

ARCHITECTURE DES ORDINATEURS

2^{ème} Année Informatique

Chapitre2:
Principaux composants d'un ordinateur
Partie 2

Centre universitaire Mila
2022-2023

1

La mémoire Cache

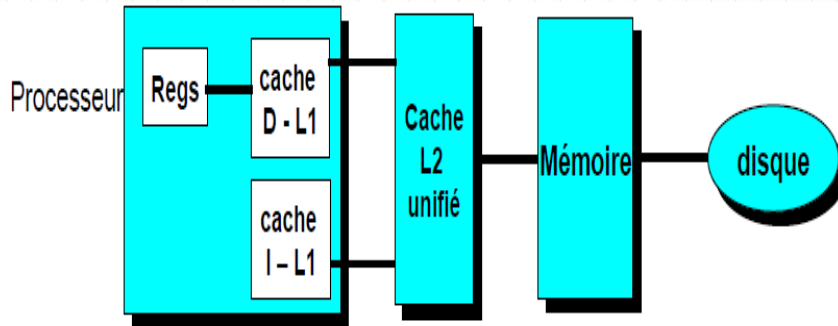
Définition

- **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 bloc** 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.

2

niveaux de cache

- Options: caches instructions et données séparés ou cache unifié



Plus gros, plus lent, moins cher

3

La mémoire Cache

➤ Idée générale

- Réduire le temps d'accès aux données
- Accélérer les traitements en ne bloquant pas le processeur

➤ Principe de localité

▪ localité temporelle

Une information demandée a de fortes chance d'être à nouveau demandée dans un futur proche

▪ localité spatiale

Les autres mots du bloc demandé ont de fortes chances d'être demandés dans un futur proche

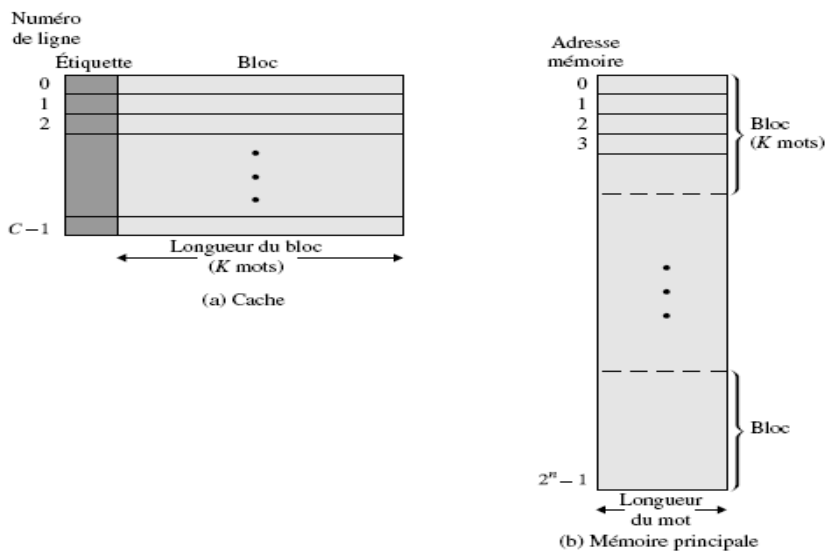
4

La mémoire Cache

- Le CPU fait référence à un mot dans le cache
- Recherche de ce mot dans le cache
 - ✓ **Présence**
 - ❖ Succès de cache
 - ❖ lecture
 - ✓ **Absence**
 - ❖ défaut de cache
 - ❖ copie depuis la mémoire principale vers le cache du bloc contenant le mot
 - ❖ nouvel accès au cache en lecture
- Problème
Où placer cette nouvelle information dans le cache ?

5

Localisation dans le cache



6

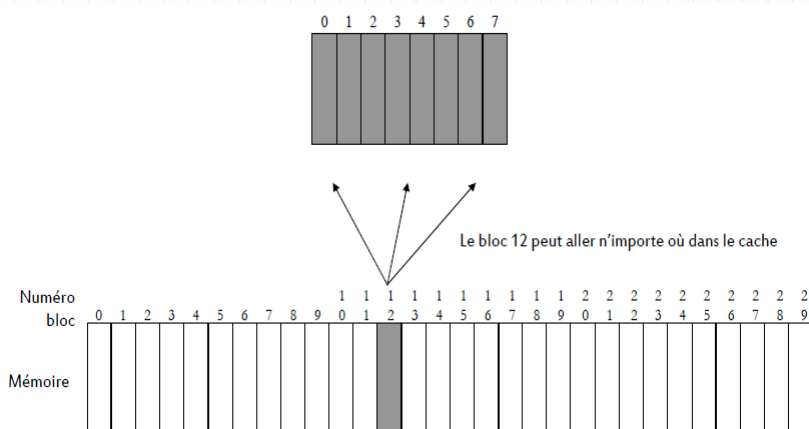
Localisation dans le cache

- Différentes stratégies pour placer l'information dans le cache
 - Correspondance directe
 - Totalement associatif
 - Associatif par ensemble
- Une donnée présente dans le niveau n-1 (plus près du processeur) est présente dans le niveau n

7

Cache totalement associatif

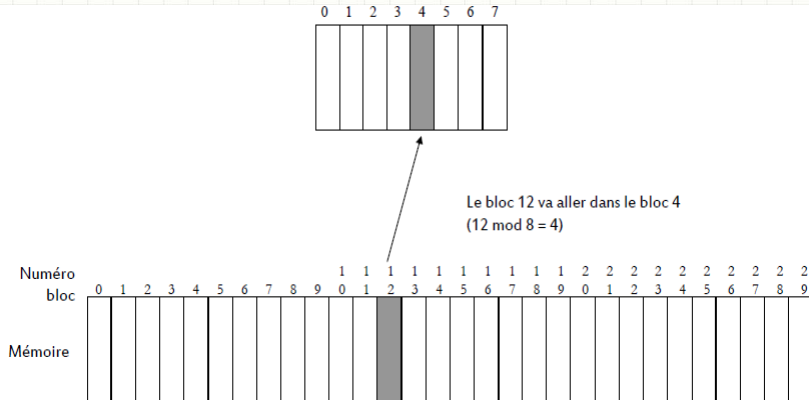
- Le bloc peut aller n'importe où dans le cache



8

Cache à correspondance directe

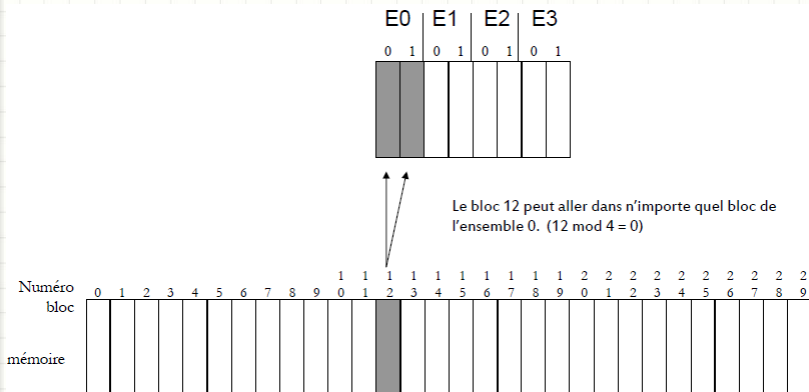
- Le bloc a un seul emplacement possible dans le cache
- $(\text{numéro du bloc}) \bmod (\text{nombre de blocs du cache})$



9

Cache associatif par ensemble

- $(\text{numéro du bloc}) \bmod (\text{nb d'ensembles du cache})$



10

Localisation dans le cache

- Décomposition d'une adresse dans le cas d'un cache
- Associatif par ensemble
- A correspondance directe

Numéro de bloc		Déplacement dans le bloc
Etiquette	Index	

- L'augmentation de l'associativité diminue la taille de l'index
- Les caches totalement associatifs n'ont pas de champ index

11

Remplacement dans le cache

Remplacement d'un bloc occupé

- Correspondance directe
 - Pas d'alternative
- Associativité totale ou par ensemble
 - Génération d'un nombre aléatoire
 - LRU (Least Recently Used)
enregistre l'ordre d'accès
 - LFU (Least Frequently Used)
garde trace de la fréquence d'accès
 - FIFO (First In First Out)

12

Cohérence du cache

- Les lectures dominent les accès mémoire
- Deux politiques d'écriture existent :
 - **L'écriture simultanée (write through)** dans laquelle l'information est écrite
 - dans le cache
 - En mémoire principale
 - en parallèle pour chaque modification
 - **La réécriture ou recopie (write back)** dans laquelle l'information est écrite uniquement le cache.
 - Le cache modifié est recopié en mémoire principale uniquement lorsqu'il est remplacé.

13

Les défauts de caches

- **Défauts obligatoires** de chargement (défaut de démarrage à froid). Un bloc accédé pour la première fois n'est pas dans le cache.
- **Défauts de capacité**. Si le cache ne peut contenir tous les blocs nécessaires au cours de l'exécution d'un programme
- **Défauts de conflits** (défaut de collision). Si la stratégie de placement de bloc est associative par ensembles de blocs ou à correspondance directe, des défauts de conflit surviendront, car un bloc peut être rejeté puis récupéré si trop de blocs sont en correspondance avec le même ensemble.

14