Faculté des Sciences de Tétouan - 2022/2023
SMI – S4

Architecture des ordinateurs

Les Mémoires

Pr. Zaynab EL KHATTABI

Plan du cours

- ✓ Introduction
- ✓ Présentation de la mémoire
- ✓ Caractéristiques de la mémoire
- ✓ Types de la mémoire
- ✓ Hiérarchie des mémoires
- ✓ Mémoire centrale
- ✓ Mémoire rapide (registres)
- ✓ Mémoire cache

Introduction

- La mémoire est un dispositif capable de :
  - Recevoir les informations
  - Conserver les informations
  - Restituer les informations

- L’information peut être un programme ou des données

- On classe les mémoires selon:
  - Caractéristiques : capacité, débit ...
  - Type d’accès : séquentiel, direct ...

Introduction

- Les mémoires sont des composants destinés à stocker des **données binaires**. Elles sont organisées en éléments simples comprenant de 1 à n bits suivant le type de mémoire.

- Un élément de 8 bits est **un octet (Byte),** un élément de 16 bit **un mot (Word),** et un élément de 32 bits un **double mot (Double).**

- Chaque élément est rangé dans la mémoire à un emplacement appelé son **adresse.**
- L'organisation des éléments dans la mémoire peut être considéré comme **un tableau (array) de cellules (cell).** Chaque cellule correspond donc à une rangée de n bits.

- Les opérations affectant une mémoire sont donc :
  - l'adressage d'un élément,
  - la lecture ou l'écriture d'un élément.

Présentation de la mémoire

- **Unité de base :** bit (Le plus petit élément de stockage)
- **Octet (ou byte) :** groupe de 8 bits
- **Mot :** groupement d'octets (8, 16, 32, 64 ...) ,(Unité d'information adressable en mémoire)
- **Enregistrement :** bloc de donnée
- **Fichier :** ensemble d'enregistrements

Exemple: Une mémoire centrale de 1 Méga mots de 16 bits

✓ 1 x 1024 K-mots

✓ 1 x 1024 x1024 mots

✓ 1 x 1024 x1024 x 2 octets

✓ 1 x 1024 x1024 x 2 x8 bits

Caractéristiques des mémoires

- **Volatilité**
  - laps de temps durant lequel elle maintient les informations
  - Alimentation de ces mémoires
  - mémoire de travail de l'ordinateur, mémoire sur support magnétique
- **Temps d'accès**
  - temps pour accéder à l'information
  - de l'ordre de la nano-seconde pour les mémoires actuelles
  - de l'ordre de la milli-seconde pour les supports magnétiques
- **Capacité ou taille**
  - Nombre d'informations que peut contenir la mémoire
  - S'exprime en nombre de mots ou d'octets

Caractéristiques des mémoires

- **Temps de cycle mémoire**
  - Temps minimal entre 2 accès successifs à la mémoire
  - Temps de cycle= temps d'accès + temps supplémentaire (stabilisation des signaux, synchronisation,…) , car besoin d'opérations supplémentaires entre 2 accès.
- **Débit**
  - Nombre d'informations lues ou écrites par seconde
  - Exemple : 300 Mo/s
- **Type d’accès:**
  - Séquentiel
  - Directe

Caractéristiques des mémoires

Accès séquentiel

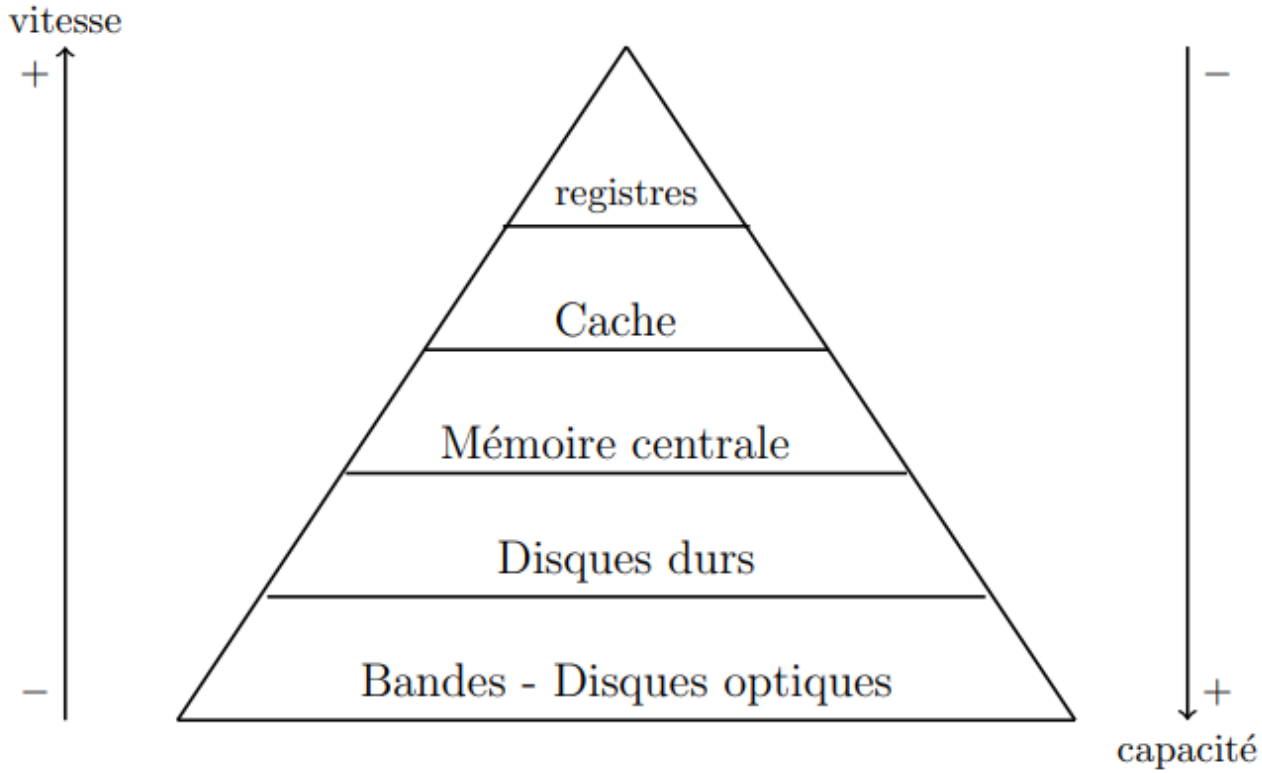
- Pour accéder à une information on doit parcourir toutes les informations précédentes
- Accès lent
- **Exemple :** bandes magnétiques

Accès direct

- Chaque information a une adresse propre
- On peut accéder directement à chaque adresse
- **Exemple :** mémoire centrale

Types de mémoire

- L'augmentation de la taille d'une mémoire s'accompagne toujours de l'augmentation du temps d'accès
- Mémoire d'accès plus vite = plus cher



Types de mémoire
Localisation

Les registres:

- Constituent la “mémoire de travail” du processeur.

La mémoire cache:

- Copie rapide de la mémoire centrale souvent décomposée en plusieurs parties, l’une collée sur le processeur et l’autre toute proche mais sur la carte mère.

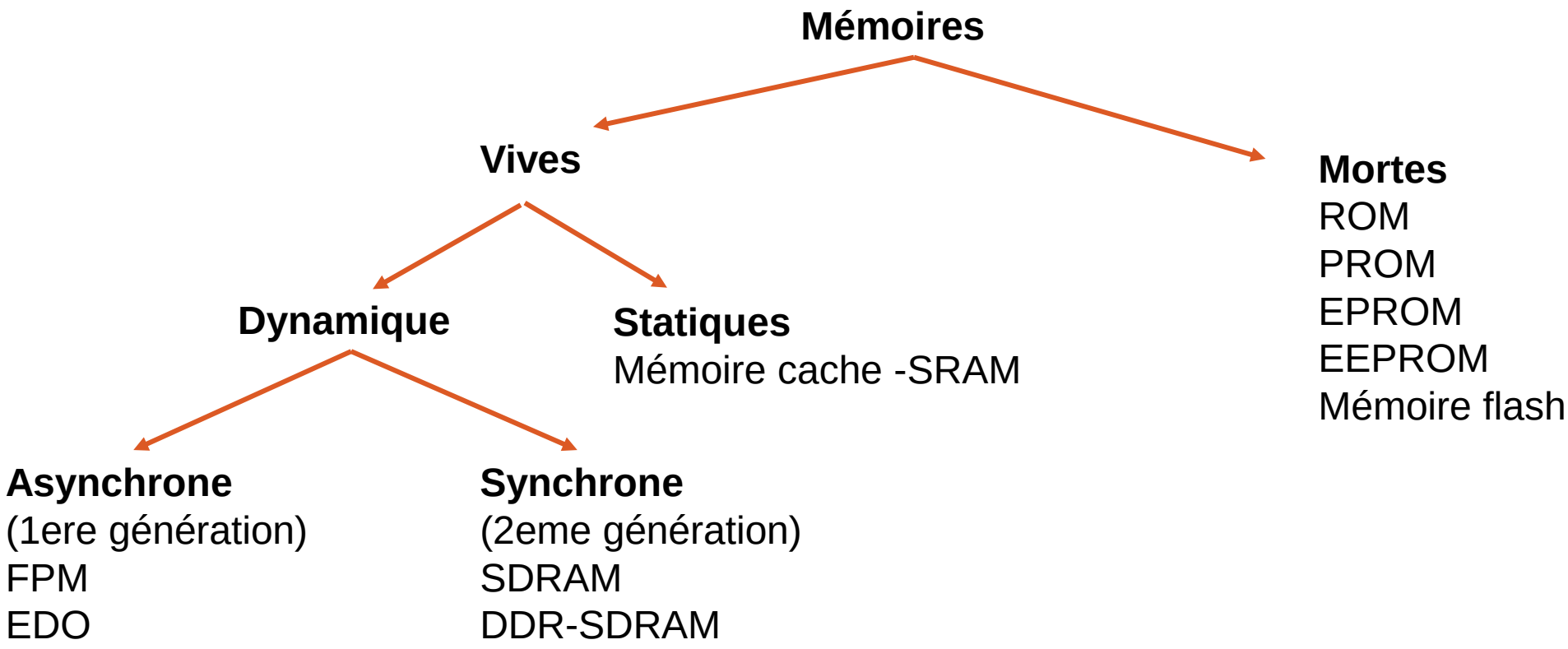
La mémoire centrale:

- Appelée mémoire vive ou RAM,(sur la carte mère). Elle stocke les données et les programmes.

La mémoire de masse :

- Appelée mémoire morte ou ROM, stockant les informations, généralement sous forme de fichiers. (hors de la carte mère)

Types de mémoire



Types de mémoire

Mémoire vive

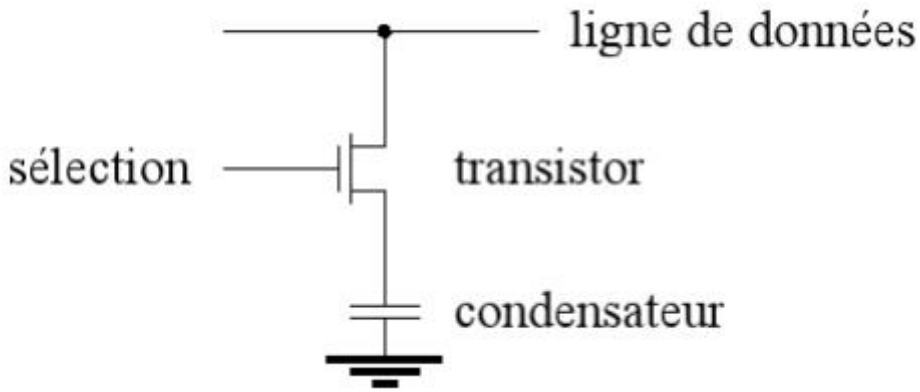
RAM (Random Access Memory: mémoire à accès aléatoire)

- Temps de cycle: très court
- Volatile (Leur contenu est modifiable et perte des informations hors alimentation électrique)
- Usage: stockage temporaire (programmes et données)
- Types:
  - DRAM (Dynamic RAM)
    - Asynchrome FPM(Fast Page Mode), EDO (Extended Data Out)
    - Synchrone SDRAM (Synchronous DRAM), DDR-SDRAM (Double Data Rate SDRAM)
  - SRAM (Static RAM)

Types de mémoire
Mémoire vive - DRAM

- Pour assurer une disponibilité des données et programmes en cours de traitement
  - => Réduction de temps d'accès

- La DRAM sert à stocker les données et programmes en cours de traitement avec un temps d'accès 10000 à 100000 fois plus petit que celui d'un disque dur.
- 1 bit = 1 transistor + 1 condensateur
- Le condensateur stocke l'information et doit être régulièrement rafraichi => augmentation du temps d'accès
- Usage : mémoire centrale (barrette RAM)



Types de mémoire
Mémoire vive – DRAM asynchrone

- Propre horloge différente de celle du bus de la carte mère
- Temps de cycle irrégulier

DRAM FPM: Fast Page Mode (1987)

- **Mode page:** Accès multiples sur une ligne dont l'adresse est fixée : faire varier les adresses des colonnes sur cette ligne
- Temps d'accès : 70 à 80 ns
- Fréquence : 25 à 33 Mhz

DRAM EDO: Extended Data Out (1995)

- Hyper FPM : démarrer un nouveau cycle d'accès au moment de la sortie de données d'un cycle
- Temps d'accès: 50 à 60 ns
- Fréquence: de 33 à 66 Mhz

Types de mémoire
Mémoire vive – DRAM synchrone

- Permet une lecture des données synchronisée avec le bus de la carte mère
- Temps de cycle régulier

SDRAM: Synchronous DRAM (1997)

- Temps d'accès: 10 ns
- Fréquence: 150 Mhz

DDR-SDRAM: Double Data Rate SDRAM (2000)

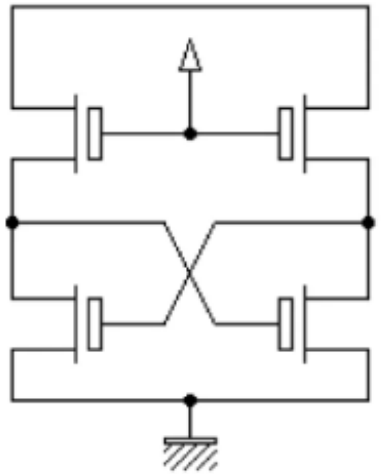
- **accès en rafale** : on charge ligne et colonne ainsi que le nombre de données à lire ; incrémentation dans la mémoire des colonnes pour les accès suivants (localité des données).
- Basée sur la technologie SDRAM, permettant de doubler le taux de transfert de la SDRAM à fréquence égale.

Types de mémoire
Mémoire vive – SRAM

Problématique:

- Le rafraichissement du condensateur de la mémoire DRAM ralenti le travail du processeur

=>Solution: Mémoire sans condensateur



- Mémoire qui ne nécessite pas de rafraichissement
- Le bit mémoire d'une SRAM est composé de 4 à 6 transistors.
- Réalisation : Bascule RS (ou D) qui stocke l'information
  - beaucoup plus rapide que la DRAM
  - beaucoup plus cher que la DRAM
- Usage: mémoire cache à l'intérieur ou à proximité du microprocesseur.

Types de mémoire
Mémoire morte

ROM (Read Only Memory): mémoire à lecture seule

- Non volatile (Leur contenu est fixe (ou presque ...) et conservé en permanence)
- Programmation: écriture des informations
- Usage: stockage permanent (programmes et données)
- Types:
  - **ROM** Programmable, une seule fois, par le constructeur
  - **PROM** (Programmable ROM) Programmable, une seule fois, par l'utilisateur
  - **EPROM** (Erasable Programmable ROM)
  - **EEPROM** (Electrically EPROM)
  - **Flash EPROM** variante d'EEPROM plus dense

Types de mémoire
Mémoire morte - ROM

- Programmée par le fabricant et son contenu ne peut plus être ni modifié, ni effacé par l'utilisateur.

Programmation:

- L'utilisateur doit fournir au constructeur un masque indiquant les emplacements des diodes dans la mémoire

Inconvénient:

- Modification impossible (toute erreur est fatale)

Usage:

- BIOS, instruction de démarrage

Types de mémoire
Mémoire morte - PROM

- Programmable une seule fois par l'utilisateur grace à un appareil appelé « programmeur de ROM »

Structure:

- Point mémoire: Fusible

Programmation :

- Placer des 1 en détruisant des fusible avec du courant
- Placer des 0 en déconnectant des jonctions

Inconvénient:

- Modification impossible (toute erreur est fatale)

Types de mémoire
Mémoire morte - EPROM

Mémoire reprogrammable et effaçable par ultraviolet.

- **Structure:**

Point mémoire : Transistor FAMOS

- **Programmation:**

Piéger des électrons dans la grille flottante en appliquant une forte tension entre grille et la source.

- **Effacement:**

Exposition d’une vingtaine de minutes à un rayonnement ultraviolet

- **Inconvénient:**

Impossible de sélectionner un groupe de cellules à effacer

Types de mémoire
Mémoire morte - EEPROM

- Programmable et effaçable électriquement

Structure:

- Point mémoire=Transistor SAMOS

Programmation:

- Piéger des électrons dans la grille flottante en appliquant une forte tension entre grille et la source.

Effacement:

- Libérer les électrons en appliquant une tension inverse

Avantage:

- Programmation et effacement mot par mot possible.

Types de mémoire
Mémoire morte - Flash EPROM

Variante d'EPROM

- **Structure:**
- Point mémoire: transistor NOR ou NAND
- **Programmation:**
- Piéger des électrons dans la grille flottante
- **Effacement:**
- Libérer les électrons dans la grille flottante
- **Usage:**
- Interne, porte-clés, cartouche amovible, disque SSD,…

Hiérarchie des mémoires

Problématique:

- Grande capacité= temps d'accès très élevé
- Très faible temps d'accès= cout très élevé

Remarque:

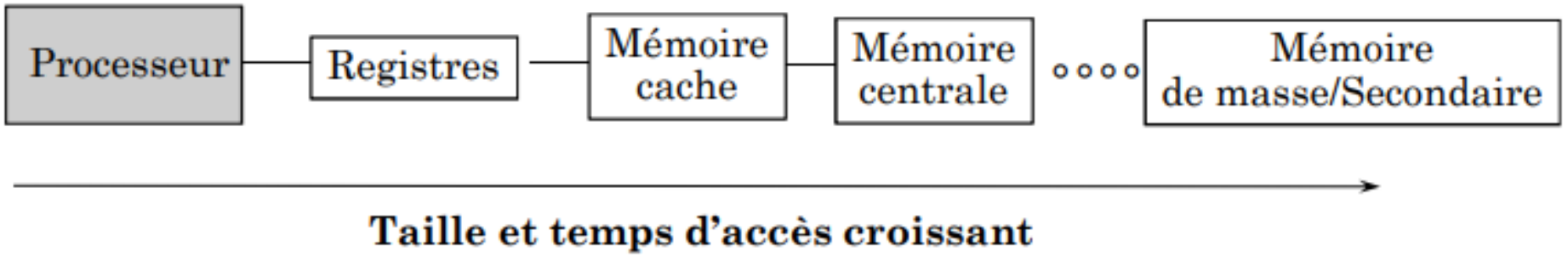
- Le processeur n'a jamais besoin de toutes les informations au même instant

Solution:

- Utiliser des mémoires de faible capacité et à très faible temps d'accès pour stocker les informations dont le µP se sert le moins.
- Ainsi, plus on s'éloigne du µP et plus la capacité et le temps d'accès des mémoires vont augmenter.

Hiérarchie des mémoires

- Les registres sont situés au niveau du processeur et servent au stockage des opérandes et des résultats intermédiares.
- La mémoire cache est destinée à accélérer l'accès à la mémoire DRAM en stockant les instructions et données les plus utilisées.
- La mémoire principale est l'organe principal de rangement des informations .elle contient les programmes et les données en cours de traitement.
- La mémoire de masse est une mémoire périphérique de grande capacité utilisée pour le stockage permanent ou la sauvegarde des programmes et des données



La mémoire centrale

- Elle mémorise les programmes utilisateurs, le système d'exploitation
- La mémoire centrale est située sur une carte mémoire
- Elle a une organisation standard
- Elle communique avec le processeur (carte mère) via des bus (de données, d'adresse)
- Représentation d'une variable en mémoire
  - Sur les intel, AMD, LITTLE ENDIAN
  - Sur les motorola BIG ENDIAN

La mémoire centrale

- Nécessaire d'implanter dans un microprocesseur
  - espace pour le système
  - espace pour l'utilisateur
  - espace pour les entrées /sorties

- Taille suffisante pour ranger les programmes
- Gestion sûre pour une intégrité des données
- Séparation entre les différents espaces (utilisateur, système)

La mémoire rapide : Registres

- Mémoire locale et privée du processeur
- Information temporaire
- Capacité limitée (1 à 128bits)
- Visibles ou invisibles (accessibles par l'utilisateur ou uniquement par le processeur).
- Exemple des processeurs Pentium :
  - **Registres Généraux sur 32 bits** : EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP
  - **Registre pointeur d'instruction (Le compteur de programme PC)** : EIP
  - **Registre de segment** : CS, DS, SS, ES, FS, GS
  - **Registre d'état** : flags, EFLAGS

La mémoire cache

- Mémoire située sur la carte mère => accès direct au processeur
- Idée principale du cache: diminuer le temps d'accès à certaines données en les ramenant de la mémoire centrale dans le cache.
- Gestion du cache se fait par microprogramme

But :

- éviter de rechercher en mémoire centrale des données déjà recherchées précédemment en les conservant près du processeur dans une mémoire à accès rapide (SRAM : 8 à 16 fois plus rapide que DRAM mais 4 à 8 fois plus volumineuse). Basée sur le principe de proximité (ex. : instructions des boucles)

La mémoire cache

Comparaison avec les autres mémoires:

Mémoire	**Vitesse**	**Capacité**
Mémoire cache	rapide	Petit (Ko)
Mémoire centrale	Moins rapide	Plus gros (Go/Mo)
Mémoire de masse	lent	Très gros (Mo/To)

La mémoire cache

Le cache interne ou de niveau 1 ou L1 :

- dans le processeur, unifié = contient instructions et données (ex. : Intel 486),
- Actuellement au moins 2 caches : 1 cache de données et 1 caches d'instructions
  - Exp (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.

Cache externe ou de niveau 2 ou L2 (unifié) :

- à côté du processeur, généralement de 256 ko.

- Chaque cache est géré par un contrôleur de cache.

La mémoire cache

Exemple de fonctionnement :

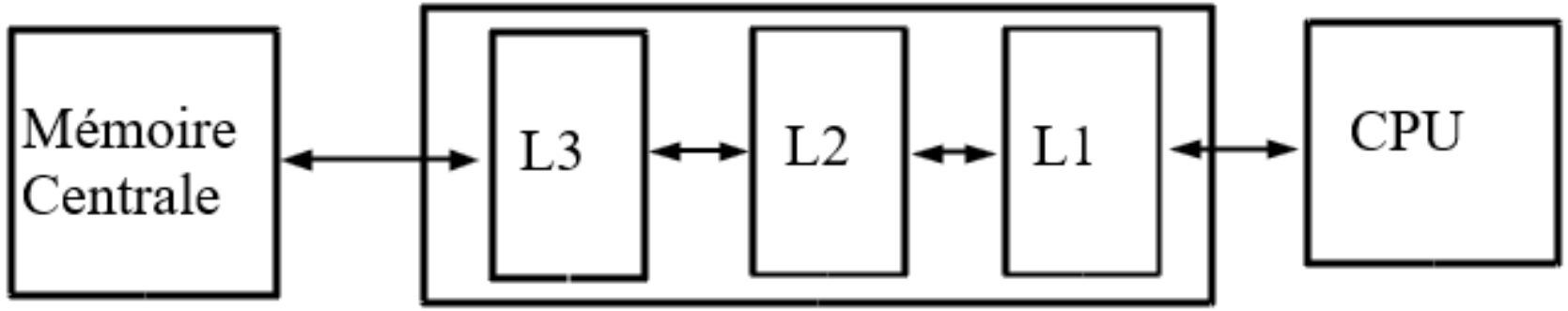
1. Le processeur demande une instruction présente dans une boucle d'un programme en plaçant l'adresse de cette instruction sur le bus d'adresse.
2. Le contrôleur de cache cherche si l'information est dans le cache en fonction de l'adresse → échec
3. Le contrôleur de cache entame un cycle de lecture en mémoire centrale.
4. La mémoire centrale envoie l'information au processeur et le contrôleur de cache la copie au passage dans le cache.

⇒ Au prochain accès à cette même instruction, le contrôleur de cache enverra directement la donnée au processeur sans démarrer un cycle de lecture en mémoire centrale.

La mémoire cache
Organisation en niveaux

Cache est organisé en plusieurs niveaux

- Niveaux L1 et L2, L3 pour certains processeurs
- Cache Li+1 joue le rôle de cache pour le niveau Li
- Cache Li+1 plus grand que Li mais moins rapide en temps d'accès aux données
- Cache L1 : généralement contient 2 parties
  - Instructions
  - Données



La mémoire cache

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

La mémoire cache
Performance

Evaluation de la performance d'une mémoire cache

- Taux de succès le plus élevé possible:
  - **Succès :** la donnée voulue par le CPU est dans le cache L1 : Ou la donnée voulue par le cache L1 est dans le cache L2 …
  - **Echec :** la donnée voulue n'y est pas =>Doit la charger à partir du niveau de cache suivant ou de la mémoire centrale

- A prendre en compte également
  - Temps d'accès à la mémoire: nombre de cycle d'horloges requis pour
    - Lire une donnée dans le cache en cas de succès
    - Aller la récupérer en cas d'échec au niveau supérieur

La mémoire cache
Performance

Choix de taille du cache et du nombre de niveaux

- Augmentation de la taille de la mémoire cache
  - Augmente le taux de succès
  - Mais ne gagne pas forcément en performance car augmentation du temps d'accès proportionnellement à la taille
- Augmentation du nombre de niveaux (plus de 3)
  - Pas de gain supplémentaire

Cache inclusif:

- Le cache L2 doit être bien plus grand que le cache de niveau L1, car sinon on stocke peu de données supplémentaires dans L2 par rapport à L1 et donc taux de succès de L2 sera faible!

La mémoire cache
Localité

- Une mémoire cache est structurée en lignes appelées **voies**.
- Chaque voie contient des données dont les adresses sont consécutives en mémoire centrale.
- La partie de l'adresse commune à ces données est appelée **étiquette**. Une voie est soit totalement inoccupée soit totalement pleine.

- **Motivation :** comme lire une donnée en mémoire est très coûteux en temps, à chaque fois qu'une donnée est lue, les données « voisines », c'est-à-dire qui seront stockées dans la même voie du cache sont aussi lues.
  - On espère que cette lecture groupée sera profitable. Cette espérance est basée **sur le principe de localité**.

La mémoire cache
Localité

Localité temporelle

- Une donnée référencée à **un temps t** aura de très fortes chances d'être référencée dans un futur proche.

Localité spatiale

- Si une donnée est référencée à **un temps t**, alors il y a de très fortes chances que les données voisines soient référencés dans un futur proche

Exemple:

for (i=0; i < N; i++) somme += A[i];

- Localité spatiale : A[i] (A[i+1], A[i+2]...)
- Localité temporelle : i,N,A

La mémoire cache
Localité

Pre-Fetching

- Chargement en avance des données dont le CPU devrait avoir besoin.
- Les algorithmes de pre-fetching sont basés sur les principes de localité.

- Localité temporelle
  - Garder en mémoire cache **les dernières données référencées** par le programme
- Localité spatiale
  - Charger **en avance les données/instructions contiguës** à une donnée/opération référencée

La mémoire cache
Accès par lignes mémoire

Pour le processeur

- La présence de la mémoire cache est transparente
- Le CPU demande toujours à accéder à une adresse en mémoire centrale (pour lecture ou écriture)

Adressage d'un mot mémoire:

- Adressage **direct** : adresse du mot en mémoire
- Adressage **via ligne** : index de la ligne et déplacement dans la ligne déplacement
- adresse mémoire = index × taille_ligne + déplacement

La mémoire cache
Interaction avec la mémoire centrale

Lecture/écriture en mémoire centrale via cache

- Le cache contient une copie d'une partie de la mémoire centrale
- Opérations de lecture/écriture sont effectuées dans le cache avec récupération sur la mémoire centrale
- **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 :
  - **Read Through :** la lecture se fait directement de la mémoire centrale vers le CPU.
  - **No Read Through :** la lecture se fait de la mémoire centrale vers le cache puis du cache vers le CPU

La mémoire cache
Interaction avec la mémoire centrale

Opération d'écriture:

- **Write Through** : l'information est écrite dans le cache et dans la mémoire centrale.
  - Avantages :
    - Un remplacement de l'information modifiée ne nécessite pas d'écriture en mémoire centrale et la lecture ayant provoqué le remplacement reste rapide ;
    - Facile à réaliser ;
  - La mémoire centrale contient toujours des informations cohérentes avec le cache.
  - Inconvénients :
    - l'écriture est lente
    - chaque écriture nécessite l'accès à la mémoire centrale.

La mémoire cache
Interaction avec la mémoire centrale

Opération d'écriture:

- **Write Back :** l'information est écrite uniquement dans le cache. Elle est écrite dans la mémoire centrale seulement lors d'un remplacement.
  - Avantage :
    - l'écriture se fait à la vitesse d'accès de la mémoire cache
    - plusieurs écritures sur une voie ne demandent qu'une écriture en mémoire centrale
  - Inconvénients :
    - implémentation matérielle compliquée
    - la mémoire centrale n'est pas toujours cohérente avec le cache

La mémoire cache
Interaction avec la mémoire centrale

Opération d'écriture:

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

- **Write Allocate :** l'information est d'abord chargée dans le cache puis modifiée.

- **No Write Allocate :** l'information est directement modifiée dans la mémoire centrale et n'est pas chargée dans le cache.

La mémoire cache
Correspondance lignes cache/mémoire

Trois méthodes pour gérer la correspondance entre une ligne dans le cache (voie) et une ligne de la mémoire centrale:

- **Correspondance directe**
- **Correspondance associative totale**
- **Correspondance associative par ensemble**

La mémoire cache
Correspondance lignes cache/mémoire

Correspondance directe:

- L lignes en cache
- La ligne d'adresse j en mémoire centrale sera gérée par la ligne i en cache avec
  - i = j mod L

Avantage:

- A partir de l'adresse d'une ligne en mémoire on sait directement dans quelle ligne du cache elle doit se trouver

Inconvénient:

- Devra parfois décharger et charger souvent les mêmes lignes alors que d'autres lignes sont peu accédées
- Peu efficace en pratique

La mémoire cache
Correspondance lignes cache/mémoire

Correspondance associative totale

- Une donnée peut être inscrite n’importe où dans le cache
- **Avantage:**
  - Très souple et efficace pour gérer les lignes de manière
  - Optimale en terme de succès d'accès
- **Inconvénient:**
  - Doit au pire parcourir toutes les lignes du cache pour savoir si la ligne cherchée s'y trouve ou pas

La mémoire cache
Correspondance lignes cache/mémoire

- **Correspondance associative par ensemble (N-way associative)**
- Il combine les deux techniques (c’est le plus couramment utilisé).
- On regroupe les lignes du cache **en ensembles de N Lignes**
- Une ligne de la mémoire centrale est gérée par un ensemble donné donc peut être une de ses N lignes
- Méthode utilisée en pratique
- Plus souple et efficace que la correspondance directe

La mémoire cache
Remplacement des données dans le cache

- Lorsqu'une donnée doit être écrite dans le cache et que les voies qu'elle peut occuper sont déjà prises, le contrôleur de cache doit choisir quelle voie remplacer.
- Une stratégie doit être mise en œuvre dans de tel cas.

Stratégies possibles :

- **Random :** pour changer uniformément les voies, celles-ci peuvent être remplacées de manière aléatoire.
  - Avantage : implémentation hard peu coûteuse.
  - Inconvénient : ignore le principe de localité.

La mémoire cache
Remplacement des données dans le cache

Least-Recently Used (LRU) :

- Pour réduire la probabilité de se débarrasser d'une information qui sera réutilisée prochainement, les données de la voie inutilisée depuis le plus longtemps sont remplacées.
  - Avantage :
    - tient compte du principe de localité.
  - Inconvénient :
    - quand le nombre de voies à gérer augmente, l'algorithme LRU devient de plus en plus
    - coûteux (difficile à implémenter, lent et souvent approximatif).

La mémoire cache
Remplacement des données dans le cache

Autres stratégies :

- **First-In First-Out (FIFO)**
- **Most-Recently Used (MRU)**
- **Least-Frequently Used (LFU)**
- **Most-Frequently Used (MFU)**