millisecondes d'où le terme de dynamique.

!!! Il ne faut pas confondre SRAM et SDRAM :

- Une **SRAM** est une mémoire statique (SRAM = **S**tatique RAM) construite avec des bascules.

- Une **SDRAM** est une mémoire dynamique DRAM qui fonctionne à la vitesse du bus mémoire, elle est donc synchrone avec le fonctionnement du processeur le "S" indique la synchronicité (SDRAM = Synchrone DRAM).

- Une **DDR 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. Le sigle DDR signifie **D**ouble **D**ata **R**ate.

- **VRAM [Video RAM] :** si elle a 2 ports pour pouvoir être accédée simultanément en lecture et en écriture
- **Mémoire flash :** 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, ...).
- **Modules mémoire DIMM (RAM) [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.
- **Modules SIMM [Single In-line Memory Module]** : idem à DIMM mais en 32 bits

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.

Les mémoires mortes (ROM)

Les mémoires mortes ou mémoires à lecture seule (ROM : Read Only Memory) sont non volatiles. Ces mémoires, contrairement aux RAM, ne peuvent être que lue. L'inscription en mémoire des données reste possible mais est appelée programmation. Suivant le type de ROM, la méthode de programmation changera. Il existe donc plusieurs types de ROM:

a) **ROM :** Elle est programmée par le fabricant et son contenu ne peut plus être ni modifié, ni effacé par l'utilisateur.

b) **PROM :** C'est une ROM qui peut être programmée une seule fois par l'utilisateur (Programmable ROM). La programmation est réalisée à partir d'un programmateur spécifique.

c) **EPROM *ou UV-EPROM :*** Pour faciliter la mise au point d'un programme ou tout simplement permettre une erreur de programmation, il est intéressant de pouvoir reprogrammer une PROM. La technique de claquage utilisée dans celles-ci ne le permet évidemment pas. L'EPROM *(*Erasable Programmable ROM*)* est une PROM qui peut être effacée.

d) **EEPROM :** pour (Electically EPROM) est une mémoire programmable et effaçable électriquement. Elle répond ainsi à l'inconvénient principal de l'EPROM et peut être programmée in situ.

e) **FLASH EPROM :** La mémoire Flash s'apparente à la technologie de l'EEPROM. Elle est programmable et effaçable électriquement comme les EEPROM.

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

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 Cette sortie va nous permettre de sélectionner un seule mot mémoire.

Lecture et écriture de l'information

- **Comment lire une information ?**

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

  – Charger dans le registre RAM l'adresse du mot à lire.
  – Lancer la commande de lecture (R/W=1)
  – L'information est disponible dans le registre RIM au bout d'un certain temps (temps d'accès)

- **Comment écrire une information ?**

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

  – Charger dans le RAM l'adresse du mot ou se fera l'écriture.
  – Placer dans le RIM l'information à écrire.
  – Lancer la commande d'écriture pour transférer le contenu du RIM dans la mémoire.

7. La mémoire cache : utilité et principe, algorithmes de gestion du cache (notions de base)

7.1. Utilité

La mémoire cache ou ***antémémoire*** est une mémoire très rapide d'accès pour le microprocesseur. Elle agit comme un tampon entre le processeur et la mémoire principale. Elle est utilisée pour maintenir les parties de données et programmes qui sont le plus fréquemment utilisé 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.

Exemple de mémoire cache

7.2. Organisation en niveau

 Les processeurs récents possèdent plusieurs niveaux de mémoire cache : Niveaux L1 et L2, voire L3 pour certains processeurs.

- La **mémoire cache de premier niveau** (appelée **L1 Cache**, pour Level **1 Cache**) est directement intégrée dans le processeur. Il est généralement scindé en 2 parties (Instructions / Données). Les caches du premier niveau sont très rapides d'accès. Leur délai d'accès tend à s'approcher de celui des registres internes aux processeurs.
- Le **cache L2** est situé entre le cache **L1** et la mémoire vive. Il est moins rapide que le cache **L1**.
- Le **cache L3,** autrefois situé au niveau de la carte mère, est aujourd'hui intégré dans le CPU.

 Le cache Li+1 joue le rôle de cache pour le niveau Li

 Le cache Li+1 plus grand que Li mais moins rapide en temps d'accès aux donnés

7.3.Relation entre les niveaux de cache

 Cache inclusif

- Le contenu du niveau L1 se trouve aussi dans L2
- Taille globale du cache : celle du cache L2

 Cache exclusif

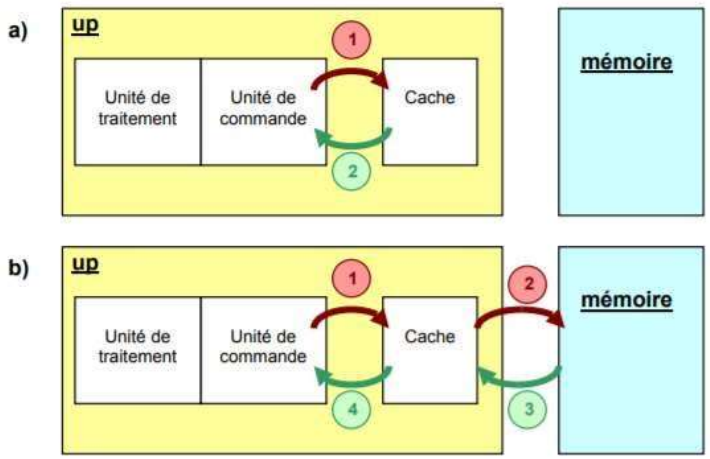
- Le contenu des niveaux L1 et L2 sont différents
- Taille globale du cache : taille L1 + taille L2

7.4.Principe

Le principe de cache est très simple : le microprocesseur n'a pas conscience de sa présence et lui envoie toutes ses requêtes comme s'il agissait de la mémoire principale :

 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)**

 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)**



Succès de cache / défaut de cache

- 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, 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 cache de niveau L1 et celui de niveau L2 peuvent être regroupés dans la même puce que le processeur (**cache interne/ on-chip**) ou n'être qu'accessible via un bus externe au processeur (**external cache**).
- 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 lent qu'un accès à une information contenue dans un registre.

7.5.Gestion de la mémoire cache

- **Définitions**
  - **Ligne :** est le plus petit élément de données qui peut être transféré entre la mémoire cache et la mémoire de niveau supérieur. (taille de la ligne = taille du bloc)
  - **Mot :** est le plus petit élément de données qui peut être transféré entre le processeur et la mémoire
- **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.

1) **Localité spatiale** qui indique que l'accès une instruction située à une adresse X va probablement être suivi d'un accès à une zone tout proche de X

   Exemple : tableaux, structures.

   La localité spatiale suggère de copier des blocs de mots dans le cache plutôt que des mots isolés.
2) **Localité temporelle** : qui 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.

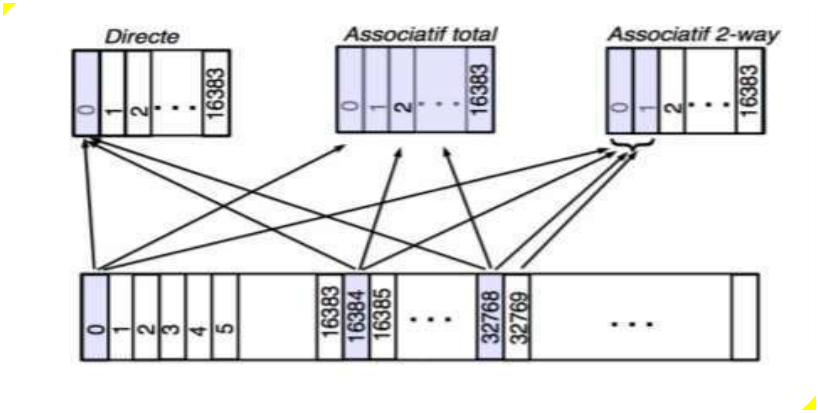
La localité temporelle suggère de conserver pendant quelque temps dans le cache les informations auxquelles on vient d'accéder.

7.6.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 :

1) **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, **sb** étant la taille en nombre de blocs de la mémoire cache ;
2) **Correspondance totalement associative (fully associative cache)** : chaque bloc mémoire peut être placé dans n'importe quel bloc du cache
3) **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, **sg** étant le nombre total de groupes de blocs dans la mémoire cache.



7.7.Correspondance cache et mémoire (le « Mapping »)

Accès à un bloc du cache

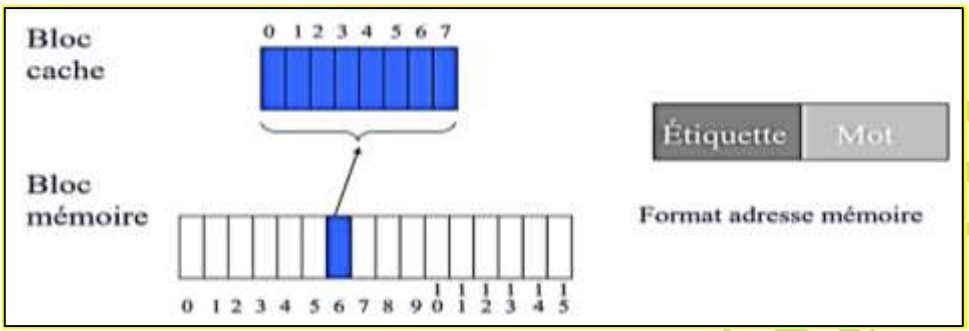
Les adresses mémoires peuvent être construites en fonction de la correspondance entre mémoire principale et cache. Dans ce cas, l'adresse mémoire d'un mot contient des informations sur sa présence dans un bloc et sa présence éventuelle dans le cache. Elle se décompose en deux parties :

- 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).

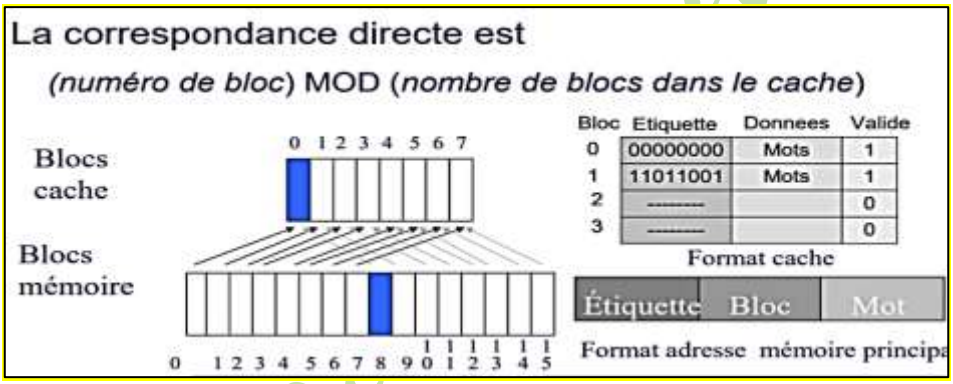
Ainsi, une table d'étiquette est maintenue, qui donne pour chaque bloc du cache l'étiquette du bloc mémoire placé dans ce bloc, ou le fait qu'aucun bloc mémoire n'a été copié dans ce bloc.

Exemples :

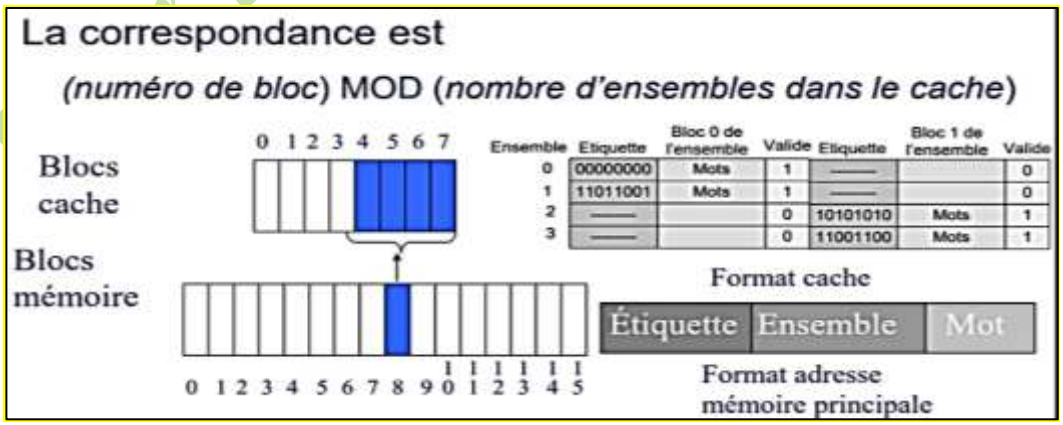
1. **Correspondance totalement associative**



2. **Correspondance directe**



3. **Correspondance associative par ensemble**



7.8.Algorithmes 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
- choisir le plus ancien bloc du cache (FIFO, First In First Out)
- choisir le bloc le moins récemment utilisé (LRU Least Recently Used)
- choisir le bloc le moins fréquemment utilisé (LFU Least Frequently Used)

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.

7.9.Politique d'écriture :

Considérons le cas d'une opération d'écriture. Deux situations se présentent selon que le bloc dans lequel on souhaite écrire se trouve dans le cache ou non. Dans le premier cas, on peut choisir

- **Ecriture immédiate** : écrire à la fois dans le bloc du cache et dans le bloc de la mémoire (écriture simultanée, ou write through)
- **Ecriture remplacement** : écrire uniquement dans le bloc du cache, et différer l'écriture de e bloc en mémoire lorsque l'emplacement qu'il occupe sera désigné pour recevoir un nouveau bloc mémoire (réécriture ou write back).

Dans le deuxième cas, on peut choisir :

- de charger le bloc de la mémoire dans le cache puis effectuer l'opération d'écriture (écriture allouée)
- d'effectuer l'écriture directement dans la mémoire (écriture non allouée).

Une optimisation classique pour diminuer l'attente de la fin d'une écriture consiste à utiliser un tampon d'écriture, permettant au processeur de continuer à travailler dès que la donnée est écrite dans le tampon, sans attendre l'acquittement de la mémoire.

Politique d'écriture

7.10. 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 x Pénalité d'échec

Temps d'accès succès = temps d'accès à une donnée résidant dans le cache

Taux d'échec = nombre de défaut de cache / nombre d'accès cache ou = 1 – taux de succès

Taux de succès = nombre de succès / nombre d'accès cache

Exemple :

Lors de l'exécution d'une instruction, le processeur prend du temps pour la décoder, accéder aux données en mémoire nécessitées par cette instruction, et déclencher les opérations sur les données. Voici le cas suivant :

- durée d'un cycle horloge : T
- pénalité d'échec : 10 cycles
- durée d'une instruction (sans référence mémoire) : 2 cycles
- nombre de références mémoire par instruction : 1,33
- taux d'échec : 2%
- temps d'accès succès : négligeable
- temps d'exécution moyen d'une instruction = (2 + 1; 33 x 2% x10)T = 2; 27T
- et dans le cas où il n'y a pas de cache, e temps passe à : temps d'exécution moyen d'une instruction = (2 + 1; 33 x 10) = 15; 3T

Remarque :

- Cas de succès ⇒ **hit** ;
- Cas d'échec ⇒ **miss**

7.11. 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 du programme qui peut être exécuté dans un temps court:...
- – Elle stocke les données pour une utilisation temporaire

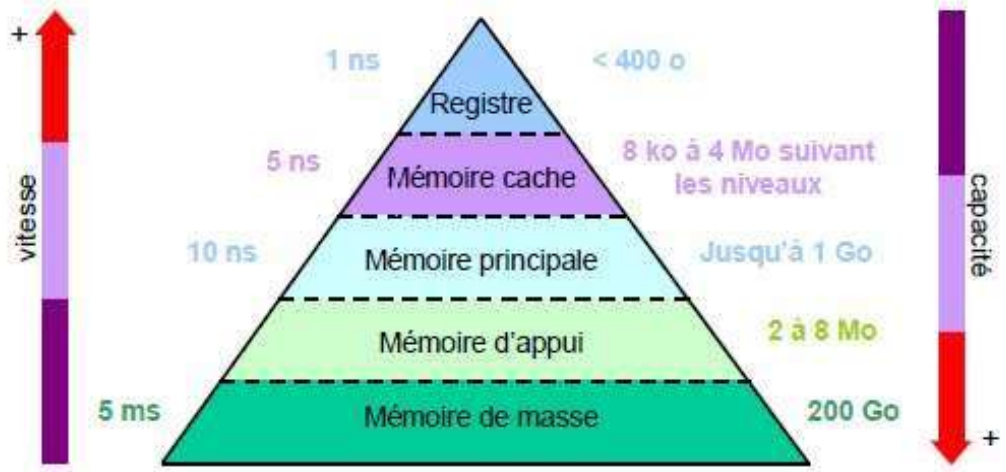
7.12. Inconvénients de la mémoire cache

- Elle a une capacité limitée
- Elle est très coûteuse

8. Hiérarchie des mémoires

Une mémoire idéale serait une mémoire de grande capacité, capable de stocker un maximum d'information et possédant un temps d'accès très faible afin de pouvoir travailler rapidement sur ces informations. Mais il se trouve que les mémoires de grande capacité sont souvent très lente et que les mémoires rapides sont très chères. Et pourtant, la vitesse d'accès à la mémoire conditionne dans une large mesure les performances d'un système.

En effet, c'est là que se trouve le goulot d'étranglement entre un microprocesseur capable de traiter des informations très rapidement et une mémoire beaucoup plus lente (ex : processeur actuel à 3Ghz et mémoire à 400MHz). Or, on n'a jamais besoin de toutes les informations au même moment. Afin d'obtenir le meilleur compromis coût-performance, on définit donc une hiérarchie mémoire. On utilise des mémoires de faible capacité mais très rapide pour stocker les informations dont le microprocesseur se sert le plus et on utilise des mémoires de capacité importante mais beaucoup plus lente pour stocker les informations dont le microprocesseur se sert le moins. Ainsi, plus on s'éloigne du microprocesseur et plus la capacité et le temps d'accès des mémoires vont augmenter.



Hiérarchie des mémoires

- **Les registres** sont les éléments de mémoire les plus rapides. Ils sont situés au niveau du processeur et servent au stockage des opérandes et des résultats intermédiaires.
- **La mémoire cache** est une mémoire rapide de faible capacité destinée à accélérer l'accès à la mémoire centrale en stockant les données les plus utilisées.
- **La mémoire principale** est l'organe principal de rangement des informations. Elle contient les programmes (instructions et données) et est plus lente que les deux mémoires précédentes.
- **La mémoire d'appui** sert de mémoire intermédiaire entre la mémoire centrale et les mémoires de masse. Elle joue le même rôle que la mémoire cache.
- **La mémoire de masse** est une mémoire périphérique de grande capacité utilisée pour le stockage permanent ou la sauvegarde des informations. Elle utilise pour cela des supports magnétiques (disque dur, ZIP) ou optiques (CDROM, DVDROM).

9. Conclusion

Nous avons présenté dans ce chapitre les principaux composants d'un ordinateur tels que : le processeur, l'UAL, les bus, les registres, la mémoire interne : mémoire RAM

(SRAM et DRAM), ROM, temps d'accès, latence,..., la mémoire cache : utilité et principe, algorithmes de gestion du cache (notions de base) ainsi que la hiérarchie des mémoires.

Le chapitre suivant est consacré à la présentation de quelques notions de base sur les instructions d'un ordinateur telles que celle de langage de haut niveau, assembleur, langage machine, les instructions machines usuelles (arithmétiques, logiques, de comparaison, chargement, rangement, transfert, sauts,…).

Nous allons découvrir, aussi, le principe de compilation et d'assemblage ainsi que le rôle de l'unité de contrôle et de commande et les principales hases d'exécution d'une instruction (recherche, décodage, exécution, rangement des résultats) avant de présenter le principe de l'UCC pipeline et ceux de l'horloge et du séquenceur.