CHAP 4: MÉMOIRE CACHE

INTRODUCTION

La mémoire informatique est un dispositif électronique numérique qui sert à stocker les données .

La mémoire est une ressource importante qui doit être géré avec attention, elle présente une large gamme de type, technologies, performances, cependant aucune technologie n'est assez optimale pour satisfaire aux exigences d'une mémoire.

De ce fait le système informatique propose une hiérarchie de mémoire présentant les mémoires internes et externes .

Nous allons par la suite étudier les éléments internes de la mémoire et la mémoire cache .

I. La mémoire interne

La gestion d'une mémoire nécessite de classer les différentes mémoires en fonction de leur caractéristiques .

parmi les caractéristiques d'une mémoire , on distingue:

- **l'emplacement :** elle peut être interne (les processeurs, les registres, les mémoires caches) ou externe (les disques optiques, les disques magnétiques.)
- **la capacité:** c'est le nombre total de bits que contient la mémoire. Elle s'exprime souvent en octet ou en bit .
- **Unité de transfert:** elle peut être par mot ou par bloc.

Le type d'accès au mémoire: on distingue ici plusieurs types d'accès à savoir:

- **Accès séquentielle**: c'est l'accès le plus lent pour accéder à une information particulière direct, aléatoire, ou associatif.
- **Accès direct**: les informations ont une adresse propre, et sont donc directement accessibles (par exemple la mémoire centrale, les registres)
- **Accès sémi séquentielle**: c est une combinaison 'est une combinaison

des accès direct et séquentiel (dans un disque magnétique, l'accès au cylindre est direct et l'accès au secteur est séquentiel).

❖ **La performance:** elle se mesure suivant 3 critères:

- **le temps de cycle**: il représente l'intervalle minimum entre deux accès successif à la mémoire .
- **le temps d'accès** qui désigne le temps qui s 'écoule entre une demande de lecture ou d'écriture et son accomplissement .
- **Le taux de transfert des données .**

❖ **le dispositif de stockage**: il permettent de stocker les informations à long terme dans la mémoire de masse. Le dispositif de stockage peut etre:

- Magnétique tel que le disque dur
- Optique tel que CD ROM , DVD ROM,
- Semi conducteur :
- Magnétique et optique.

❖ **Les caractéristiques physiques :**

- volatiles et non volatiles: il caractérise la permanence des informations dans un mémoire
- Effacables et non effacables.:

1. L’emplacement de la mémoire

La mémoire a pour rôle principal le stockage et la restitution des données . Dans un système informatique , une mémoire peut être interne à l’ordinateur , on parle dans ce cas de mémoire principale (des registres, RAM) ou externe à l’ordinateur (disque dur, cd)
La mémoire interne désigne un composant du pc qui permet d’accéder aux données pendant une période de temps limitée, c’est une mémoire volatile qui stocke temporairement les données. Elle est utilisé pour stocker des informations que votre ordinateur est entrain d’utiliser afin qu’elles puissent etre consultées rapidement , de même que de nombreuses taches du quotidien tel que charger des applications, naviguer sur internet…..
La mémoire externe par contre est encore appelé mémoire externe ou mémoire de masse stocke les informations à long terme même après l’arret de l’ordinateur.

2. La capacité d’une mémoire:

il désigne le nombre d’information qu’on peut enregistrer sur cette mémoire.
La capacité d’une mémoire interne est exprimée sous forme d’octet ou mots (1mot = 8 bits)variant de 8 à 16 ou bits. De même pour la mémoire externe.

3. Unité de transfert

Il désigne le nombre de lignes électriques entrant ou sortant du module de mémoire , il doit être égale à la taille du mot mais le plus souvent , il est plus grand à savoir 64,128 ou 256 bits . 3 concepts permettent de mieux comprendre cette notion à savoir:

- **Le mot:** c'est l'unité naturel d'organisation de la mémoire, la taille d'un mot est égale au même nombre de bit utilisé pour représenter un entier à quelques exception prêt.
- Unités adressables: dan certains cas il représente le mot mais dans d'autres cas, l'adressage se fait au niveau de l'octet. Dans tous les cas le rapport entre la longueur en bit d'une adresse nommée L et le nombre N d'unité adressables est : $2A = N$.
- **Unité de transfert:** c'est le nombre de bits lu ou écrit dans la mémoire à un instant donné , il n'est pas forcement égale au mot ou à une unité adressable. Au niveau de la mémoire externe, les unités sont transférées dans les unités beaucoup plus grande qu'un mot appelées blocs.

4. Les types d'accès aux mémoires

- Accès séquentiel : c'est l'accès le plus lent ; pour accéder à une information particulière on est obligé de parcourir toutes celles qui la précèdent (exemple
les bandes magnétiques)

- **Accès direct** : les informations ont une adresse propre, et sont donc directement accessibles (par exemple la mémoire centrale, les registres)

- **Accès semi-séquentiel** : c est une combinaison 'est une combinaison
des accès direct et séquentiel (dans un disque magnétique, l'accès au cylindre est direct et l'accès au secteur est séquentiel)

- **Accès aléatoire** : le temps d’accès aux informations ici est indépendant de la séquence des accès précédents et est constant . Ainsi n’importe quel emplacement peut etre sélectionné au hazard et directement accessible. On retrouve ce type d’accès dans la mémoire principale de certain system de cache.

5- la performance

La performance d'une mémoire se mesure suivant le temps d'accès , le temps de cycle et le taux de transfert des données **:**

- ✓ **Le temps d'accès:** C'est le temps nécessaire pour effectuer une opération de lecture ou d'écriture. Par exemple pour l'opération de lecture , le temps d'accès est le temps qui sépare la demande de la lecture de la disponibilité de l'information.
- ✓ **Le temps de cycle:** il est constitué du temps d'accès plus le temps supplémentaire requis avant que le prochain temps d'accès ne recommence. Le temps de cycle ne concerne que le système de bus
- ✓ **taux de transfert :** il s'agit du taux auquel les données peuvent etre transferées vers une unité de mémoire . Pour la mémoire vive, le taux de transfert est égale à 1.

6- Technologie utilisé

Suivant la technologie utilisé , les mémoires peuvent etre classés en 3 catégories:

– **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,..)

7- **variétés physiques.**

- **La volatilité:**

Si une mémoires perd sont contenu (les informations) lorsque la sources d'alimentation est coupée alors la mémoire est dite volatile.

Si une mémoire ne perd pas (conserve) sont contenu lorsque la sources d'alimentation est coupée alors la mémoire est dite non volatile (mémoire permanente ou stable)

- **Mode d'accès lecture écriture:**

Sur une mémoire on peut effectuer l'opération de :

• lecture : récupérer / restituer une information à partir de la mémoire.

• écriture : enregistrer une nouvelle information ou modifier une information déjà existante dans la mémoire .

• Il existe des mémoires qui offrent les deux modes

- lecteur/écriture , ces mémoire s'appelles mémoires vives.

• Il existent des mémoires qui offrent uniquement la possibilité de la lecture (c'est pas possible de modifier le contenu). Ces mémoires s'appelles mémoires mortes.

8- Hiérarchie de mémoire:

Au vue de tous ces caractéristiques des mémoires, il parait très compliqué de trouvé un système de mémoire précis susceptible de correspondre à un système informatique pratique. D'où la nécessité d'utiliser une hiérarchie de mémoire .

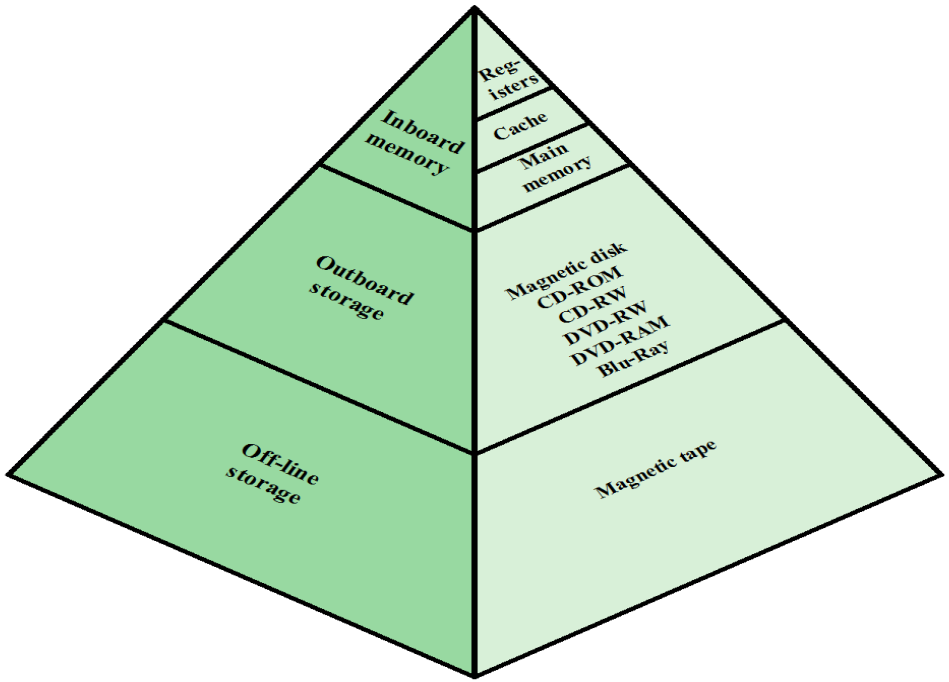

Figure 4.1 The Memory Hierarchy

Selon le schéma précèdent, plus on descend de la mémoire, on assiste à :
- une diminution du cout par bit,
- une augmentation de la capacité, du temps d'accès,
- une diminution de la fréquence d'accès de la mémoire.

Ainsi, des mémoires plus petites, plus chères et plus rapides sont complétées par des mémoires plus grandes et plus lentes .
afin de mieux optimiser l'hierarchie de mémoire, nous devons diminuer la fréquence d'accès.

Nous allons par la suite , mieux expliquer ce phénomène en étudiant la mémoire cache .

II- Mémoire cache

1. Notion de cache

Un cache est une mémoire intermédiaire dans laquelle se trouvent stockées toutes les informations qu'un élément demandeur est le plus susceptible de demander. Un cache sert donc à accélérer la communication entre un élément fournisseur (disque dur par exemple) plus lent que l élément demandeur (processeur par 'élément demandeur (processeur par exemple). Comme ces informations sont immédiatement disponibles, le temps de traitement se diminué d'autant, ce qui mécaniquement accroît notablement les performances de l'ordinateur.

il existe souvent plusieurs niveaux de mémoire cache : une interne au processeur, une autre intégrée sur la carte mère, mais on peut en avoir aussi sur le disque dur.

2. Mémoire cache

la Mémoire cache est la traduction littérale de l'expression anglaise cache memory qui vient elle même de mémoire cachée, principe inventé à Grenoble dans les années 1960, l'académie française propose antémémoire.

La différence entre **mémoire cache et mémoire tampon(buffer**) réside dans le fait que la mémoire cache duplique l'information, tandis que le tampon exprime l'idée d'une salle d'attente, sans impliquer nécessairement une duplication.
Le cache buffer (p) q tampon de cache) du disque dur ou disk cache (cache de disque) est à la fois un tampon où transite l'information et une mémoire cache qui recopie sous forme électronique les données stockées dans le disque sous forme magnétique.

3. Fonctionnement du cache

Le cache contient une copie des données originelles Lorsqu'elles sont coûteuses (en terme de temps d'accès) à récupérer ou à calculer par rapport au temps d'accès au cache.
Une fois les données stockées dans le cache, l'utilisation future de ces données peut être réalisée en accédant à la copie en cache plutôt qu'en récupérant ou recalculant les données, ce qui abaisse le temps d'accès moyen.

Le processus fonctionne ainsi :

- L'élément demandeur (microprocesseur) demande une information
- Le cache vérifie s'il possède cette information. S'il la possède, il la retransmet à l'élément demandeur; on parle alors de succès de cache. S'il ne la possède pas il la demande à l'élément fournisseur (mémoire principale); on parle alors de défaut de cache
- L'élément fournisseur traite la demande et renvoie la réponse au cache ;
- Le cache la stocke pour utilisation ultérieure et la retransmet à l'élément demandeur.

Si les mémoires cache permettent d'accroître les performances, c'est en partie grâce à deux principes qui ont été découverts suite à des études sur le comportement des programmes informatiques :

- **Le principe de localité spatiale** : qui indique que l accès à une 'accès à une instruction située à une adresse X va probablement être suivi d'un accès à une zone tout proche de X.
- **Le principe de 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.

On trouve une zone de cache :

- **cache de premier niveau (L1)** dans les processeurs (cache de données souvent séparé du cache d'instructions)
- **Cache de second niveau (L2)** dans certains processeurs (peut se situer hors de la puce) ;
- **Cache de troisième niveau (L3)** rarement présent sur les Intel Core i7), Dans les disques durs, Dans les serveurs proxy.

4- mémoire cache des microprocesseurs

Elle est souvent subdivisée en niveaux qui peuvent aller jusqu'à trois . Elle est très rapide, et donc très chère. Il s'agit souvent de SRAM (Static Random Access memory).

En programmation, la taille de la mémoire cache revêt un attrait tout particulier, car pour profiter de l'accélération fournie par cette mémoire très rapide, il faut que les parties de programme tiennent le plus possible dans cette mémoire cache. Comme elle varie suivant les processeurs, ce rôle d'optimisation est souvent dédié au compilateur. De ce fait, plus la taille de la mémoire cache est grande, plus la taille des programmes accélérés peut être élevée.

C'est aussi un élément souvent utilisé par les constructeurs pour faire varier les performances d'un produit sans changer d'autres matériels. Par exemple, pour les microprocesseurs, on trouve des séries bridées (avec une taille de mémoire cache volontairement réduite) tels que les Duron chez AMD ou Celeron chez Intel, et des séries haut de gamme avec une grande mémoire cache comme les processeurs Opteron chez AMD, ou Pentium EE ou Core i7 chez Intel.

Définitions

❖ **Une ligne** est le plus petit élément de données qui peut être transféré entre la mémoire cache et la mémoire de niveau supérieur.

❖ **Un mot** est le plus petit élément de données qui peut être transféré entre le processeur et la mémoire cache.

5- Défauts de cache

Il existe trois types de défauts de cache en système mono processeur et quatre dans les environnements multiprocesseurs :

- **les défauts de cache obligatoires :** ils correspondent à la première demande du processeur pour une donnée/instruction spécifique et ne peuvent être évités,
- **les défauts de cache capacitifs** : l'ensemble des données nécessaires au programme excèdent la taille du cache, qui ne peut donc pas contenir toutes les données nécessaires.
- **les défauts de cache conflictuels** : deux adresses distinctes de la mémoire de niveau supérieur sont enregistrés au même endroit dans le cache et s'évincent mutuellement, créant ainsi des défauts de cache,
- **les défauts de cache de cohérence** : ils sont dus à l'invalidation de lignes de la mémoire cache afin de conserver la cohérence entre les différents caches des processeurs d'un système multiprocesseurs

5- Le mapping

La mémoire cache ne pouvant contenir toute la mémoire principale il faut définir une méthode , il faut définir une méthode indiquant à quelle adresse de la mémoire cache doit être écrite une ligne de la mémoire principale. Cette méthode s'appelle le mapping.

Il existe trois types de mapping :

- **La mémoire cache complètement associative** (fully associative cache),
- **La mémoire cache directe** (direct mapped cache),
- **La mémoire cache N-associative** (N-way set associative cache)

- **Mémoire cache complètement associative**

Chaque ligne de la mémoire de niveau supérieur peut être écrite à n'importe quelle adresse de la mémoire cache. Cette méthode requiert beaucoup de logique car elle donne accès à de nombreuses possibilités. Ceci explique pourquoi l'associativité complète n'est utilisée que dans les mémoires cache de petite taille. Cela donne le format suivant de l'adresse :

Tag = n° de la ligne mémoire enregistrée .

Offset = n° du mot dans la ligne.

❖ Mémoire cache directe

Chaque ligne de la mémoire principale ne peut être enregistrée qu’ à une seule adresse de la mémoire cache. Ceci crée de nombreux défauts de cache conflictuels si le programme accède à des données qui sont mappées sur les mêmes adresses de la mémoire cache. La sélection de la ligne où la donnée sera enregistrée est habituellement obtenue par: *Ligne* = *Adresse* mod *Nombre de lignes*. Tag Index Offset

Une ligne de cache est partagée par de nombreuses adresses de la mémoire de niveau supérieur. Il nous faut donc un moyen de savoir quelle donnée est actuellement dans le cache. Cette information est donnée par le *tag*, qui est une information stockée dans le cache. L'index correspond à la ligne où est enregistrée la donnée. En outre, le contrôleur de la mémoire cache doit savoir si une ligne contient une donnée ou non. Un bit additionnel (appelé bit de validité) indique si la ligne est libre ou non.

❖ Mémoire cache N-associative

Il s'agit d'un compromis entre le mapping direct et complètement associatif essayant d'allier la simplicité de l'un et l'efficacité de l'autre. La mémoire cache estdivisée en ensembles (sets) de N lignes de cache.

Une ligne de la mémoire de niveau supérieur est affectée à un ensemble, elle peut par conséquent être écrite dans n'importe laquelle des voies. Ceci permet d'éviter de nombreux défauts de cache conflictuels.

À l'intérieur d'un ensemble, le mapping est complètement associatif. En général, la sélection de l'ensemble est effectuée par:

Ensemble = Adresse mémoire mod (Nombre d'ensembles).

❖ **Caches unifiés ou caches séparés**

Pour fonctionner, un processeur a besoin de données et d'instructions. Il existe donc deux solutions pour l'implémentation des mémoires cache:

- **le cache unifié** : données et instructions sont enregistrées dans la même mémoire cache,
- **les caches séparés de données et d'instructions**: Séparer données et instructions permet notamment d'augmenter la fréquence de fonctionnement du processeur, qui peut ainsi accéder simultanément à une donnée et une instruction.

6- Politique d'écriture dans la mémoire de niveau supérieur

Quand une donnée/instruction se situe dans le cache, le système en possède deux copies: une dans la mémoire de niveau supérieur et une dans la mémoire cache Deux différentes politiques s'affrontent:

- **write through:** la donnée/instruction est écrite à la fois dans le cache et dans la mémoire de niveau supérieur. La valeur de la mémoire principale est constamment cohérente entre le cache et la mémoire de niveau supérieur, simplifiant ainsi de

nombreux protocoles de cohérence,

- **write back**: l'information n'est écrite dans la mémoire de niveau supérieur que lorsque la ligne disparaît du cache (invalidée par d autres processeurs 'autres processeurs évincée pour écrire une , évincée pour écrire une autre ligne...). Cette technique est la plus répandue car elle permet d'éviter de nombreuses écritures mémoires inutiles.

Cependant, afin de ne pas écrire des informations qui n'ont pas été modifiées chaque ligne de la mémoire cache est pourvue d'un bit indiquant si elle a été modifiée.

7- **Algorithmes de remplacement des lignes de cache:**

Les caches associatifs de N voies et complètement associatifs impliquent le mapping de différentes lignes de la mémoire de niveau supérieur sur le même ensemble. Ainsi, il faut désigner la ligne qui sera effacée au profit de la ligne nouvellement écrite. Le but de L'algorithme de remplacement des lignes de cache est de choisir cette ligne de manière optimale

Les algorithmes de remplacement des lignes de cache les plus répandus sont :
aléatoire pour sa simplicité de création de l'algorithme FIFO(First In First Out) pour sa simplicité de conception, LRU (Least Recently Used) qui mémorise la liste des derniers éléments accédés, FINUFO - First In Not Used, First Out
(algorithme de l'horloge ou Clock) = approximation du LRU.