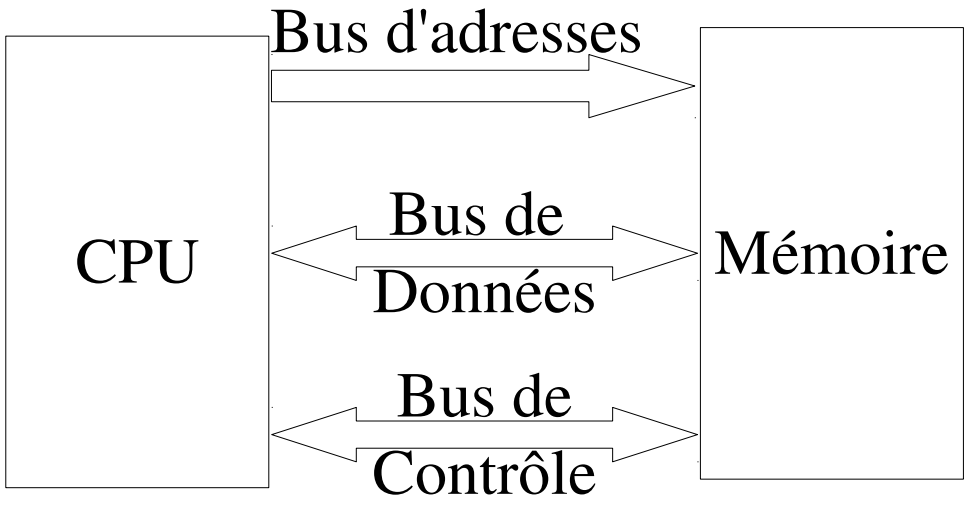
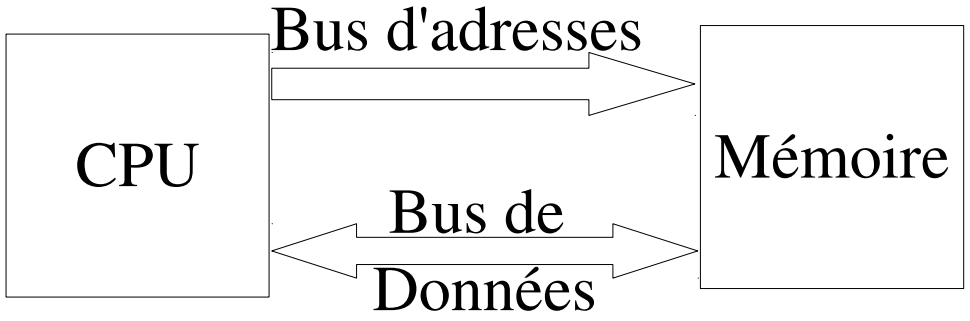
Informatique Embarquée
M2 / 2017

Gestion Mémoire

Interaction CPU / mémoire

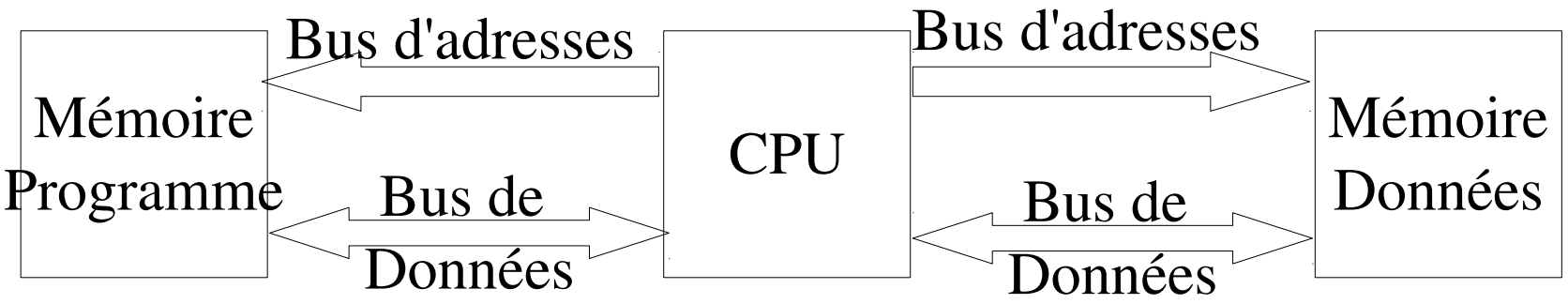


Architecture Von Neumann



- Mémoire contient:
  - Instructions et Données
  - CPU charge les instructions depuis la mémoire
- CPU possède des registres
  - PC: instruction à exécuter, SP: sommet de pile
  - registres généraux

Architecture Harvard

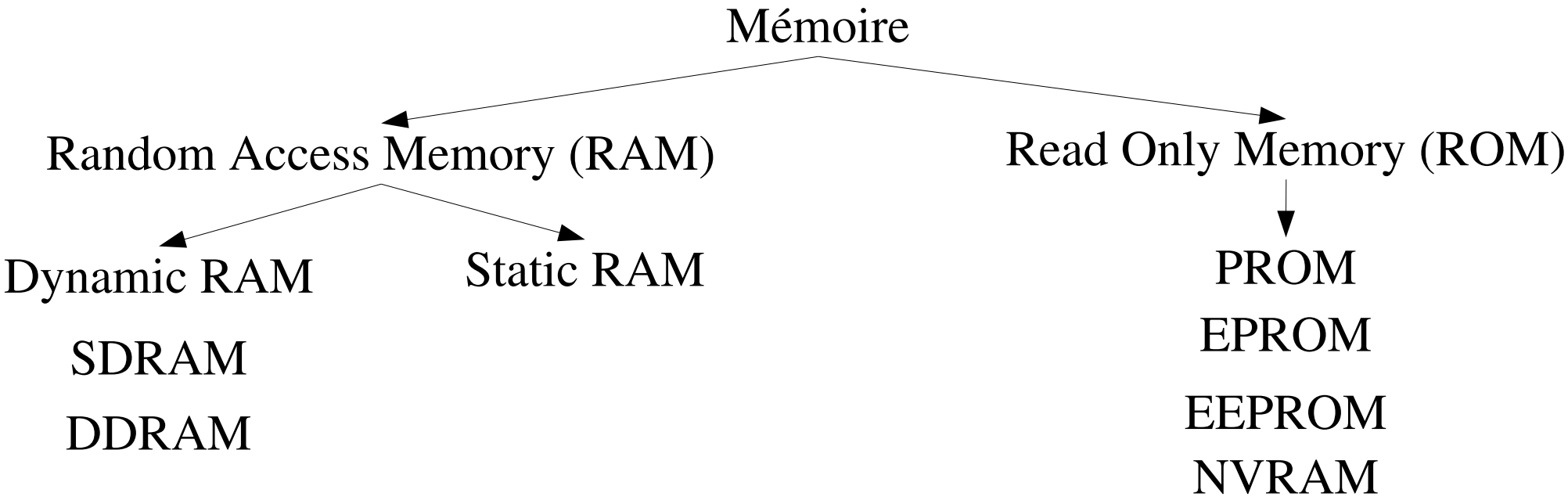


- Deux types de mémoire
  - Instructions et Données
  - Accès en parallèle aux instructions et aux données
    - Données accédées plus fréquemment que les instructions
- CPU idem à Von Neumann

Mémoire

- Dans les systèmes embarqués, requis pour stocker programmes et données:
  - Code, "firmware"
    - Permanent, en général jamais changé durant la vie de l'appareil
  - Données temporaires
    - variables, tas, pile,...
    - Rapide et effaçable
  - Données de configuration
    - Changent, évoluent, ne doivent pas être volatiles
    - Ex: répertoire téléphone mobile

Types de Mémoires

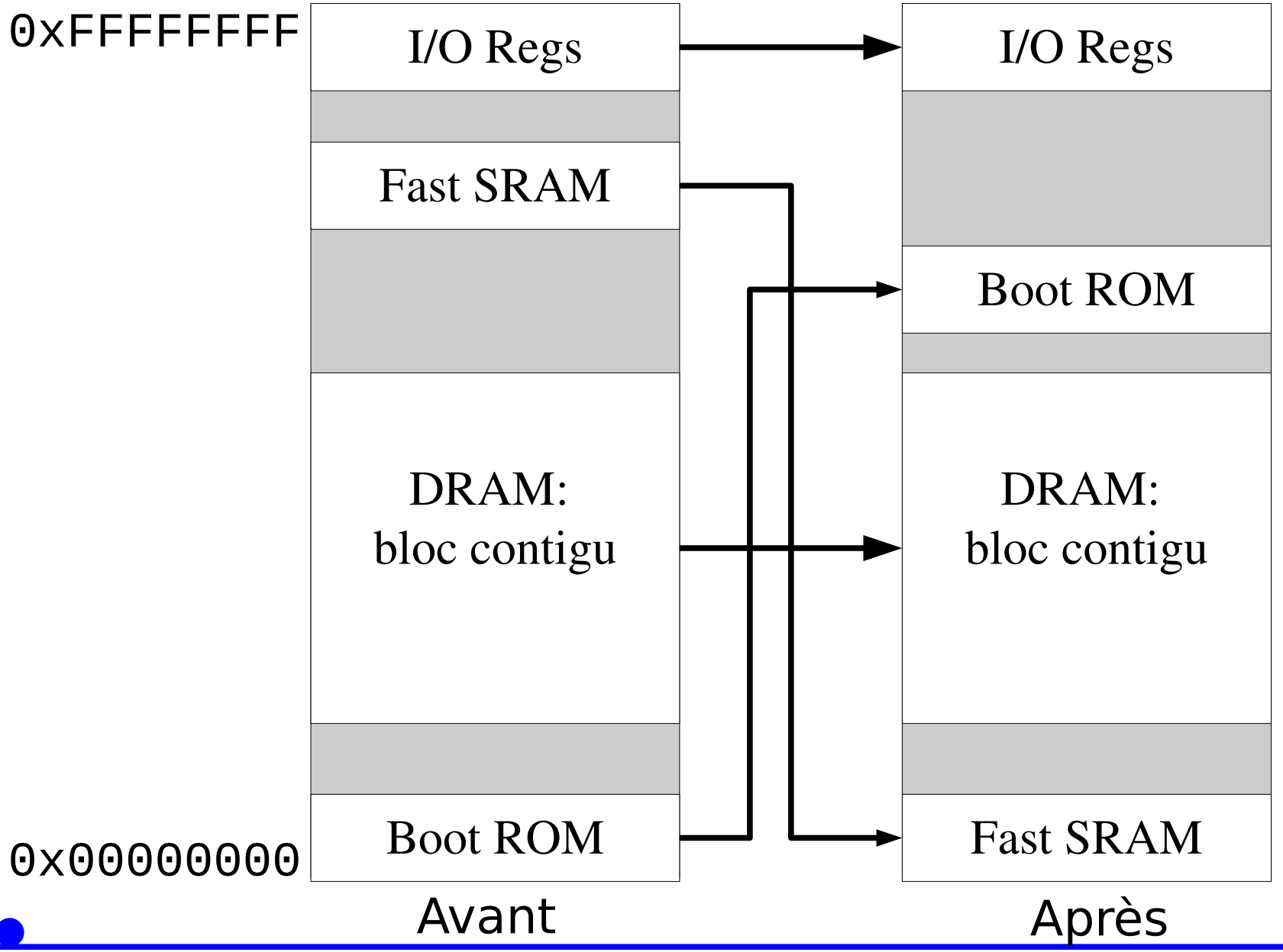


- SRAM: pas de rafraîchissement
- DRAM: rafraîchissement, Contrôleur

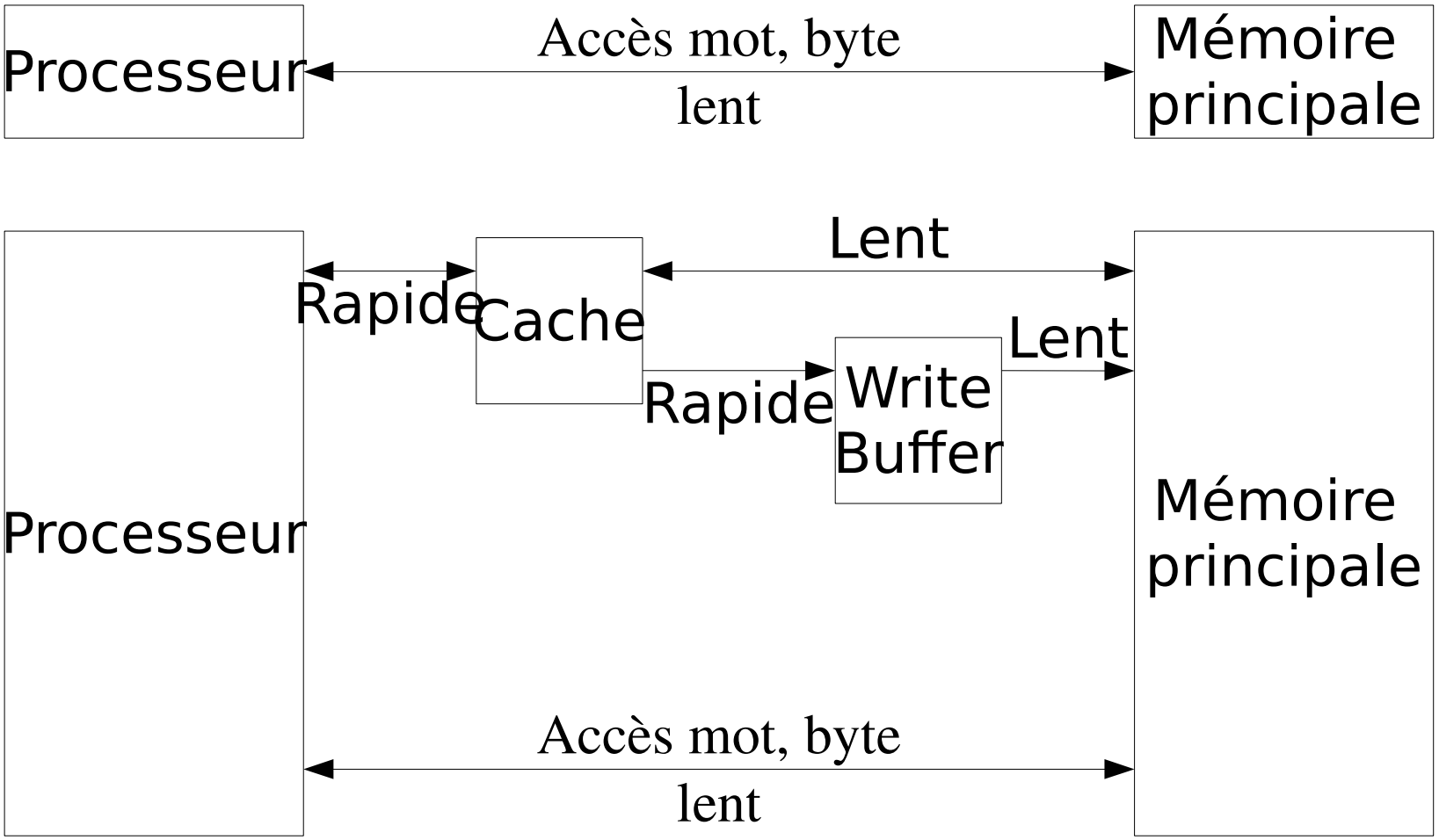
Initialisation (ARM)

- État initial : "reset" et configure le matériel de telle manière que l'OS puisse s'exécuter.
- Configuration:
  - contrôleur mémoire, caches, quelques périphériques
- Diagnostics:
  - Vérifie si le matériel fonctionne, identifie et isole les fautes
- Boot:

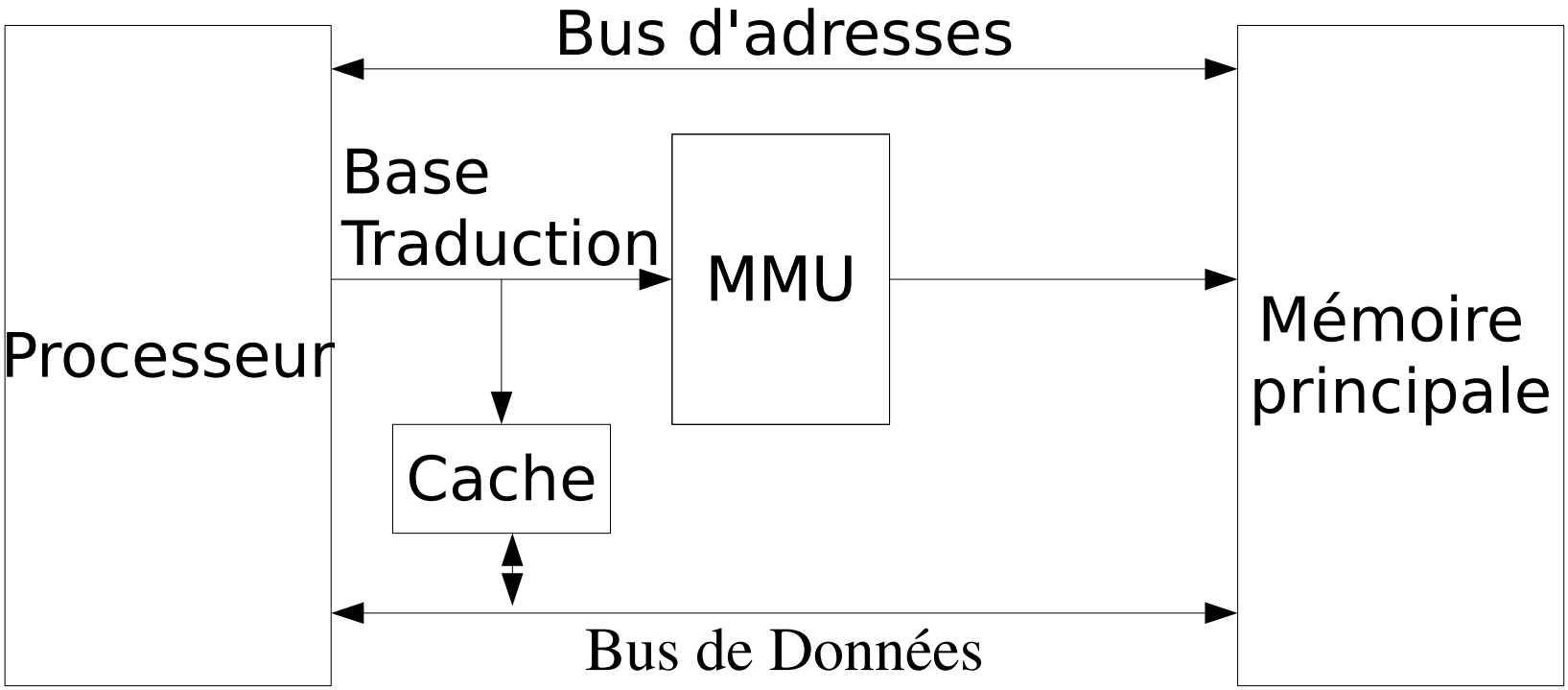
Réorganisation Mémoire



Caches Mémoire

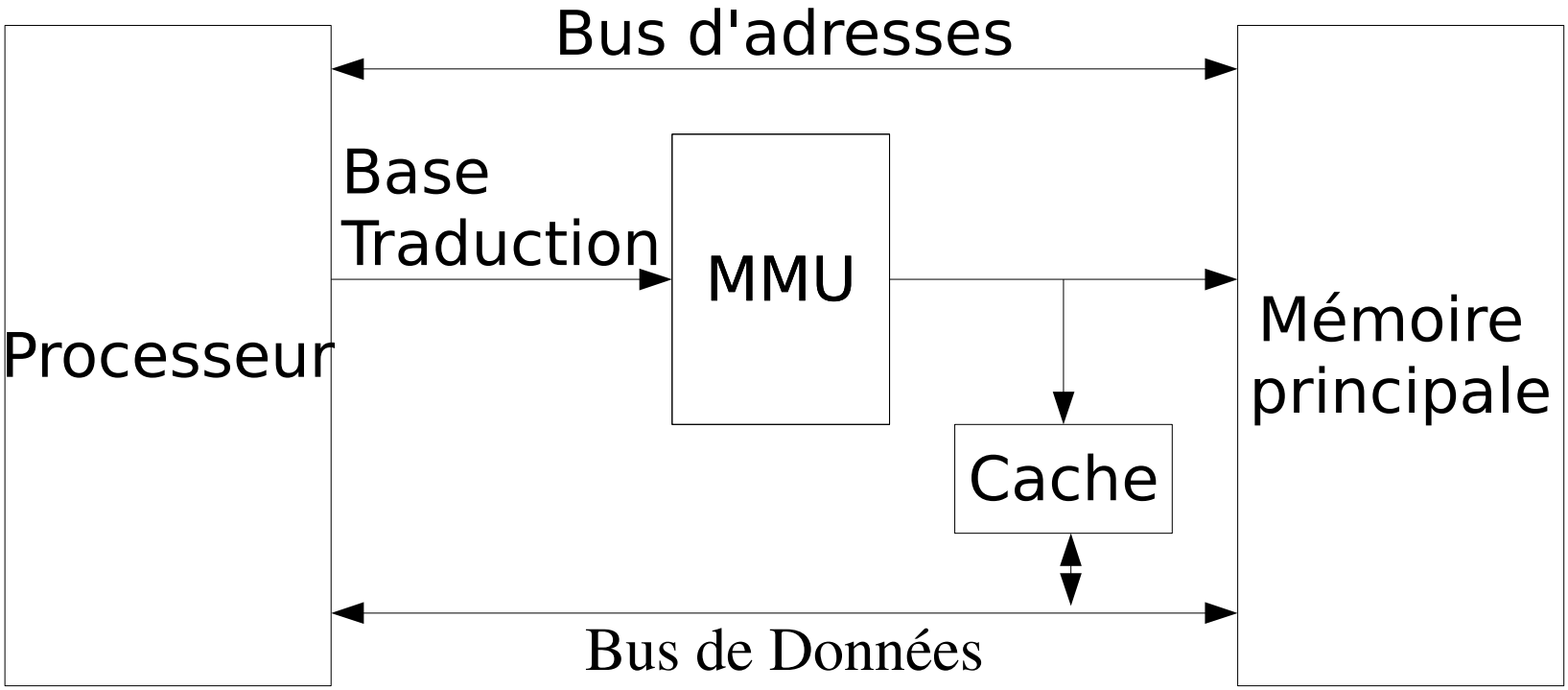


Cache et MMU



- Cache Logique

Cache et MMU

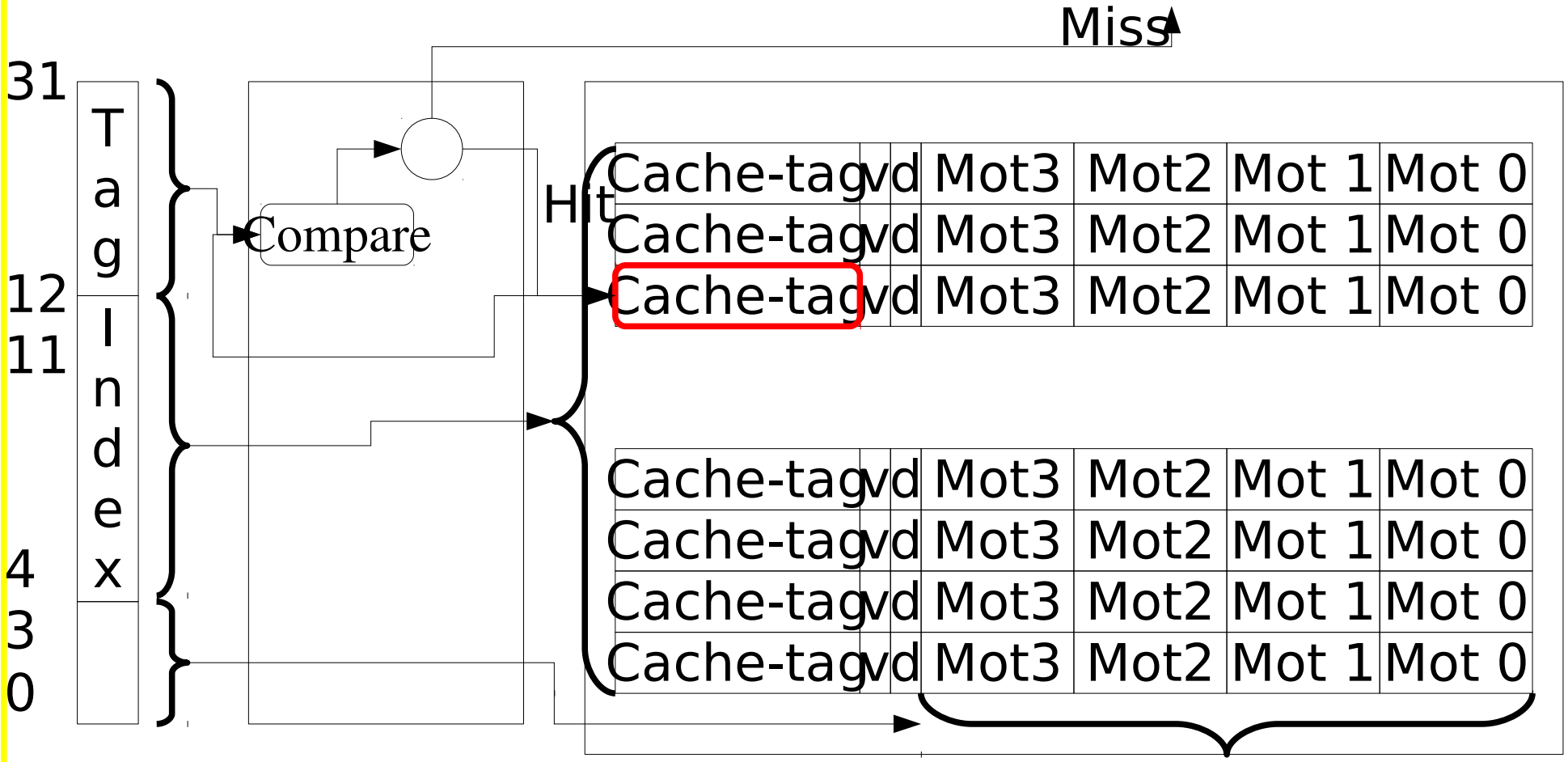


- Cache Physique

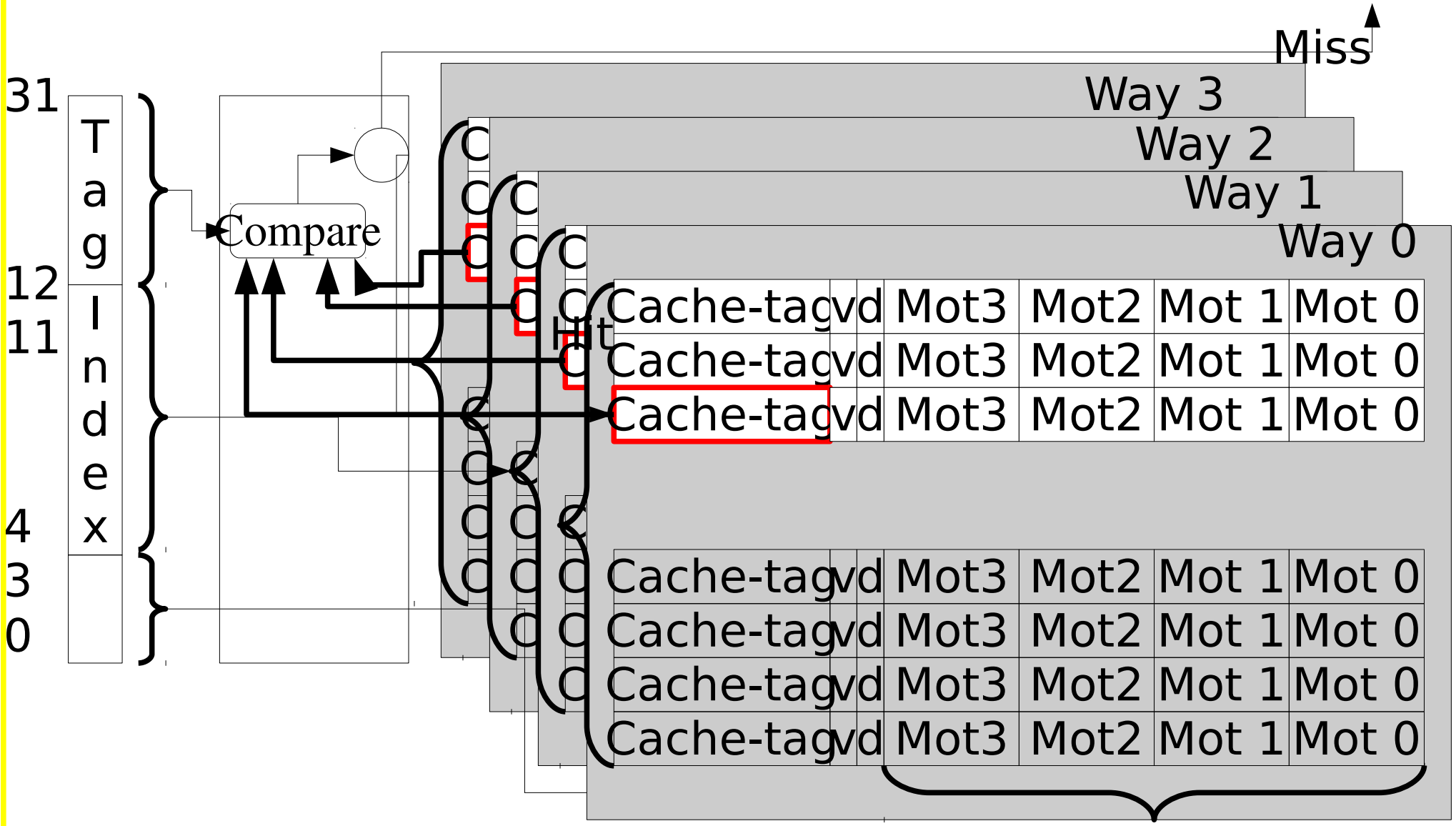
Cache

- Le cache est chargé avec des "blocs"
  - Zones contiguës de mots mémoire
  - Lignes
  - Identifier ces lignes
- État:
  - Ligne valide ou non
  - Ligne modifiée ou non

Architecture d'un cache



Caches Associatifs



Caches: Politique de gestion

- Writethrough
  - Toute modification est immédiatement répercutée sur la mémoire centrale. Lent
- Writeback (utilisation du dirty bit)
  - Écrit dans une ligne de cache valide
  - Ligne de cache contient des données plus récentes
  - Ligne écrite quand nécessité de remplacer la ligne
- Politiques de remplacement:
  - Round robin ou pseudo aléatoire

Caches : contrôle

- Flush: vider tout le contenu
- Clean: pousser toutes les données modifiées en mémoire
- Possibilité d'effectuer ces opérations par portion (granularité: la ligne)
- Opérations pour I caches et D caches
- Possibilité de verrouiller du code et des données en cache.

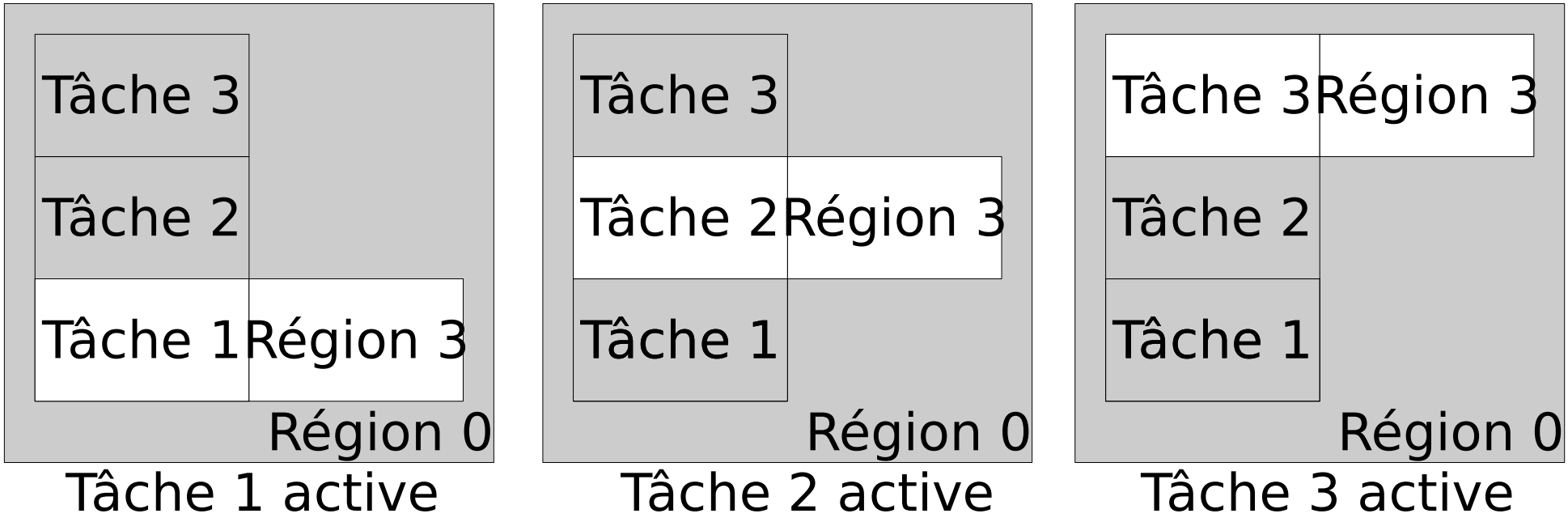
Memory Protection Unit

- Définit au niveau matériel des "régions"
  - Adresse de départ et longueur
  - Droits: lecture, lecture/écriture, pas d'accès, cache et write buffer
- Droits comparés avec le mode du processeur lors d'un accès mémoire pour déterminer si l'accès est valide ou non.
  - Invalide: génération d'une exception "abort"

MPU: régions

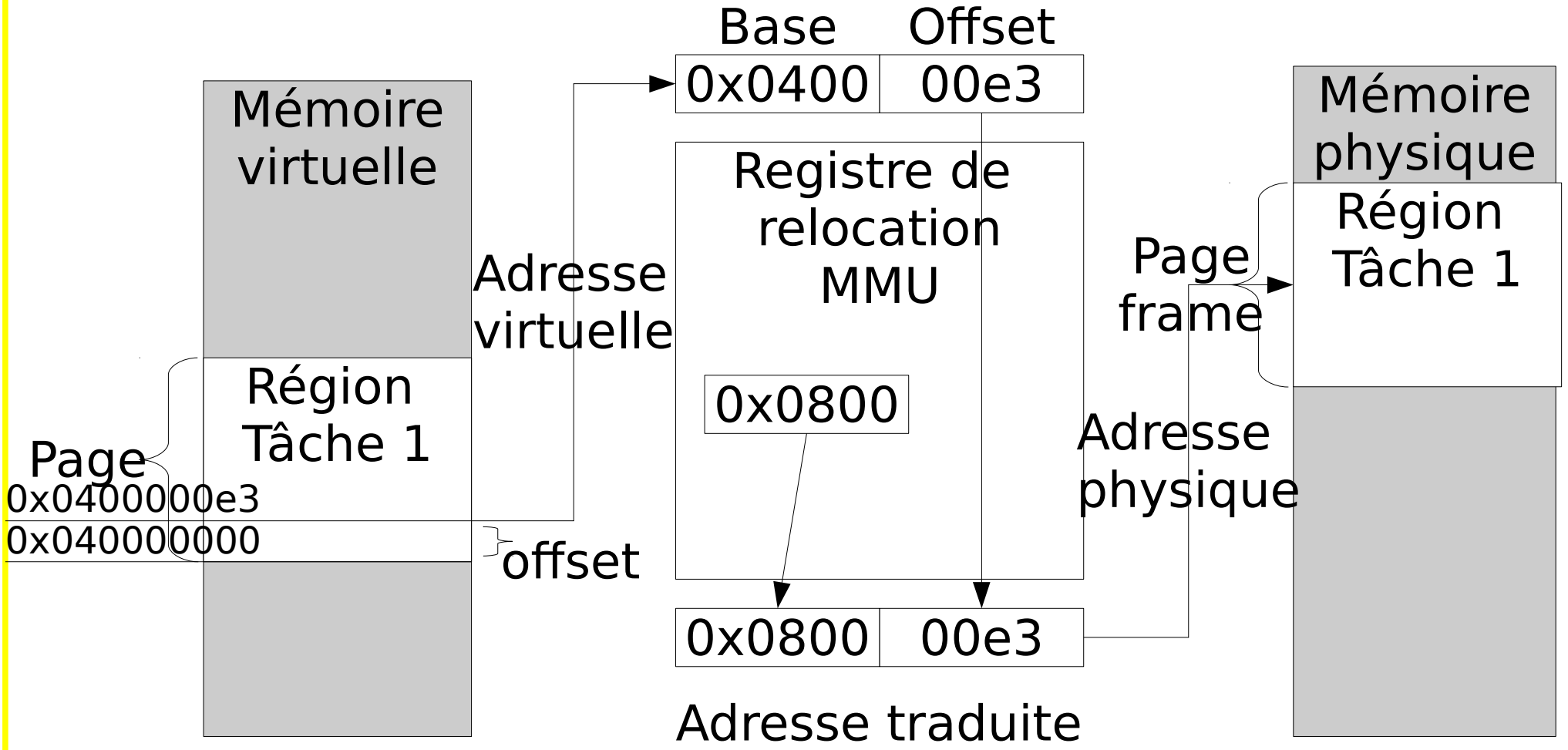
- Régions peuvent se recouvrir
- A chaque région est attribué une priorité, indépendante de ses privilèges
- En cas de recouvrement les droits de la région de plus haute priorité sont appliqués
- L'adresse de départ d'une région est un multiple de sa taille
- La taille d'une région est une puissance de 2 entre 4KB et 4GB
- Accès hors région entraîne un abort

Exemple régions MPU



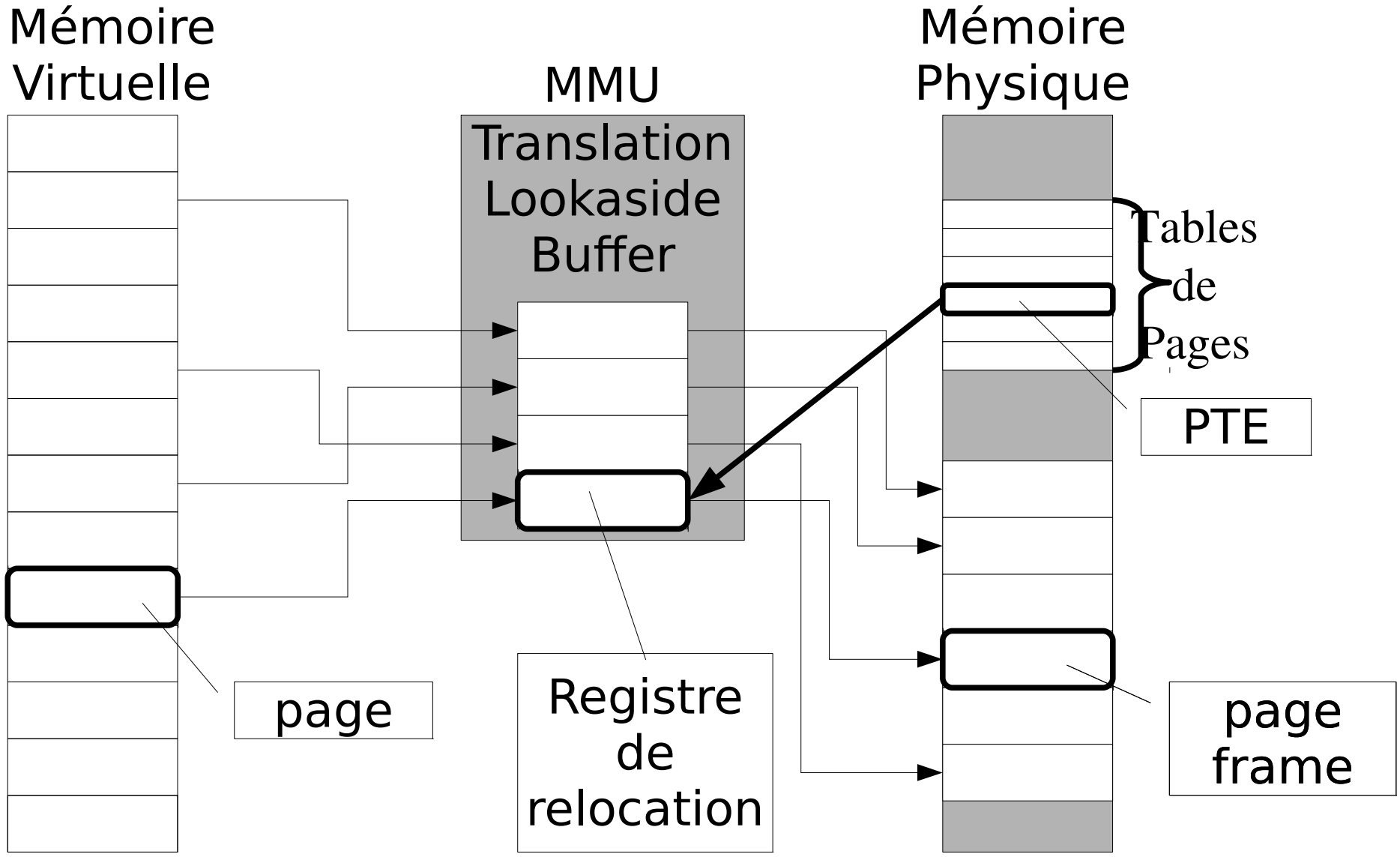
- Région 0: accès privilégié seulement
- Région 3: redéfinie à chaque changement de tâche, accès en mode utilisateur:
  - une tâche active ne peut pas corrompre les régions dont elle n'a pas besoin

Memory Management Unit

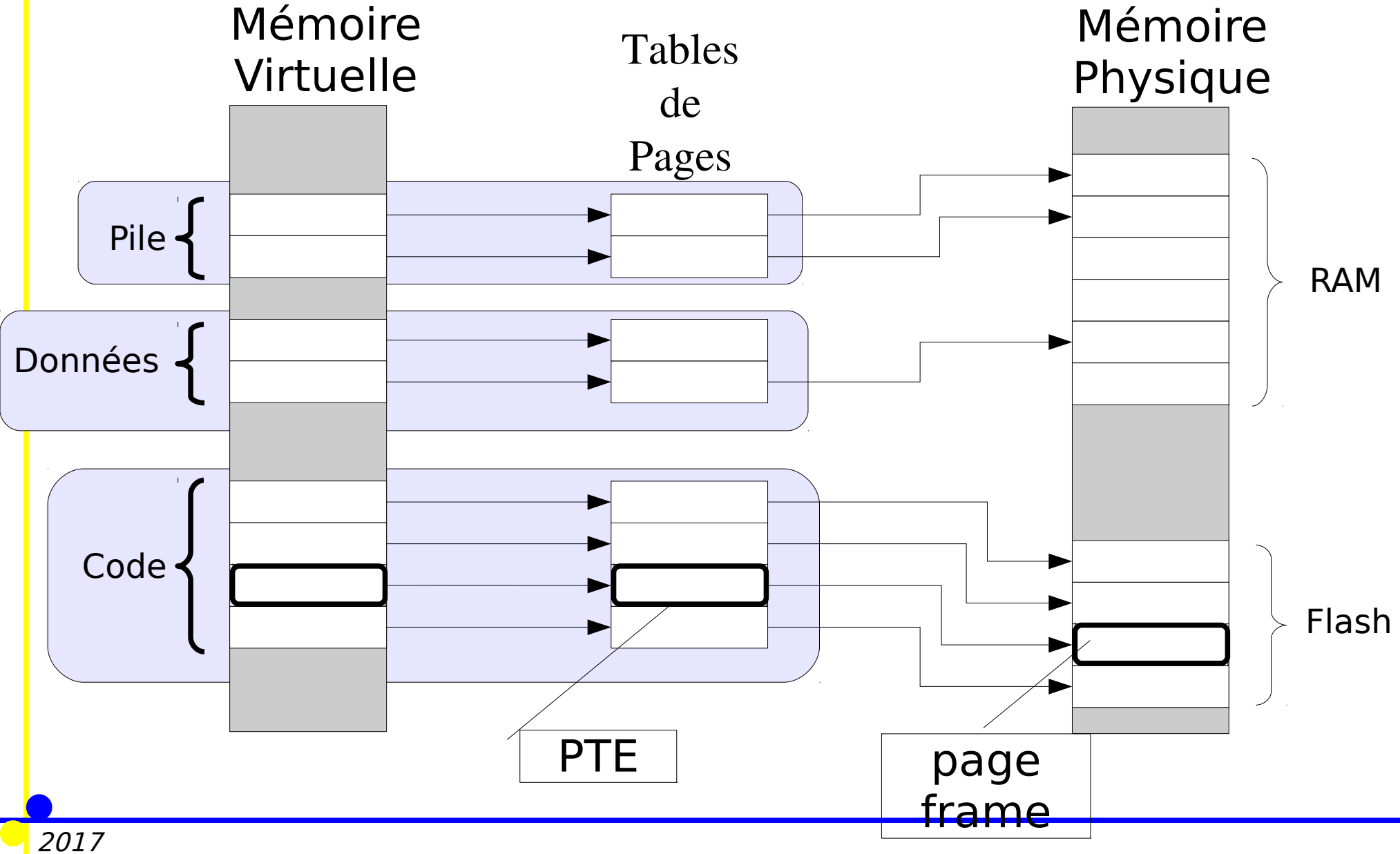


- Une autre tâche avec la même adresse virtuelle accédera une page physique différente

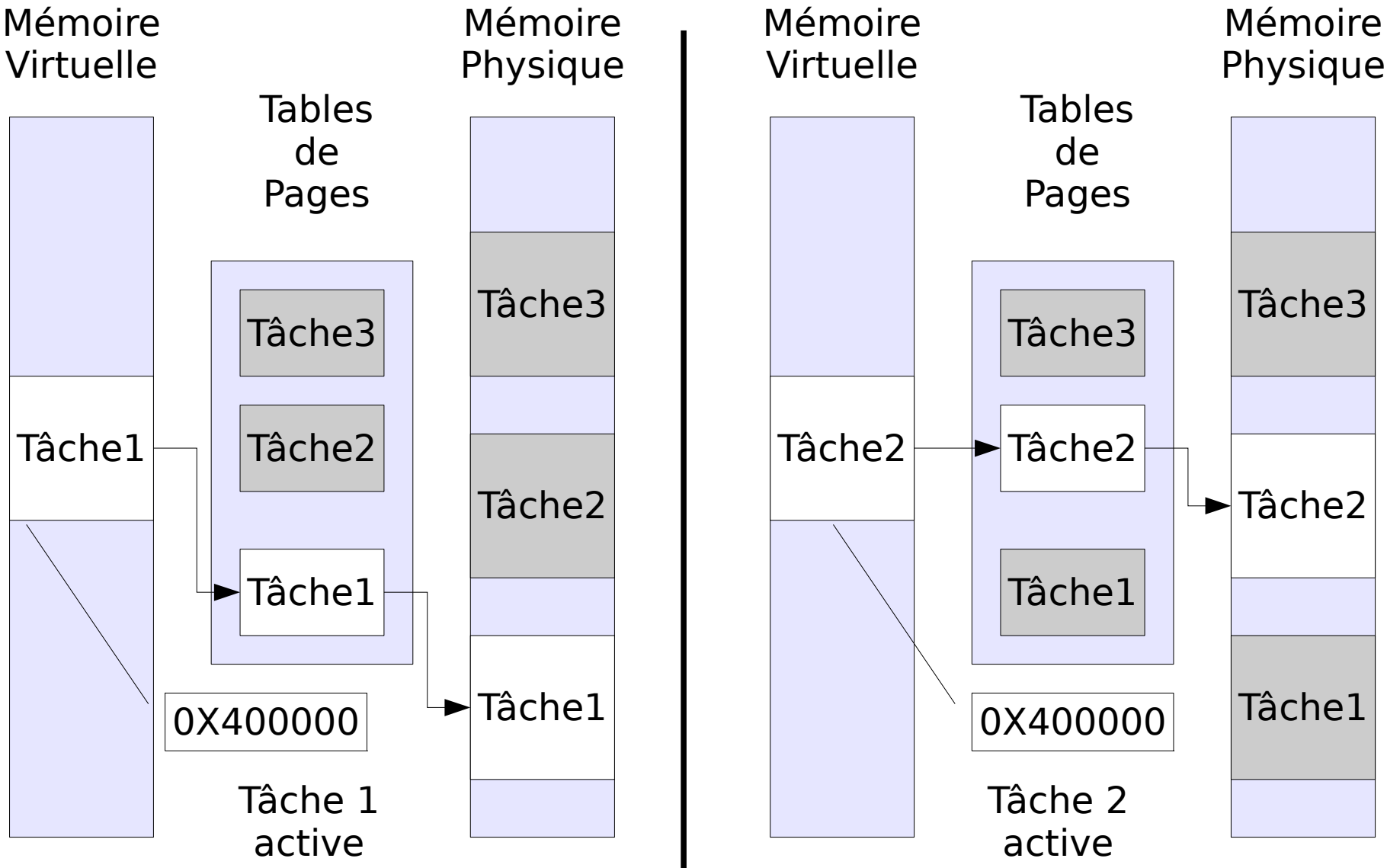
Mémoire Virtuelle: composants



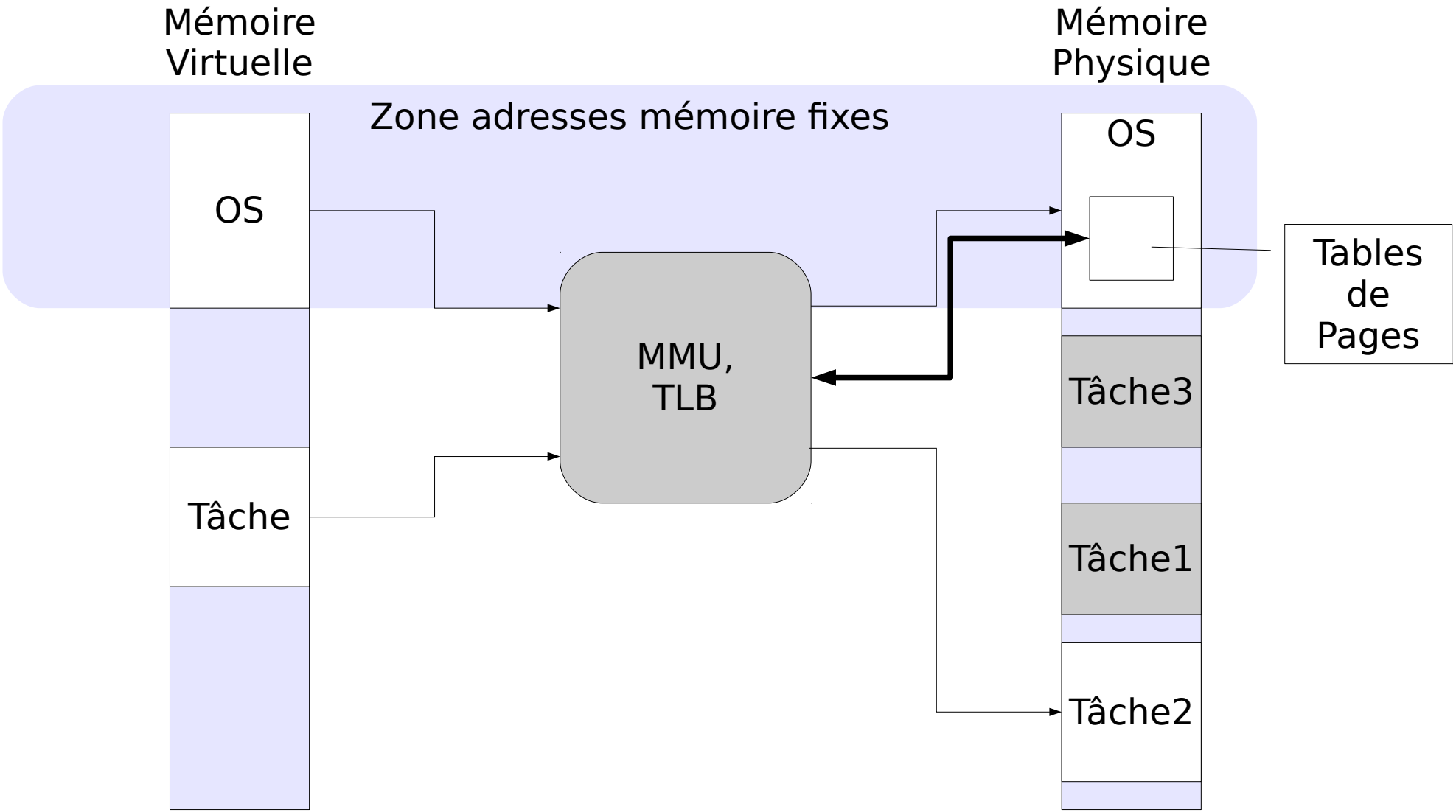
Exemple de "Mapping"



MMU et multi-tâches



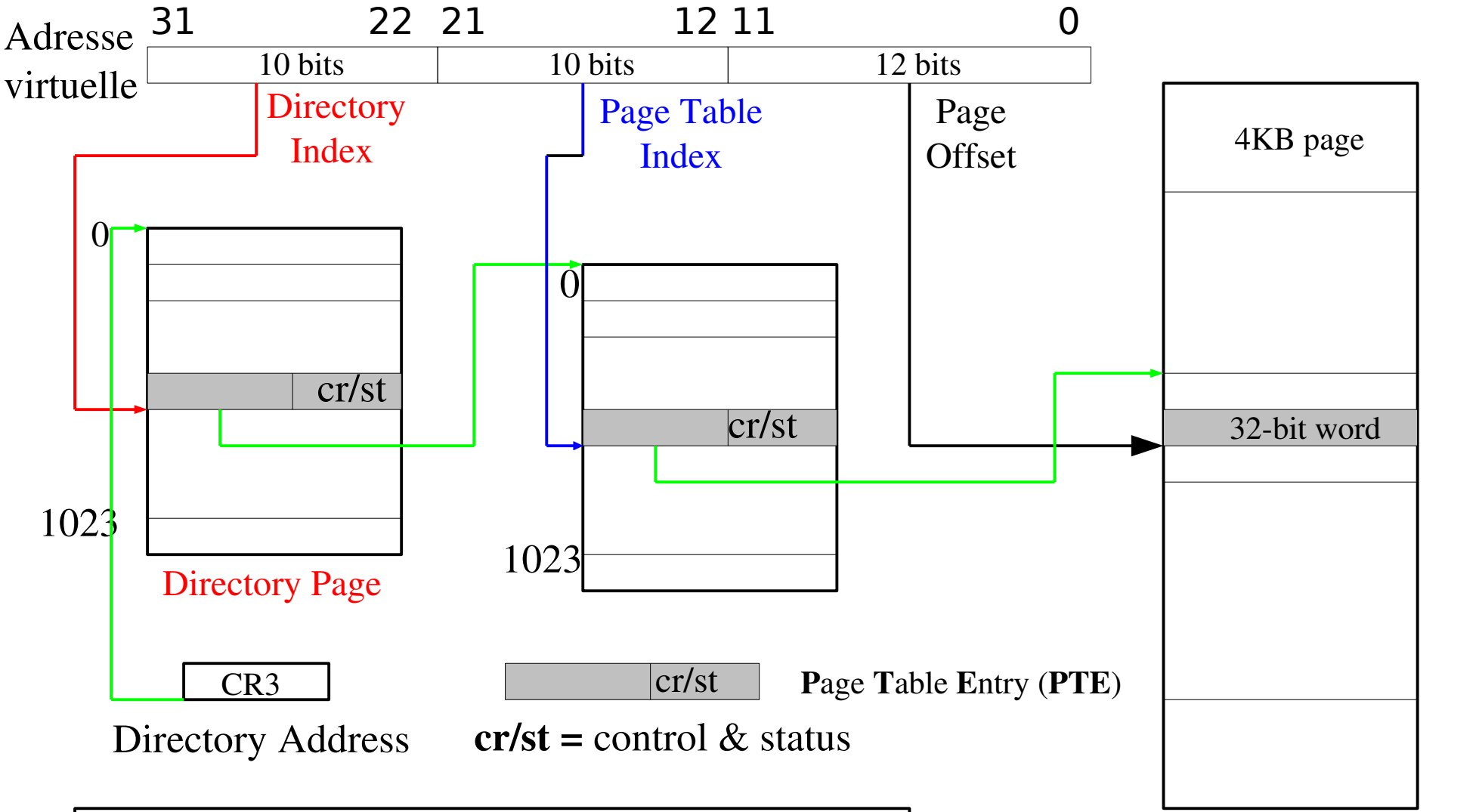
MMU, système et applications



Tables de Pages

- Conversion Virtuel => Physique
- Souvent, grandes zones non définies
  - => beaucoup de "PTE", donc de place mémoire, pour rien
- Comme souvent (i.e.: blocs de fichiers)
  - utilisation structure "arborescente" à plusieurs niveaux
  - ARM: 1 ou 2 niveaux (Level 1, Level 2)

MMU Intel


Adresse virtuelle
31
22
21
12
11
0
10 bits
10 bits
12 bits
Directory Index
Page Table Index
Page Offset
4KB page
0
cr/st
1023
Directory Page
0
cr/st
1023
32-bit word
CR3
Directory Address
cr/st
Page Table Entry (PTE)
cr/st = control & status
Translation Lookaside Buffer (TLB) = cache for PTEs
Mémoire physique

Page Table Entry

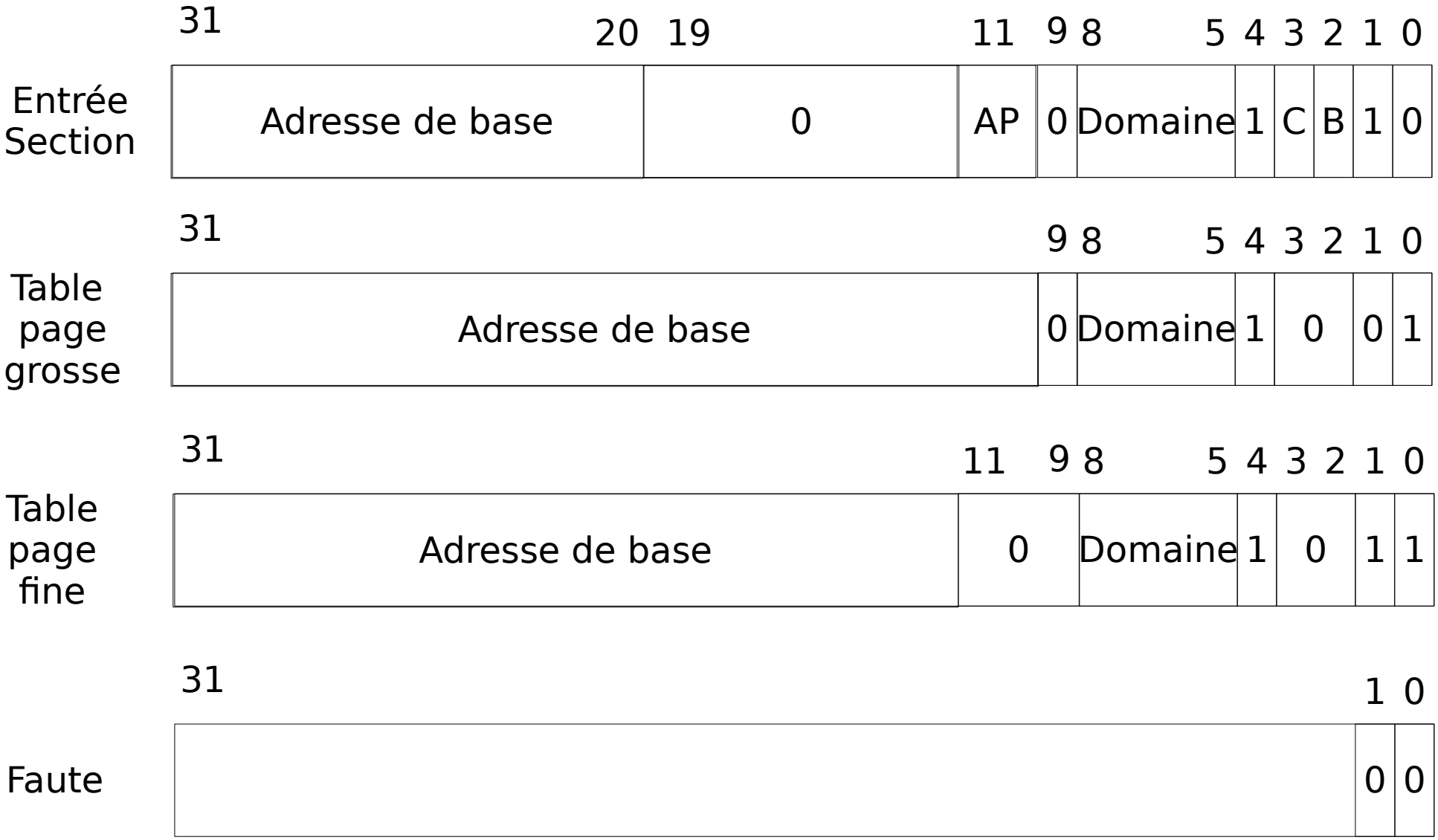
- Décrit une page d'un espace virtuel
- Adresse de la page physique associée, si valide
- Bits de contrôle
  - Page physique associée valide
  - Droits d'accès (lecture, écriture, exécution)
- Bits de status
  - Page accédée
  - Page modifiée

Level 1 PTE

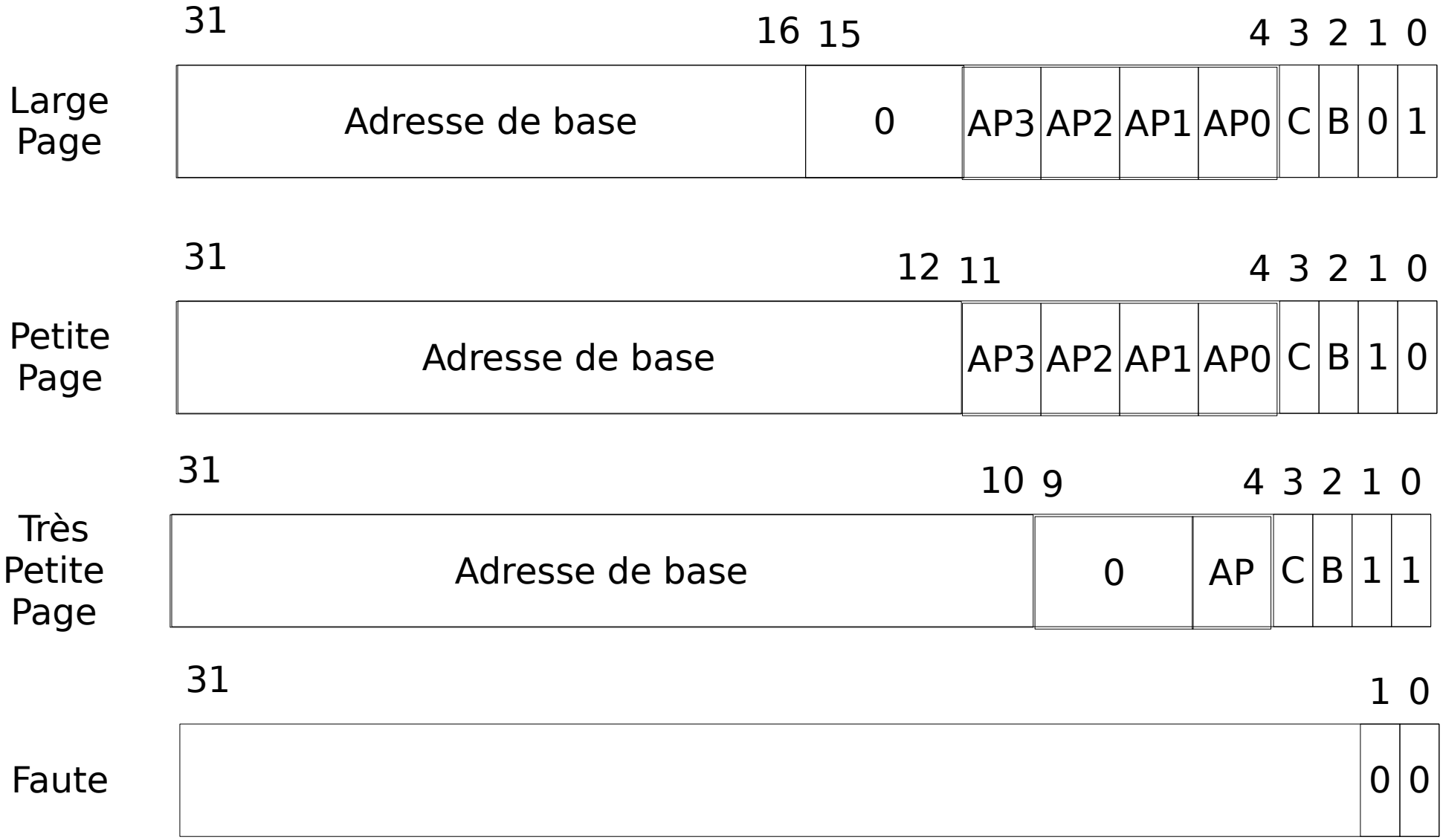
- 4096 entrées (=> 4GB) (taille 16 KB)
- Quatre types d'entrées:
  - Description de zone de 1MB
  - Entrée de catalogue vers une table de pages L2 fines

    (1024 * &, 4 ou 64 KB) (taille 4KB)
  - Entrée de catalogue vers une table de pages L2 grosses (256 * 4 ou 64 KB) (taille 1KB)
  - Entrée invalide générant un "abort"
- Chargement adresse de la table L1 dans registre C2 du co-processeur C15

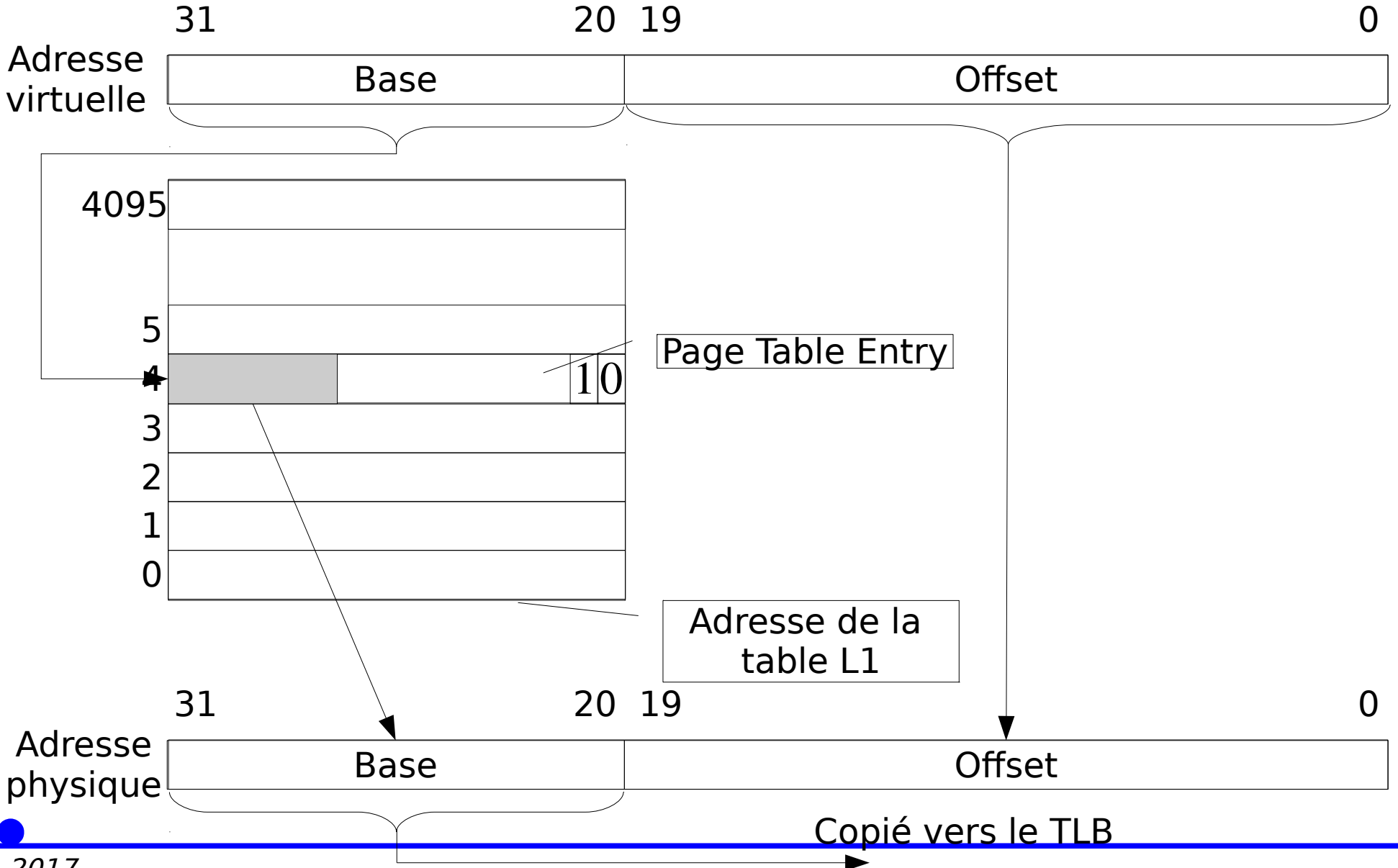
Level 1 PTE


31
20
19
11
9
8
5
4
3
2
1
0
Entrée Section
Adresse de base
0
AP
0
Domaine
1
C
B
1
0
31
9
8
5
4
3
2
1
0
Table page grosse
Adresse de base
0
Domaine
1
0
0
1
31
11
9
8
5
4
3
2
1
0
Table page fine
Adresse de base
0
Domaine
1
0
1
1
31
1
0
Faute
0
0

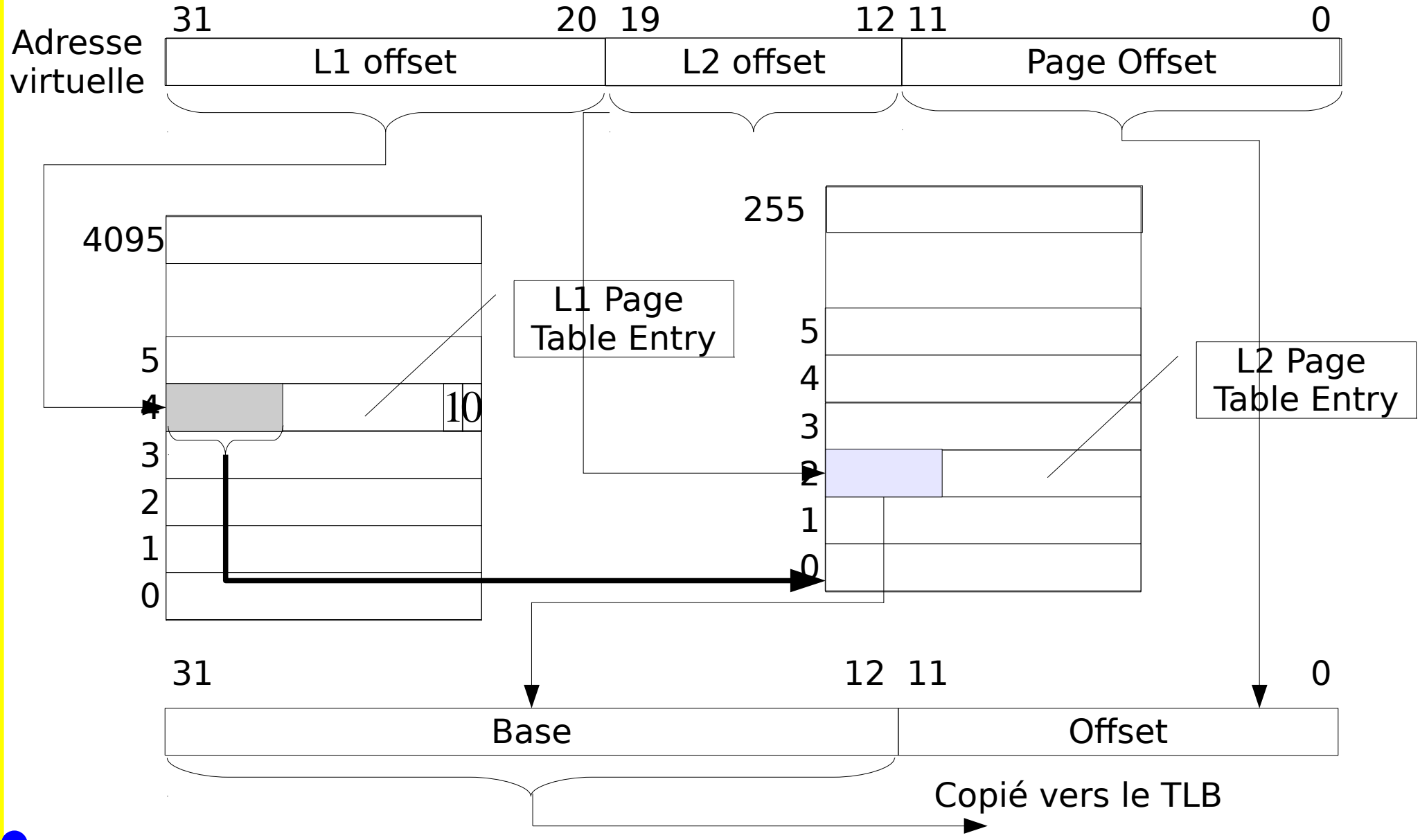
Level 2 PTE


31
16
15
4
3
2
1
0
Large Page
Adresse de base
0
AP3
AP2
AP1
AP0
C
B
0
1
31
12
11
4
3
2
1
0
Petite Page
Adresse de base
AP3
AP2
AP1
AP0
C
B
1
0
31
10
9
4
3
2
1
0
Très Petite Page
Adresse de base
0
AP
C
B
1
1
31
1
0
Faute
0
0

Conversion avec table L1



Conversion avec table L2



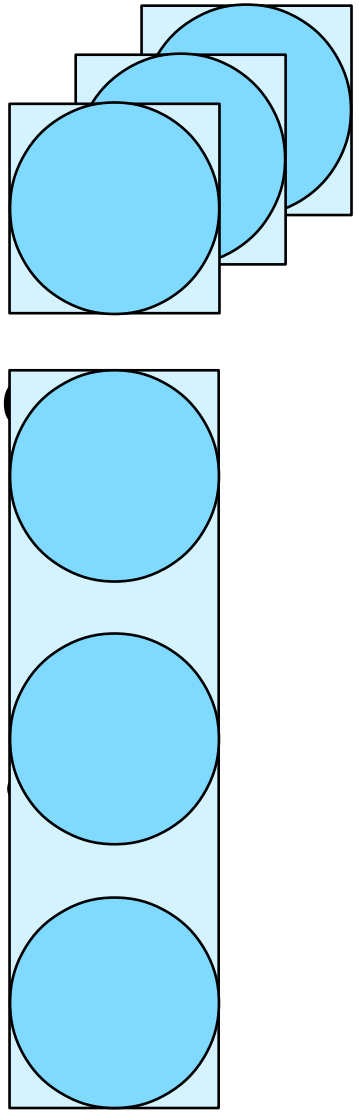
Memory Management (Chorus)

- Based on notion of **region**
  - address, size, access modes (R,W,X)
  - inheritance properties
- Configurable memory management support
  - **FL**at **M**emory (FLM)
  - **PR**otected **M**emory (PRM)
  - **V**irtual **M**emory (VM)
- Depends upon hardware support (MMU)
- Tradeoff performances/protection

Flat Memory (FLM)

- Single Supervisor Memory Space
- Unprotected
- Shared by microkernel and all actors
- Basic MMU support
  - to enable memory cache(s)
  - to setup non-cached memory regions (DMA)
  - 1-to-1 mapping
  - invalid address
    => unrecoverable system error

Protected Memory (PRM)



- Multiple user address spaces
- Mutually protected
- No lazy on-demand page allocati
- Invalid user-level address error
  - impacted to faulting thread
  - can be recovered by faulting applic
- Slightly impacts performances
  - system calls
  - context switches

Virtual Memory (VM)

- Includes PRM features
- Dynamic lazy physical page allocation
  - fill-zero option (“bss” region)
- Copy-on-write optimization
  - page inheritance (“init data” region)
- Optional page swapping
  - external swapper
  - swap space accounted in available memory