# Généralisation

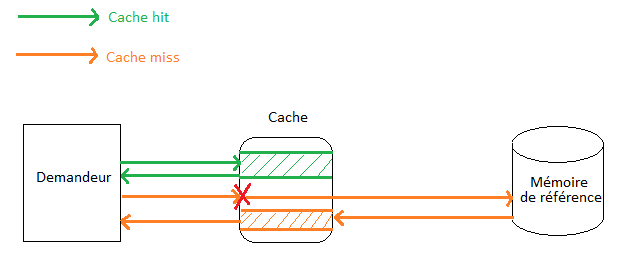
De manière générale, la mémoire cache (ou l’antémémoire en Français), permet d’économiser et de réduire le temps d’accès aux ressources. Par exemple le CPU qui souhaite accéder aux données contenues en mémoire centrale.

# Principe de fonctionnement

L’antémémoire se positionne entre un élément « demandeur » et une mémoire de références contenant les ressources (données). Elle est de très petite capacité et est très onéreuse. Par définition l’accès à l’antémémoire est beaucoup plus rapide.

Lorsque le demandeur à besoin d’accéder à une ressource en écriture ou en lecture, le principe de fonctionnement est le suivant :

1. L’antémémoire vérifie qu’elle ne contient pas une copie de la ressource réclamée. Si la ressource est contenue, l’antémémoire la fournie directement au demandeur. On parle alors de ***succès de cache*** (Cache hit en anglais) (voir flèche verte sur le schéma plus bas).  
   En revanche si la ressource n’y est pas, on parle alors de ***défaut de cache*** (Cache miss) (voir flèche orange sur le schéma plus bas). Le cas échéant, on procède aux étapes suivantes…
2. La demande se propage au niveau de mémoire supérieur. Cette dernière renvoie la ressource souhaitée.
3. L’antémémoire stocke la ressource demandée pour une utilisation ultérieur puis la retransmet à l’élément demandeur.



Ce qui détermine la performance d’une antémémoire est ce que l’on appelle le taux de succès (Hit ratio) qui est le nombre de succès de cache par le nombre d’accès mémoire.  
Il est évident que plus celui-ci est élevé, plus efficace sera l’antémémoire et plus l’on aura économisé du temps d’accès aux ressources.

# Les types de cache

On distingue deux types de mémoire cache. Le cache matériel essentiellement (ou totalement) implémentée de manière électronique, ou bien le cache logiciel, implémenté directement par un programme informatique.

## Les caches matériels

Une mémoire cache matériel est très performante mais aussi très onéreuse. Un exemple de cache matériel très utilisé est la SRAM (Static Random Access Memory). La SRAM est un type de mémoire vive utilisant des bascules pour mémoriser l’information.

Une SRAM de 1999 :



## Les caches logiciels

L’un des caches logiciel le plus connus est le cache implémenté par les navigateurs WEB. Lorsque qu’une page web est chargée, il est fréquent que celle-ci requière un certain nombre de fichiers annexes (images, javascript, css, etc.). Afin de limiter le nombre de requête et d’améliorer la vitesse de chargement des pages ultérieurement, les fichiers annexes sont stockés sur la mémoire de masse de l’ordinateur et seront réutilisé à la demande.

Un autre exemple concerne le SGF d’un système d’exploitation. Lorsqu’un fichier est ouvert pour lecture et/ou écriture, celui-ci est en réalité mis dans un cache logiciel sur la mémoire centrale afin de rendre son accès plus rapide.

# Problématiques d’implémentation

## Placement de l’information

Le choix de la façon d’associer le contenue du cache avec le fournisseur est primordiale, car les performances en dépendent.

Pour l’heure il existe 4 types de caches :

* Directement adressés (direct mapped)
* Totalement associatifs (fully associative)
* Associatifs par voie (N-Way associative)
* Pseudo-associatifs

### Cache directement adressé

## Politique de remplacement

Par définition une antémémoire est de petite taille. Par conséquent il va finir par arriver une situation où lors d’un cache miss il n’y ai plus de place pour accueillir la nouvelle ressource. L’antémémoire devra se débarrasser d’une des ressources préalablement mises en cache afin de la remplacer.

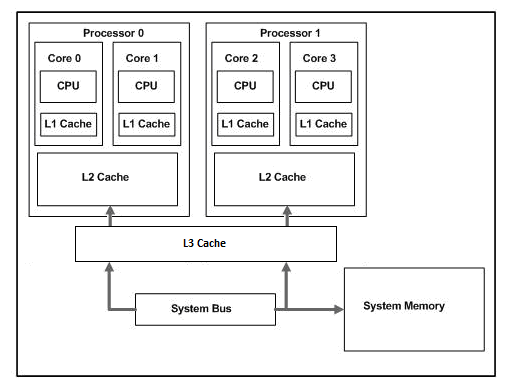
En fonction du contexte d’utilisation du cache le choix va se porter sur un algorithme ou un autre.

Les algorithmes sont similaires à ceux utilisé par le système d’exploitation pour gérer la pagination de la mémoire centrale.

## Gestion des mises à jour des données

## Dimensionnement

## Hiérarchie des caches



# Références

<https://fr.wikipedia.org/wiki/Static_Random_Access_Memory>

<https://fr.wikipedia.org/wiki/M%C3%A9moire_cache>

<https://fr.wikibooks.org/wiki/Fonctionnement_d%27un_ordinateur/Les_m%C3%A9moires_cache#On_n%E2%80%99a_pas_qu%E2%80%99un_seul_cache_!>