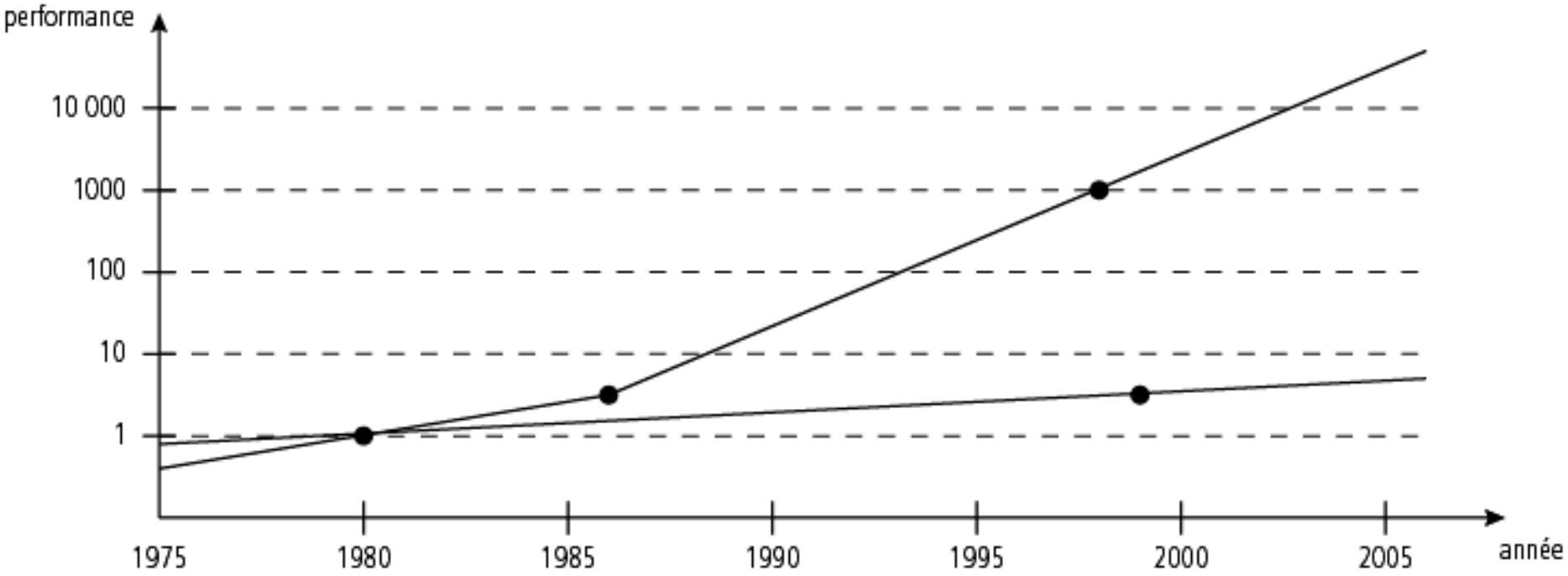
Architecture des ordinateurs

62 - La mémoire cache

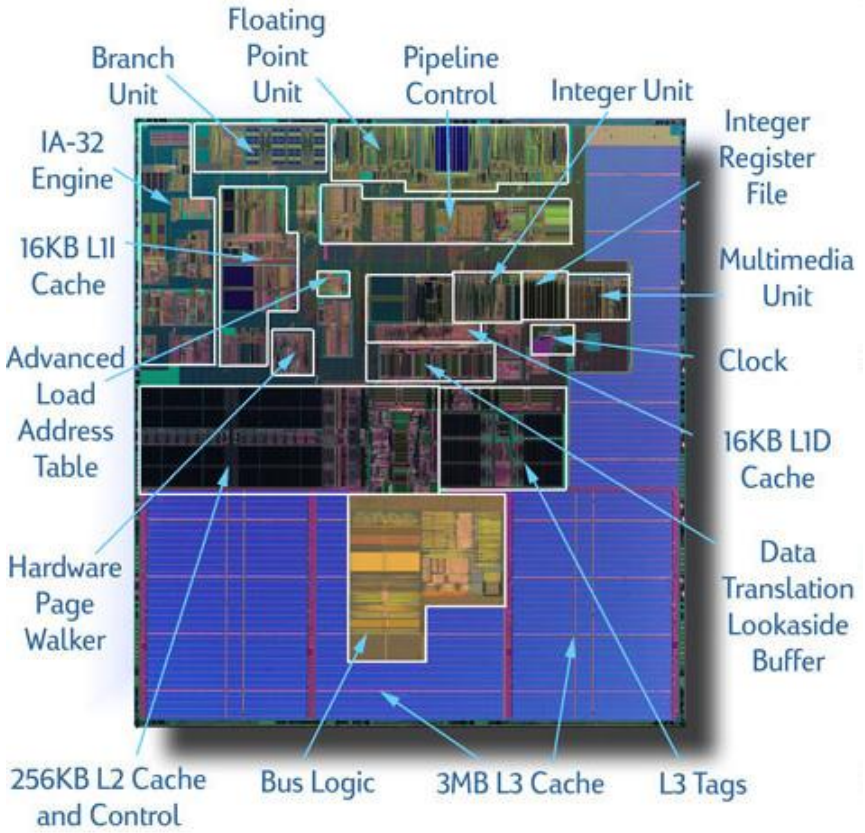
Philippe Darche
IUT Paris Descartes

Situation : Le fossé se creuse !



D'après [Hennessy et Patterson 03]

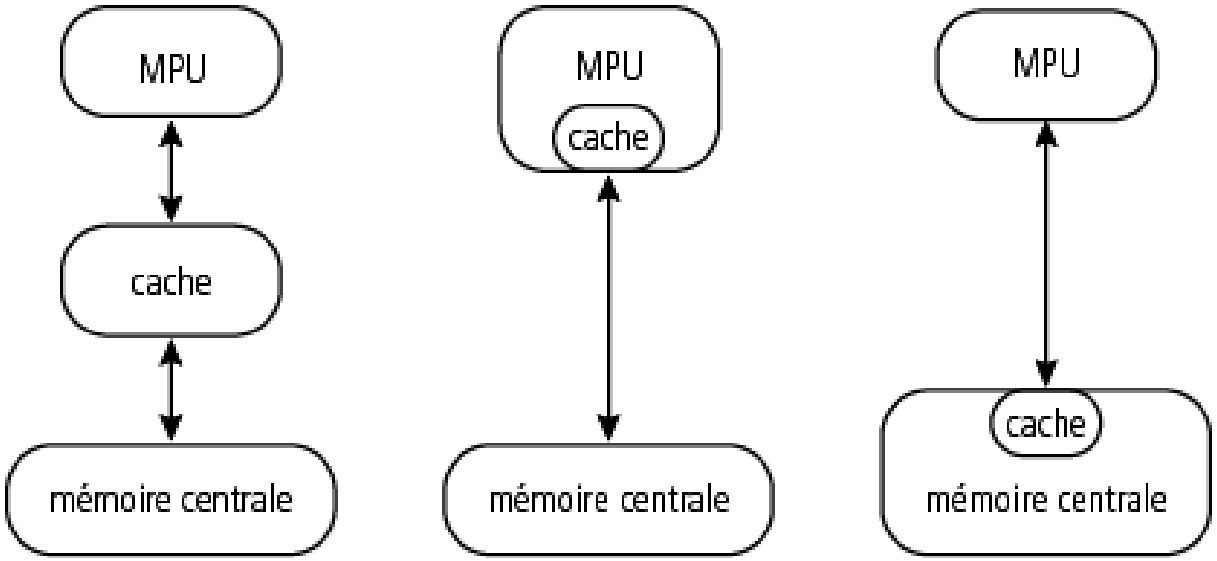
Une puce moderne



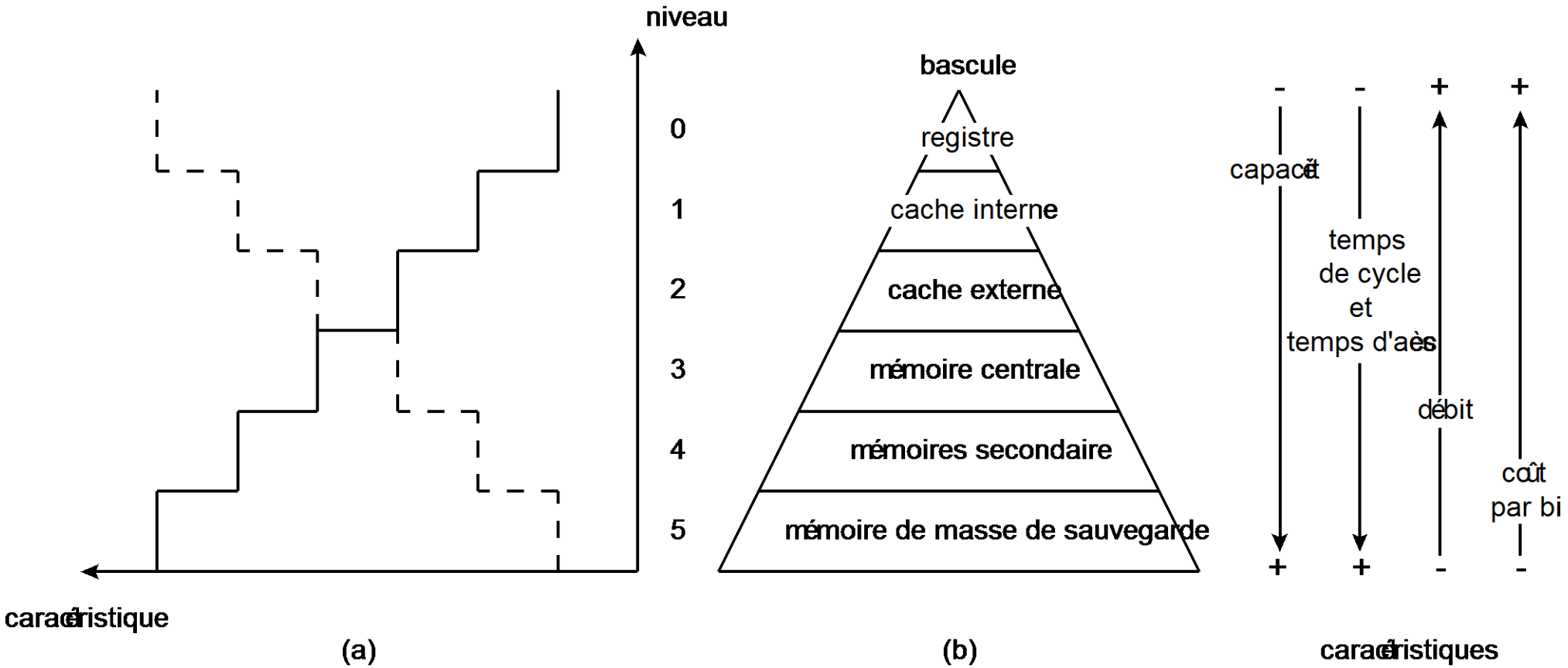
Intel® Itanium® 2 microprocessor

L’idée de base

- Intercaler une mémoire rapide de petite taille dans la hiérarchie mémoire



Place du cache dans la hiérarchie mémoire

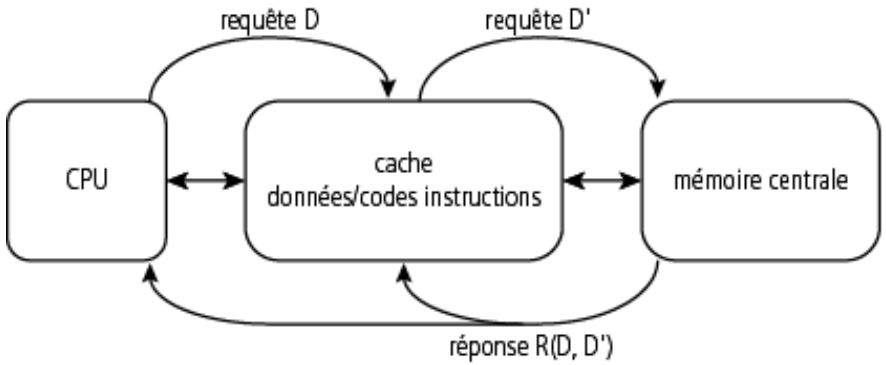


Cache

- Appelé aussi antémémoire
  - une catégorie de mémoire tampon
- D'abord mis en œuvre dans les *mainframes* et les supercalculateurs, puis dans les micro-ordinateurs
- Facteur 2^{10} entre tailles mémoire centrale et cache
- Algorithmes d'accès et de manipulation de données câblés
  - transparence
  - rapidité

Fonctionnement

- Lors d'une demande d'accès du CPU, si la donnée n'est pas le cache
  - ⇒ échec (*miss*) ou défaut de cache
  - accès à la mémoire centrale et mise-à-jour du cache
  - accès suivants : accès dans le cache
    - succès (ou *hit*)
  - ⇒ gain de temps



Important !

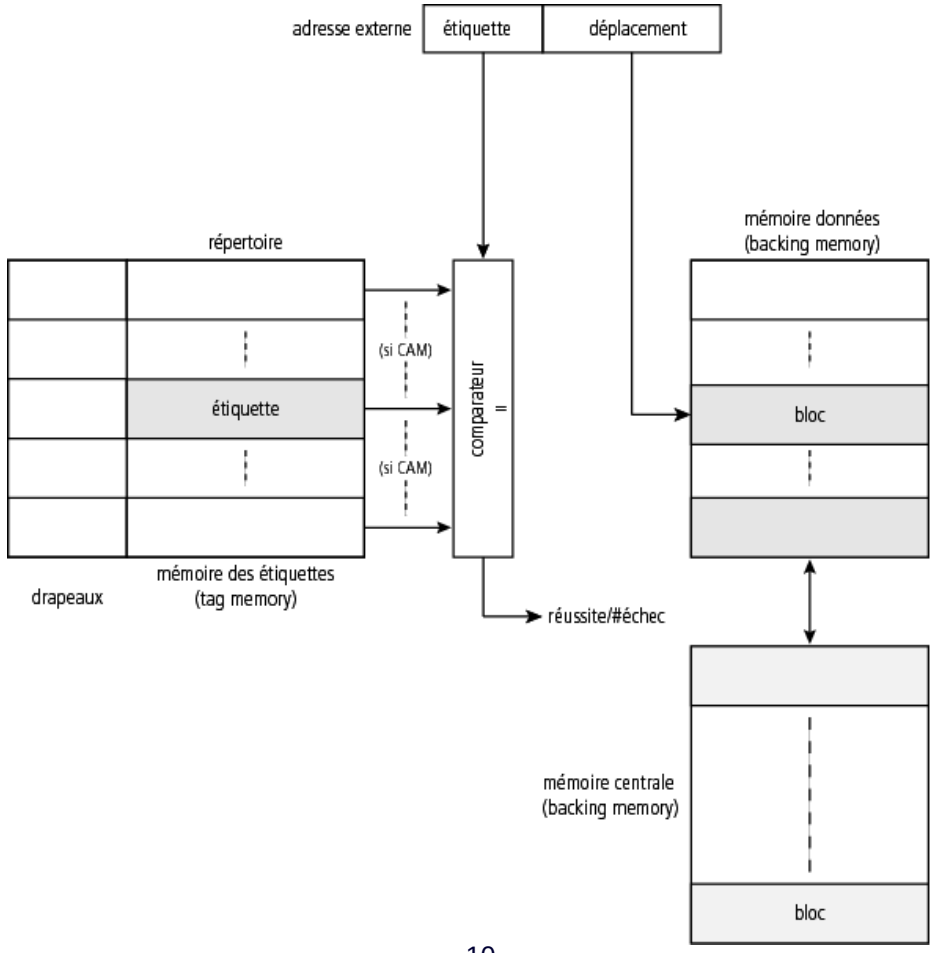
- L'unité d'échange entre CPU et cache : le mot
- L'unité d'échange entre cache et mémoire centrale : le bloc
- Choix du type de mémoire
  - mode rafale des mémoires synchrones adapté à l'accès bloc
  - rapidité

  $\Rightarrow$ modèle synchrone statique (SSRAM)

Pourquoi ce principe fonctionne ?

- Principes de localité
  - spatiale
    - les voisines d'une donnée adressée ont une forte chance d'être accédée
      - exemple : le tableau
  - temporelle
    - une donnée adressée a une forte chance d'être accédée de nouveau

Structure interne simplifiée d'un cache



Caractéristiques de la mémoire cache

- Dimensions du cache
  - taille globale
  - taille d'un bloc
- Mécanismes d'adressage
  - correspondance directe (*direct mapped*)
  - totalement associatif (*fully associative*)
  - mixte : associatif par ensemble de b blocs (*set-associative*)

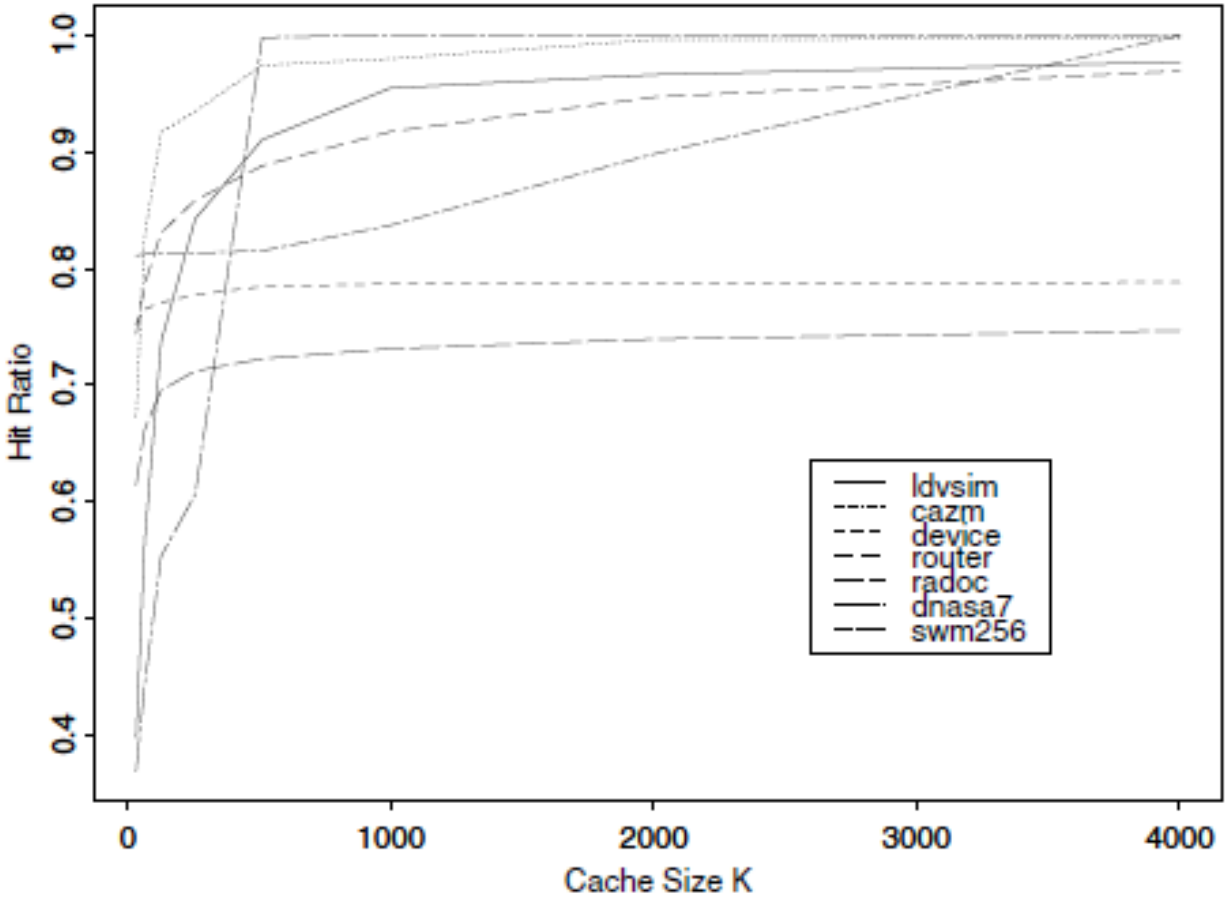
Caractéristiques de la mémoire cache (suite)

- Algorithmes de remplacement
  - LRU, LFU, FIFO, aléatoire
- Politique de mise à jour de la mémoire
  - simultanée (*write-through*)
  - différée (*deferred-write or write-back*)

Dimensionnement du cache

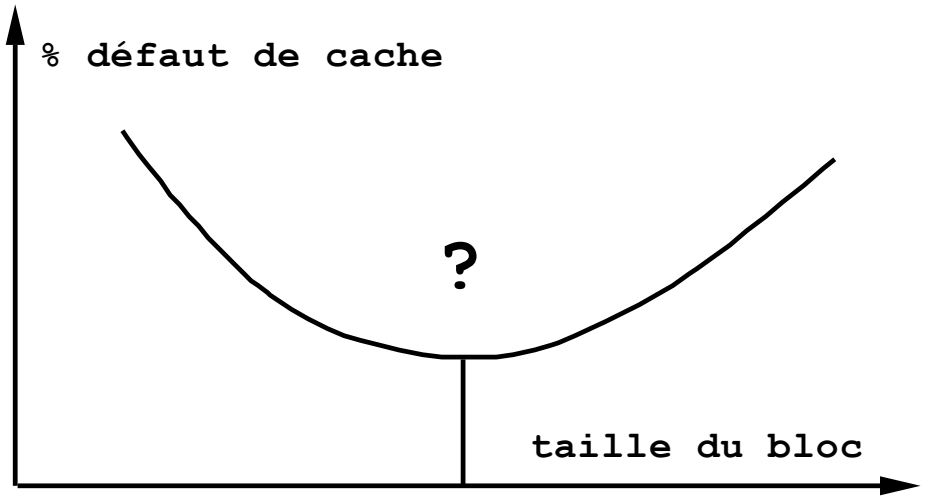
- Taille globale
  - compromis coût / performance
  - gain de performance décroissant en fonction de la taille
  - limitations physiques
    - taille de la mémoire centrale
    - capacité d'adressage du CPU !

Taille d'un cache L2

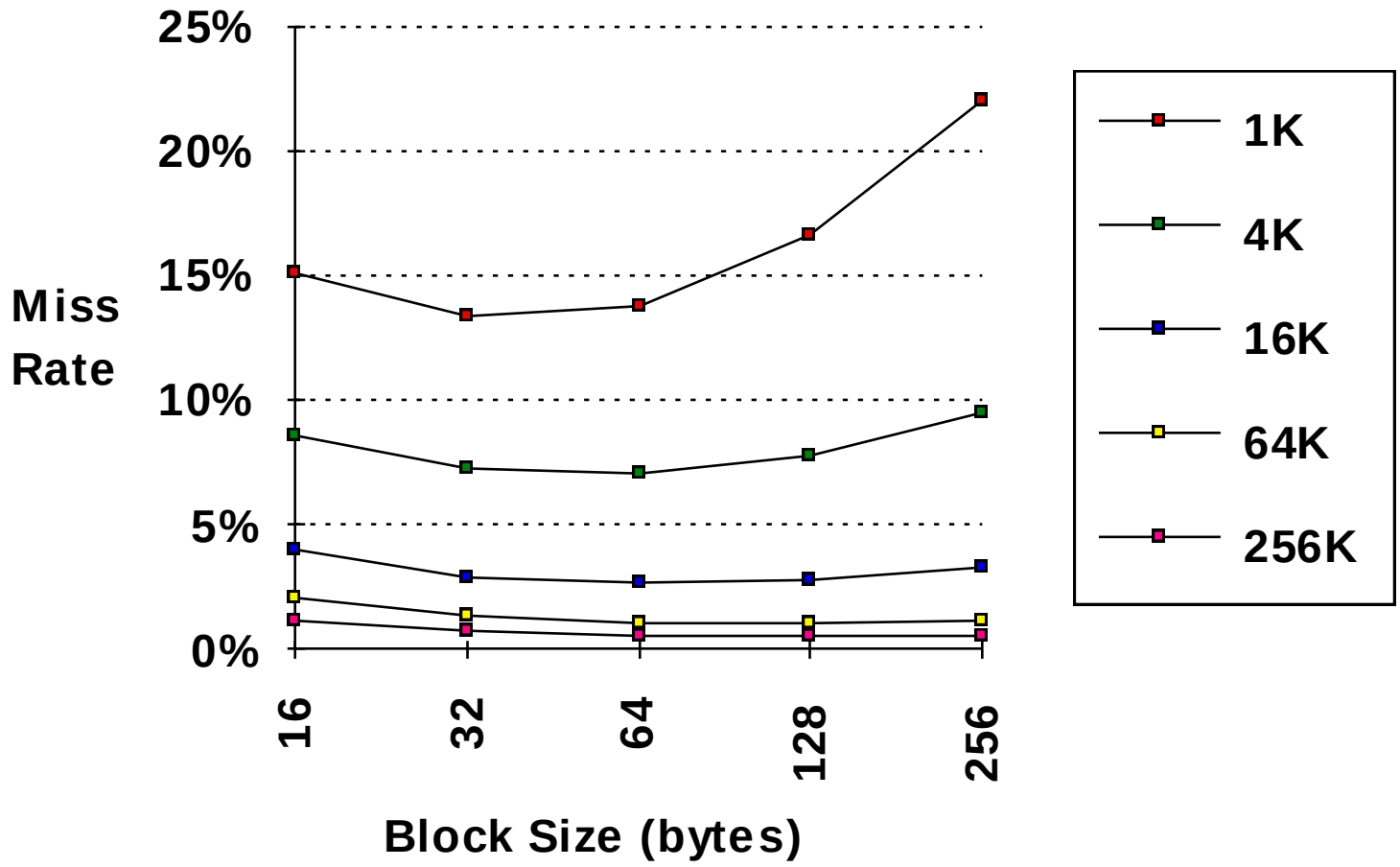


Dimensionnement du cache (suite)

- Taille du bloc
  - bloc plus grand ⇒ moins de blocs
  - bloc plus grand ⇒ moins de localité
  - empiriquement, 4 à 256 octets

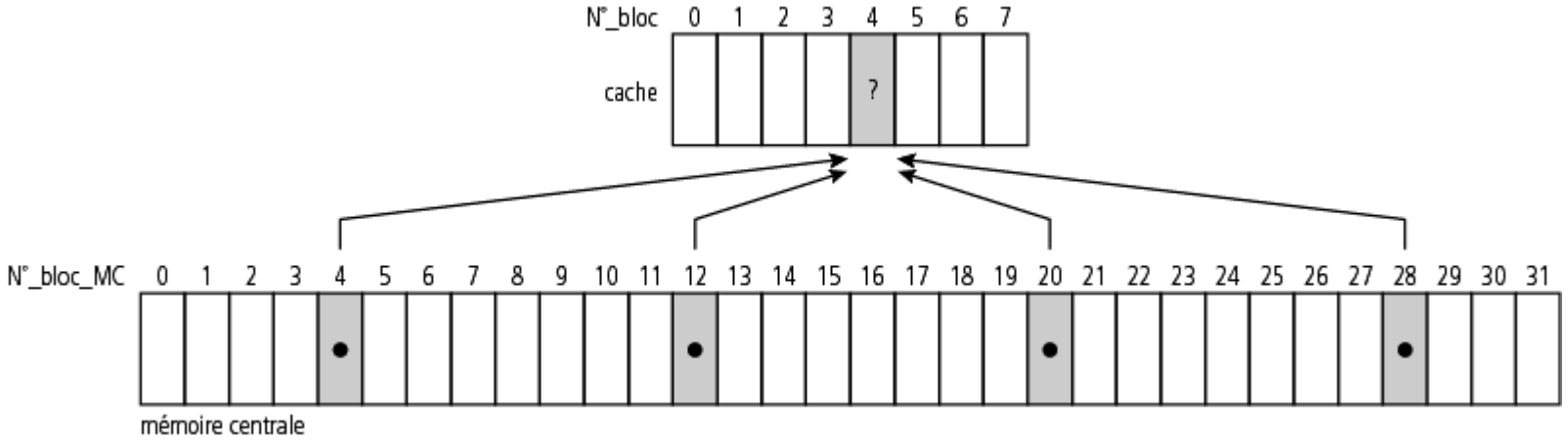


Taille d'un bloc

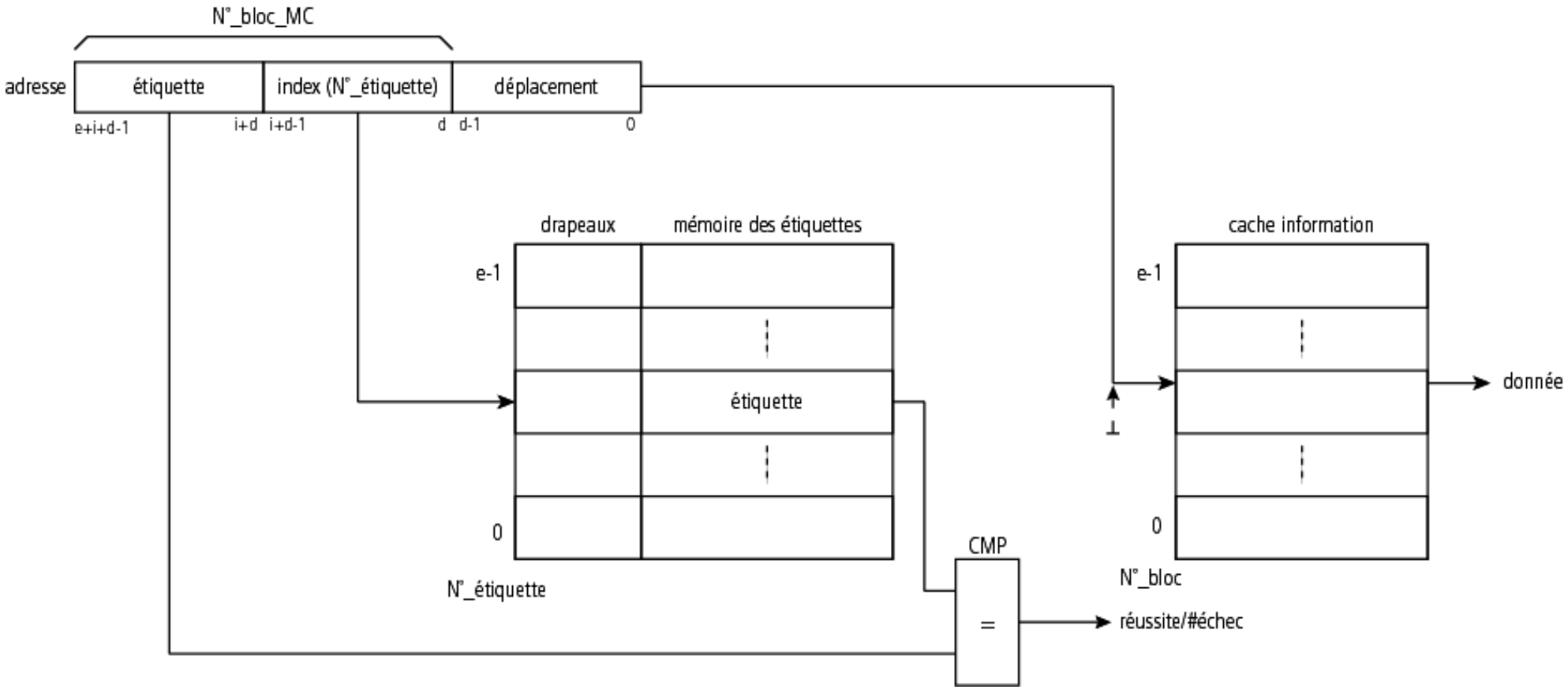


Adressage par correspondance directe

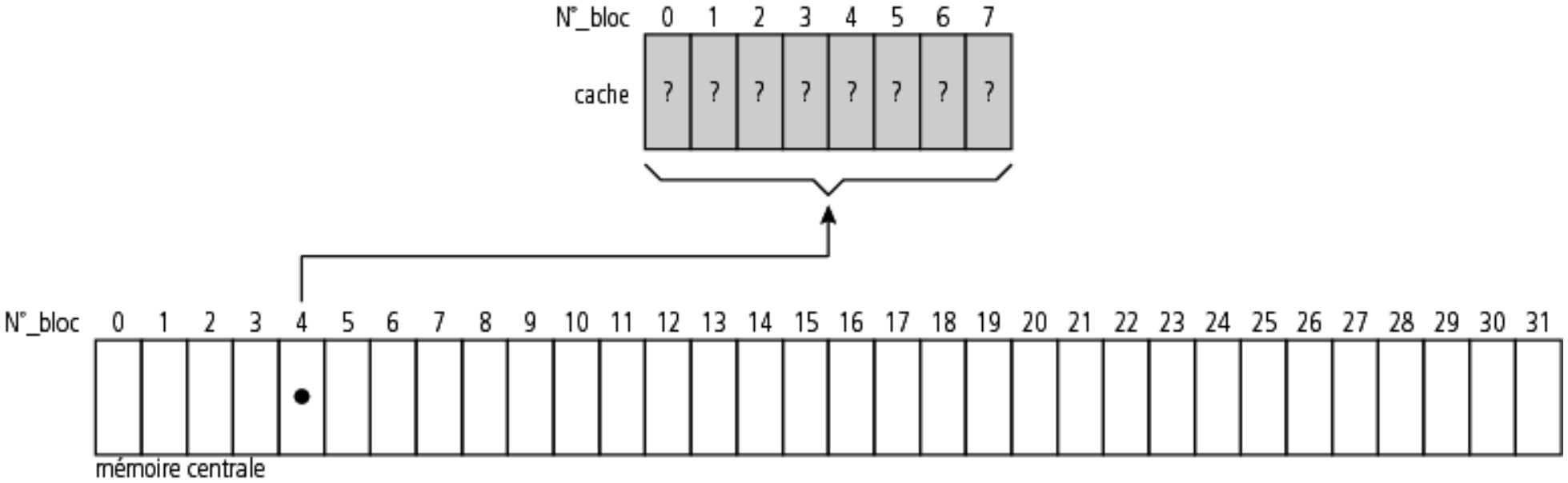
- Principe du carnet de N° téléphoniques



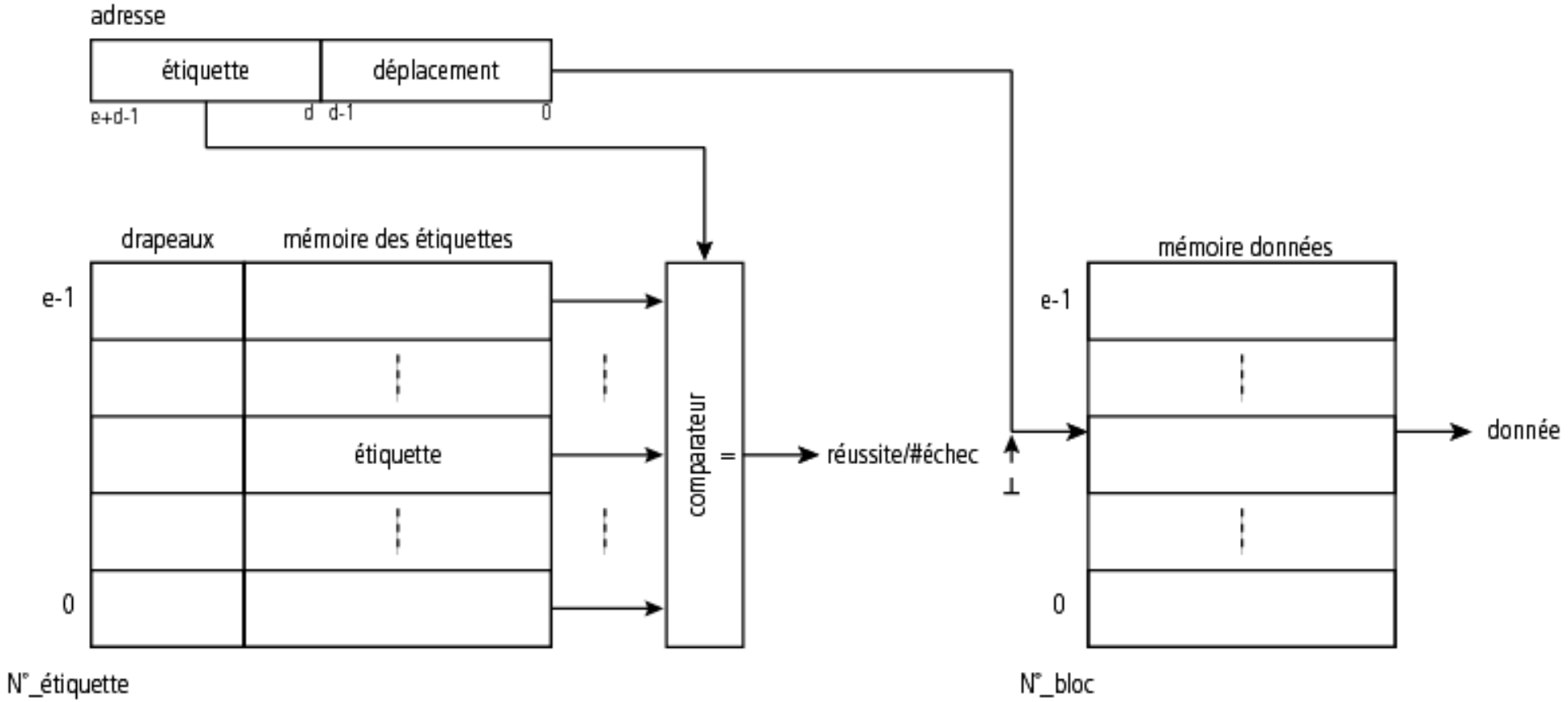
Adressage par correspondance directe



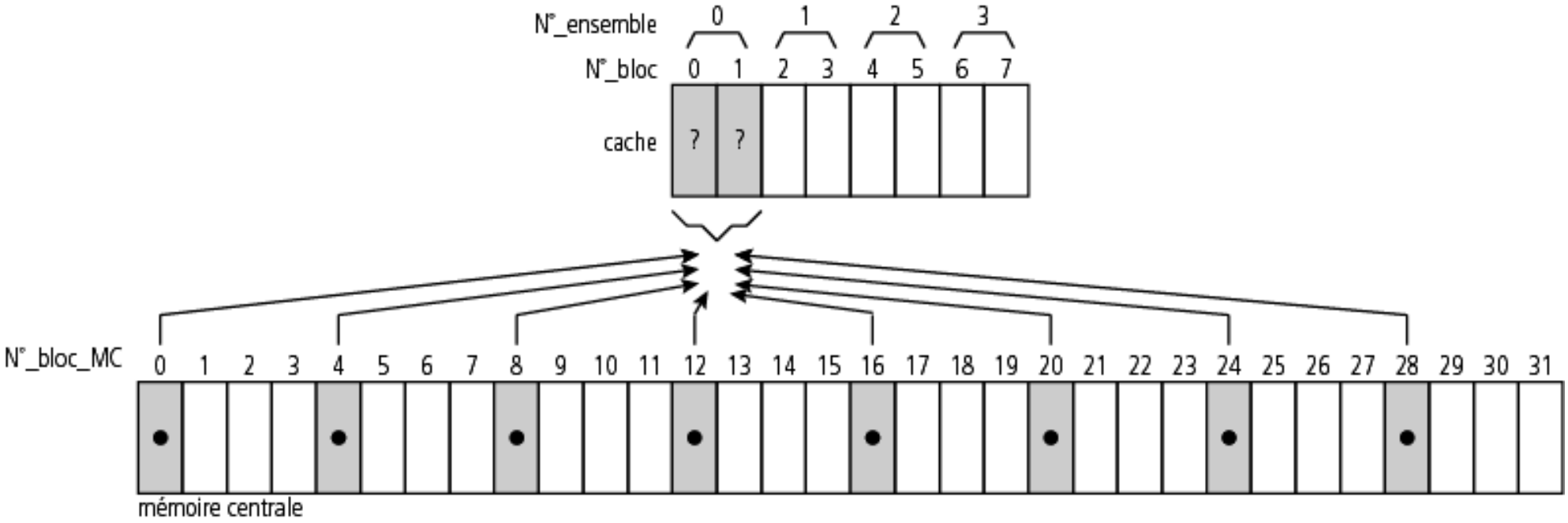
Adressage totalement associatif



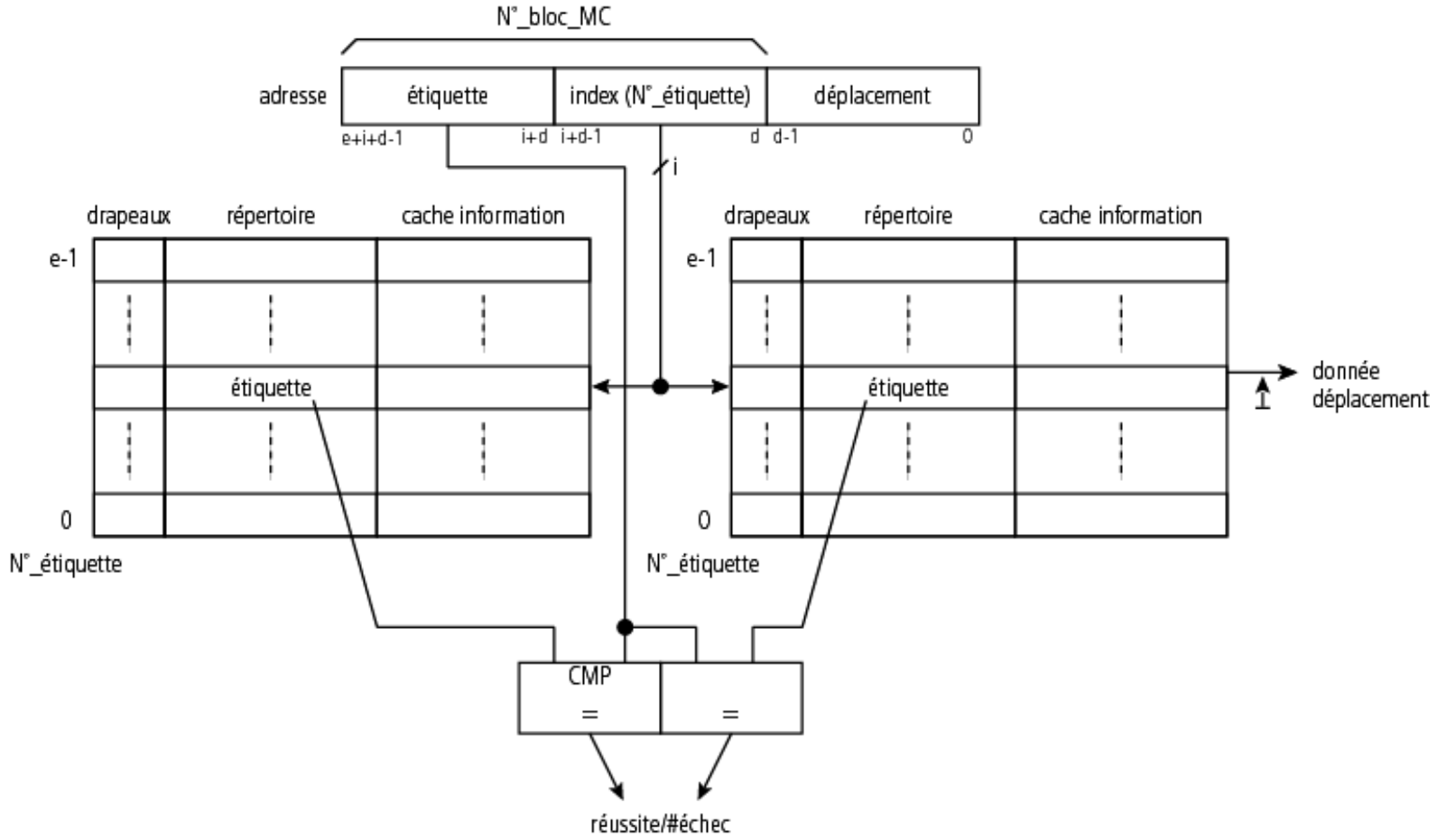
Adressage totalement associatif



Adressage associatif par ensemble de b blocs



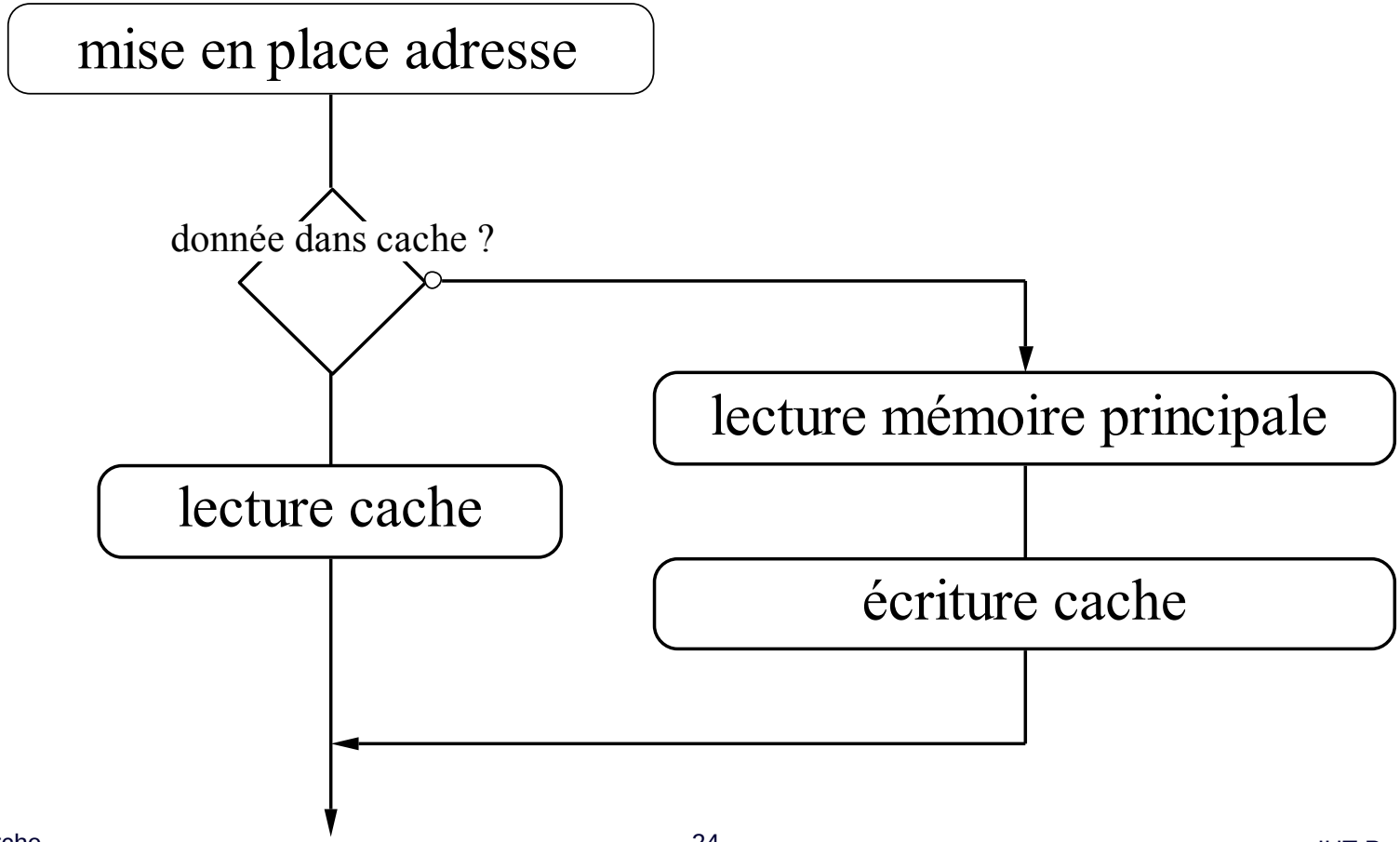
Adressage associatif par ensemble de b blocs



Caches de microprocesseur

- PowerPC601
  - associatif par ensemble de 8
  - 8 ensembles de 64 entrées de 16 mots
  - 2 caches de 16 Ko instruction et données
  - politique de remplacement LRU
- Pentium
  - associatif par ensemble de 2
  - 2 caches de 8Ko instruction et données
  - politique de remplacement LRU
- i486
  - associatif par ensemble de 4
  - bloc de 16 octets

Algorithme de lecture



Algorithmes de remplacement

- Quel bloc éliminer pour faire place à un nouveau bloc ?
- Objectifs
  - simple et peu coûteux à réaliser
  - respectant au mieux le principe de localité

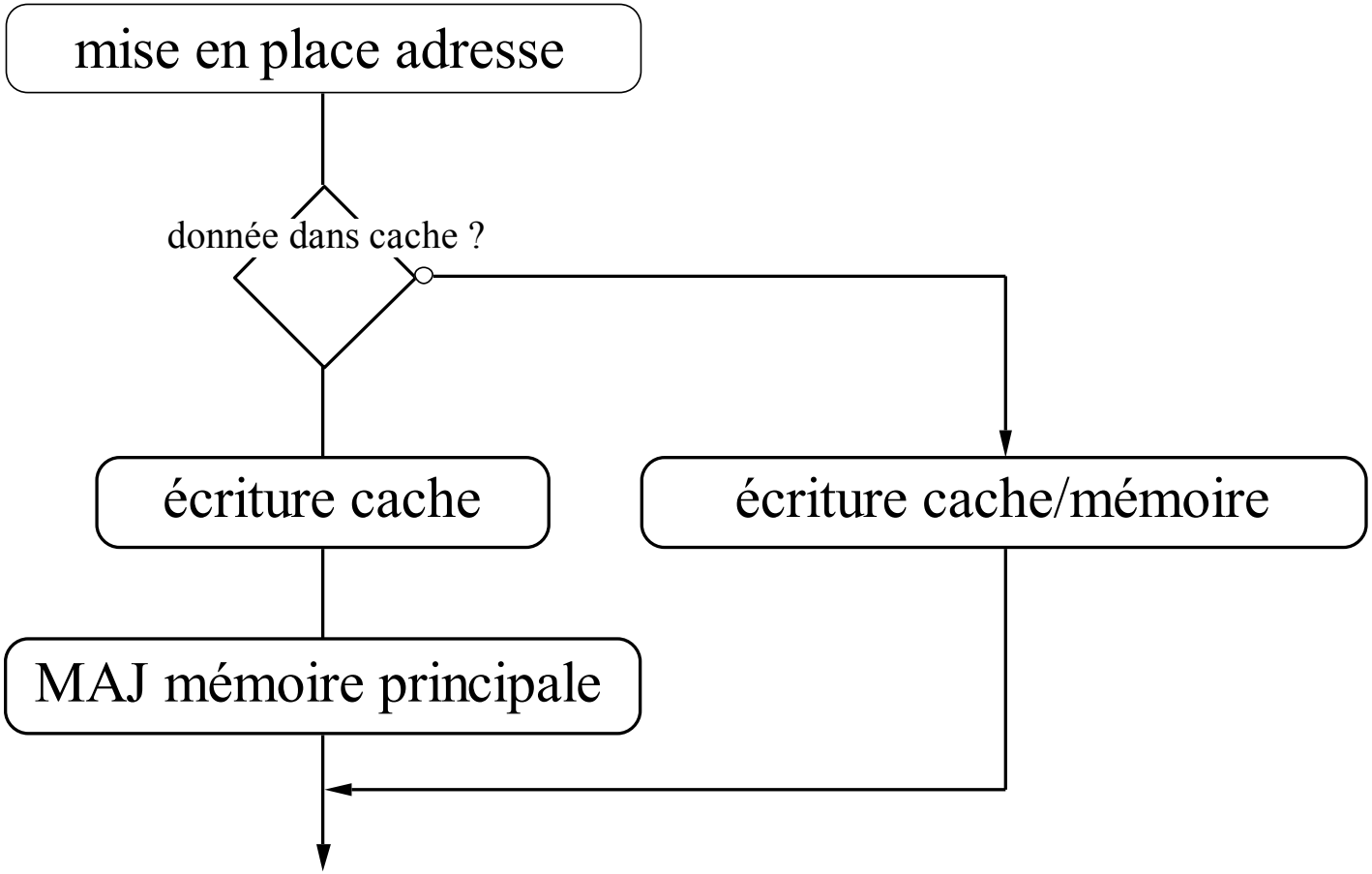
Techniques de remplacement

- Aléatoire
  - suppose l'utilisation future indépendante du passé
- FIFO : premier entré , premier sorti (*First In First Out*)
  - liste des blocs
- LRU : moins récemment utilisé (*Last Recently Used*)
  - bits d'utilisation
- LFU : moins fréquemment utilisé (*Least Frequently Used*)
  - compteur d'accès pour chaque bloc

Politiques d'écriture du cache

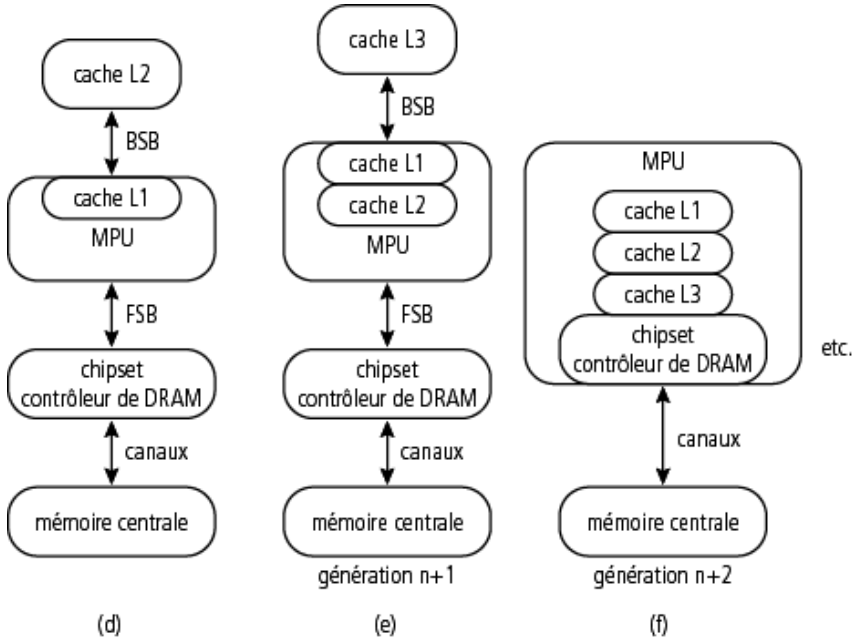
- Quand réécrire en mémoire un bloc modifié dans le cache ?
  - écriture simultanée cache et mémoire principale (*write-through*)
    - \+ simplification du remplacement
    - \- augmentation du trafic sur le bus
  - écriture différée (*deferred-write or write-back*)
    - \+ optimisation des écritures
    - \- problème de cohérence avec les modules d'E/S
- Critère quantitatif déterminant
  - % écritures / accès (≈ 0,15)
  - nombreuses écritures ⇒ écriture différée
  - peu d'écritures ⇒ écriture simultanée !

Algorithme d'écriture



Plusieurs niveaux de cache

- Primaire (*primary (or first-level) cache*) → L1
- Secondaire (*secondary (or second-level) cache*) → L2
- tertiaire (*tertiary (or third-level) cache*) → L3



Conclusion

- Tendance forte à l'intégration
  au niveau du processeur
- Attention à la cohérence lors de l'arrêt
- Consommation électrique non négligeable
- Principe réutilisable au niveau du HDD