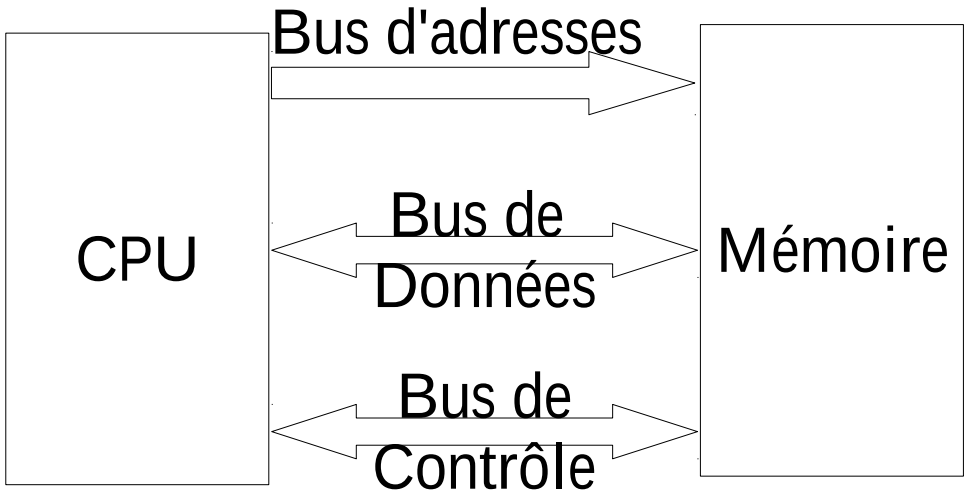
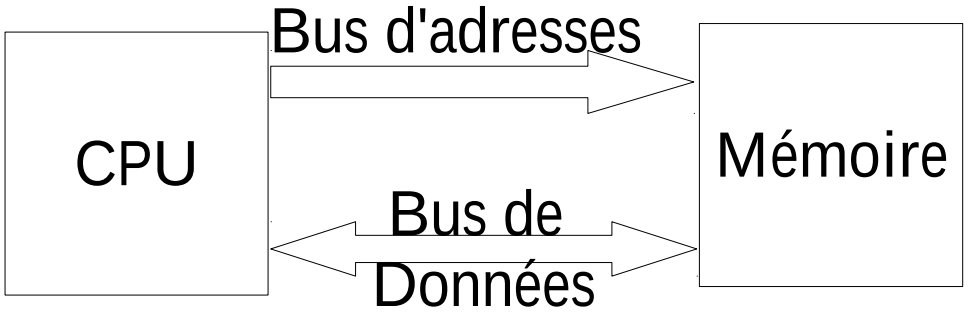
Informatique Embarquée
M2 / 2014

Gestion Mémoire

Interaction CPU / mémoire

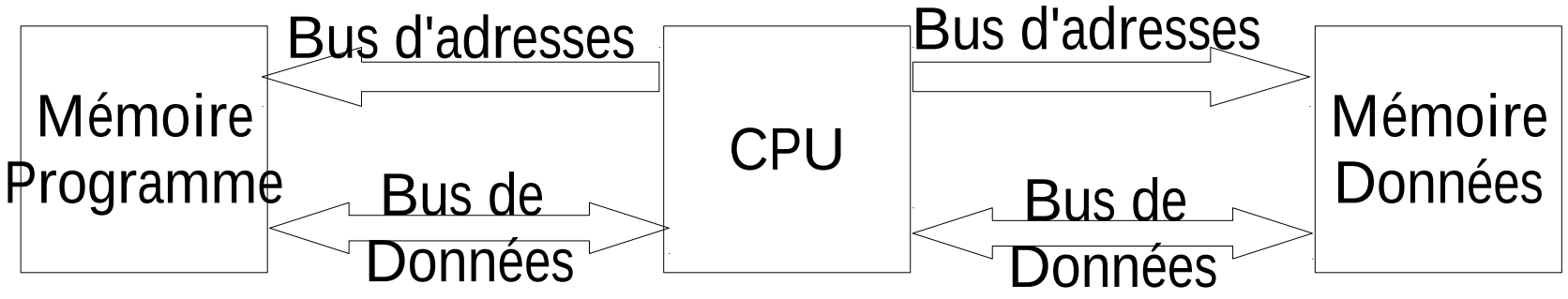


Architecture Von Neumann



- 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

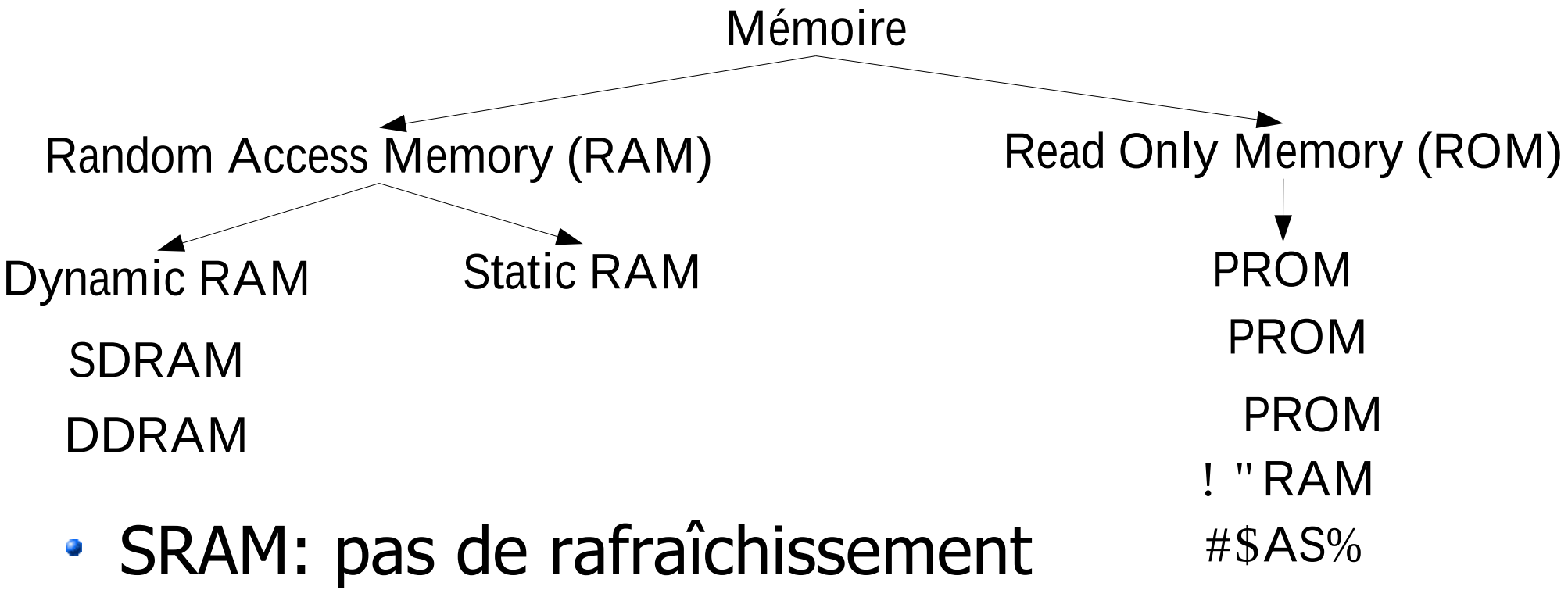


- Deux types de mémoire
  - Instructions et Données
  - Accès en parallèle aux instructions et aux données
    - Données accédées plus fréquemment que les instructions
- CPU idem à Von Neumann

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

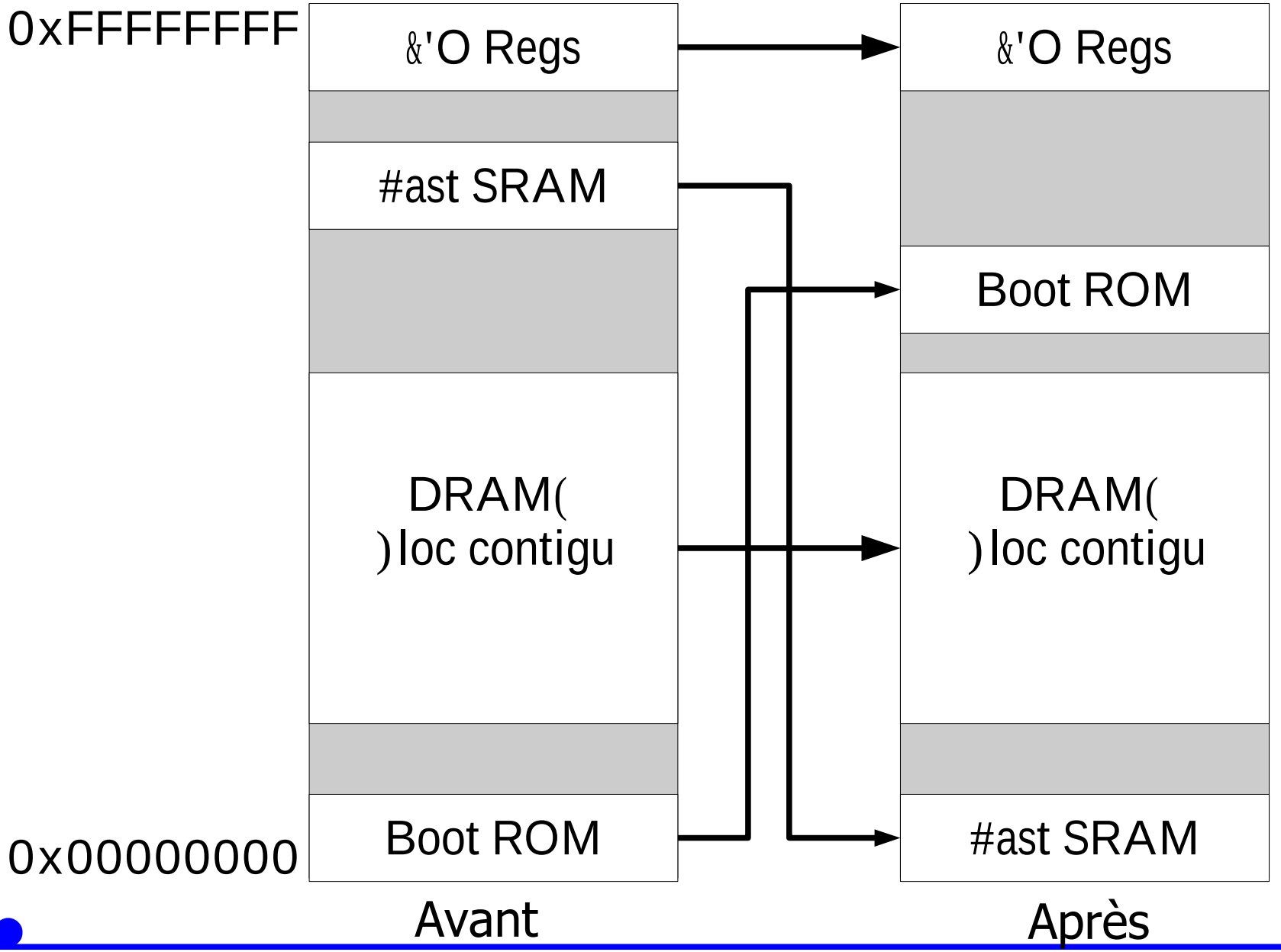


- SRAM: pas de rafraîchissement
- DRAM: rafraîchissement, Contrôleur

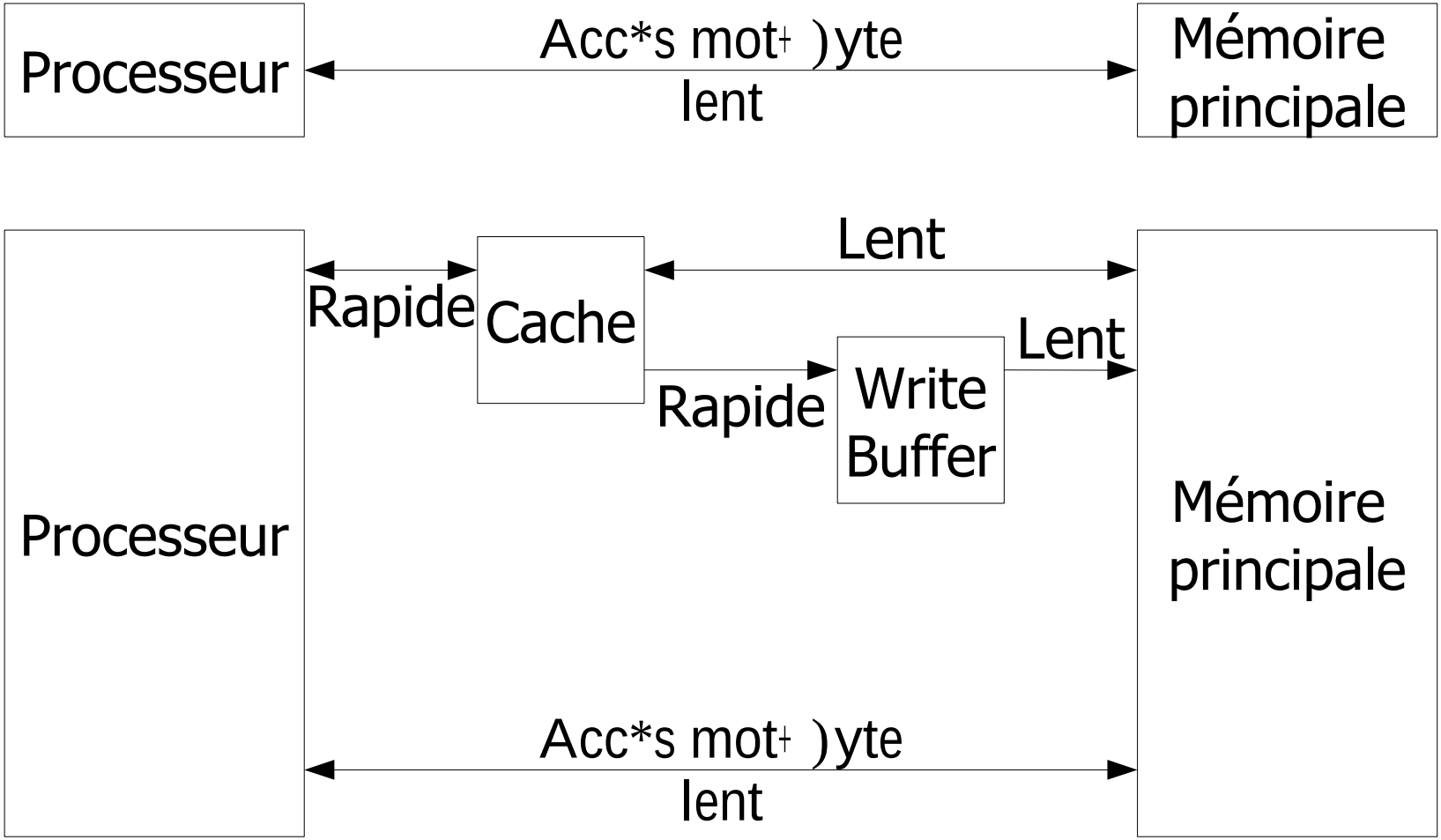
Initialisation (ARM)

- 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:
  - charge une "image" [compressée] de l'OS.

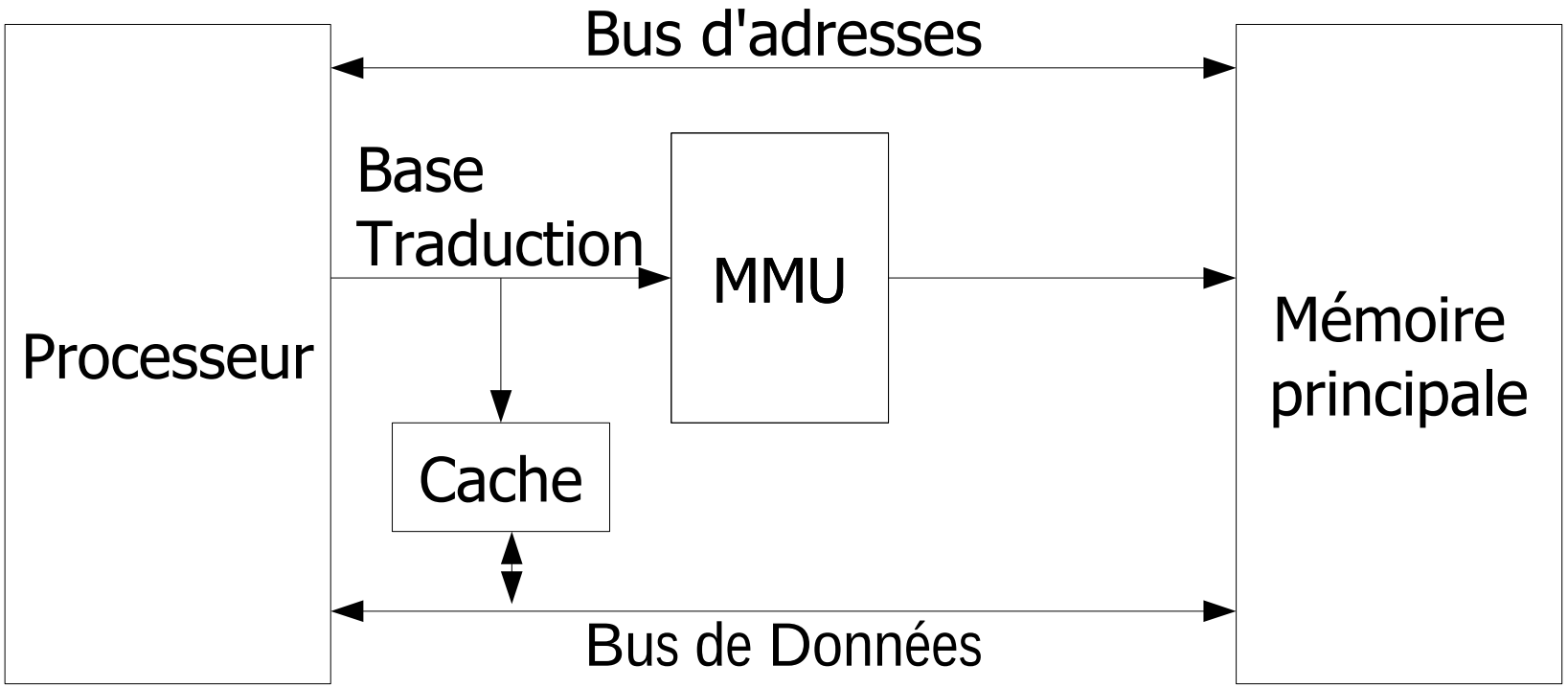
Réorganisation Mémoire



Caches Mémoire

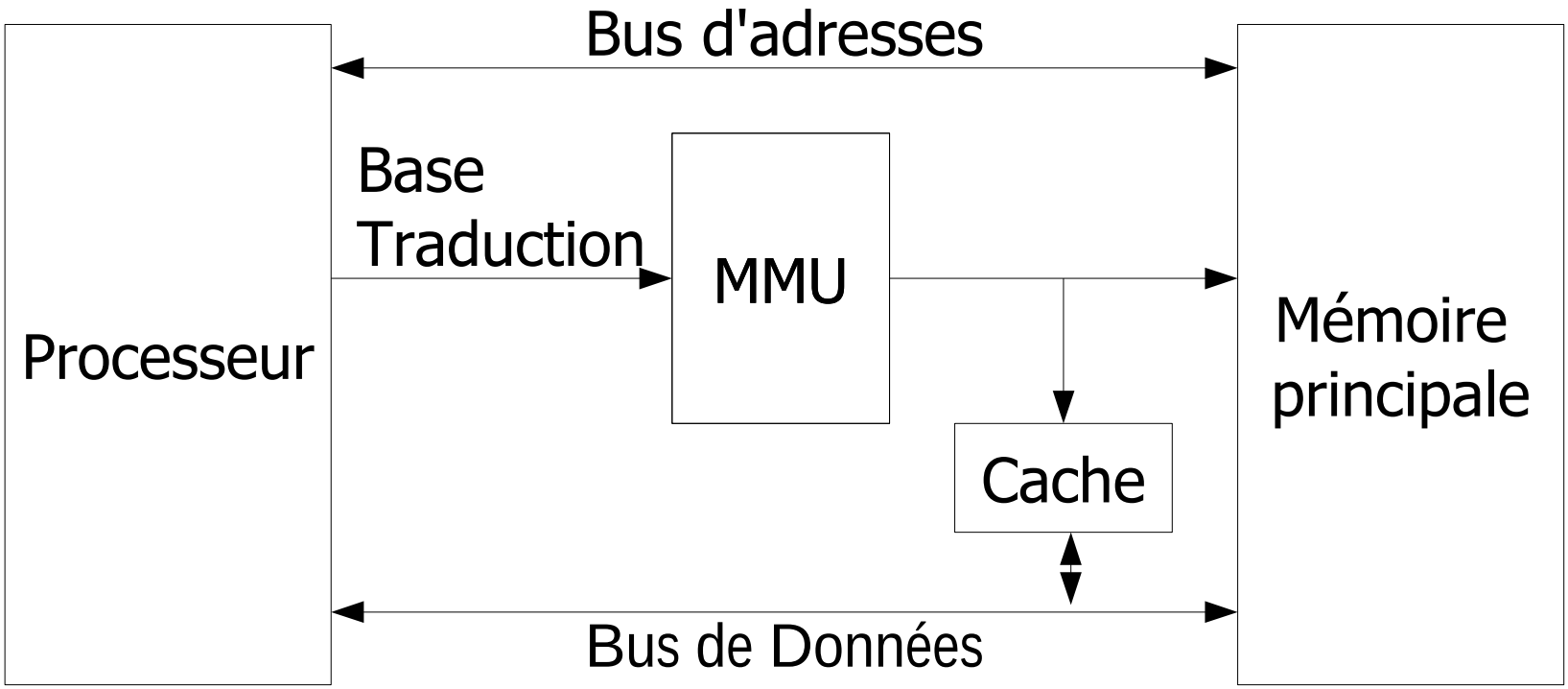


Cache et MMU



- Cache Logique

Cache et MMU

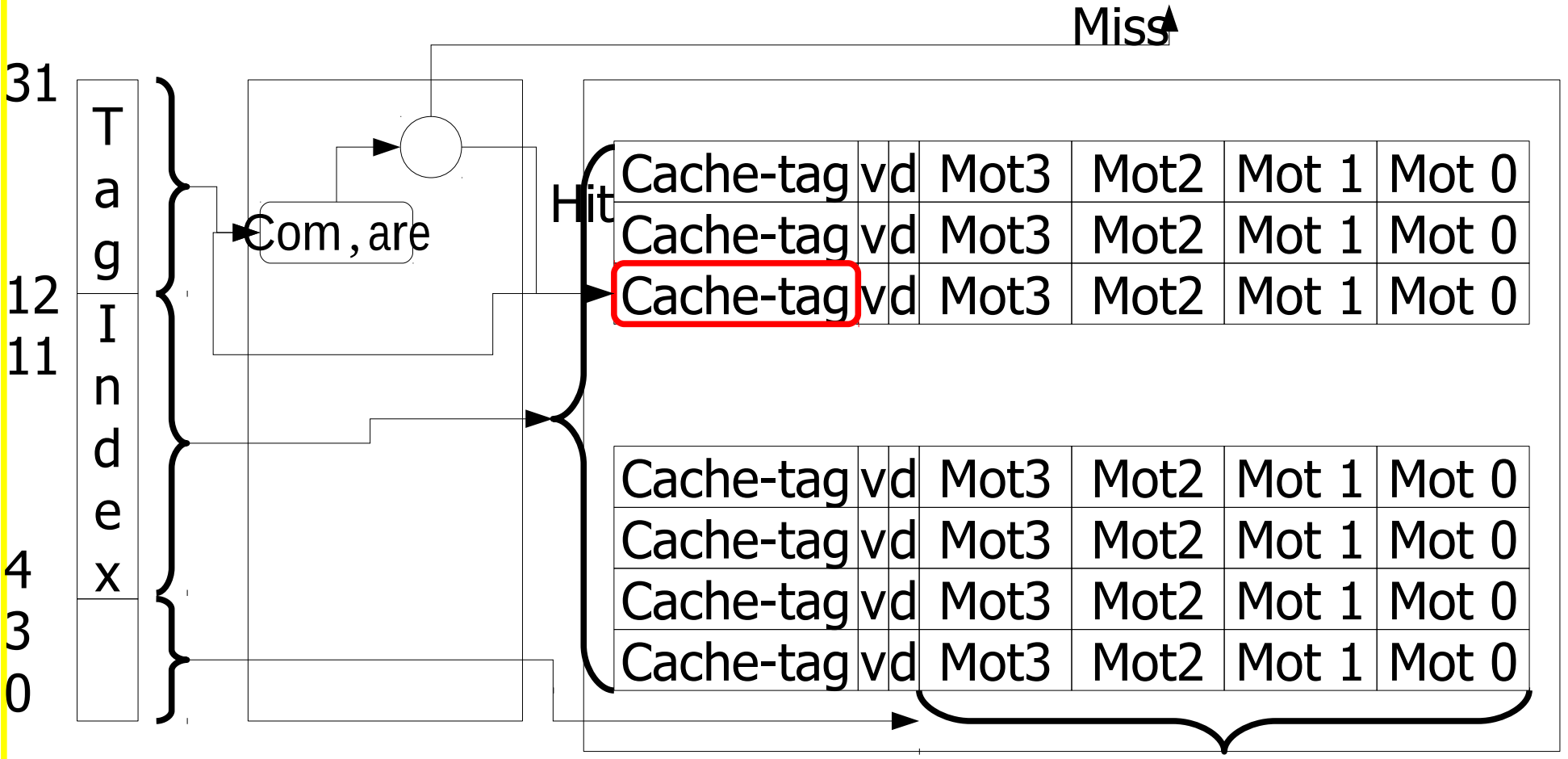


- Cache Physique

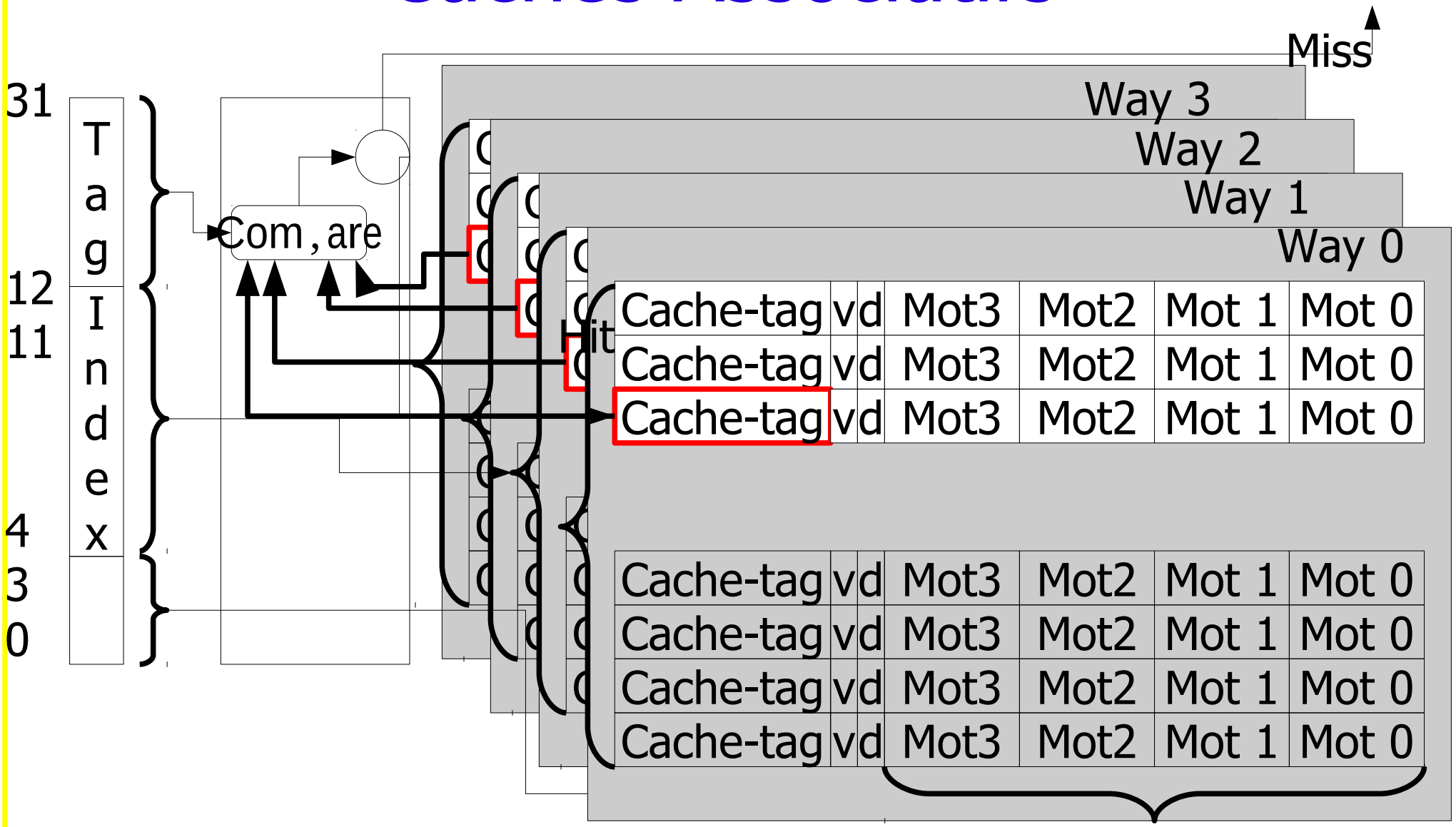
Cache

- 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.

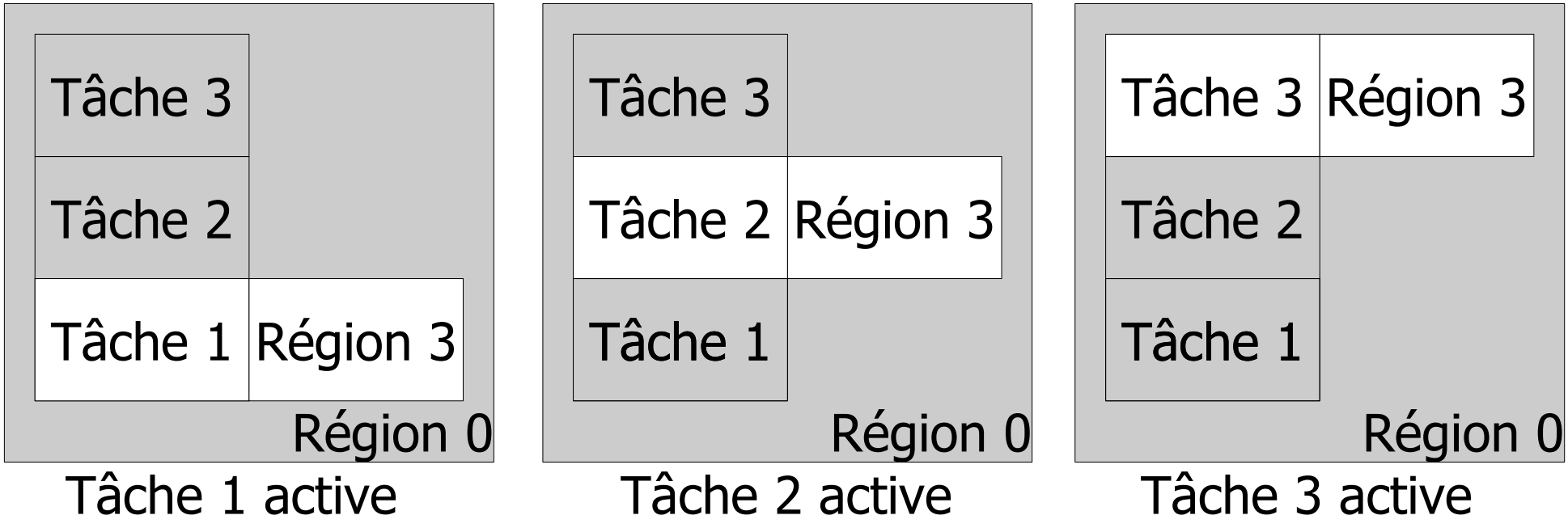
Memory Protection Unit

- 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

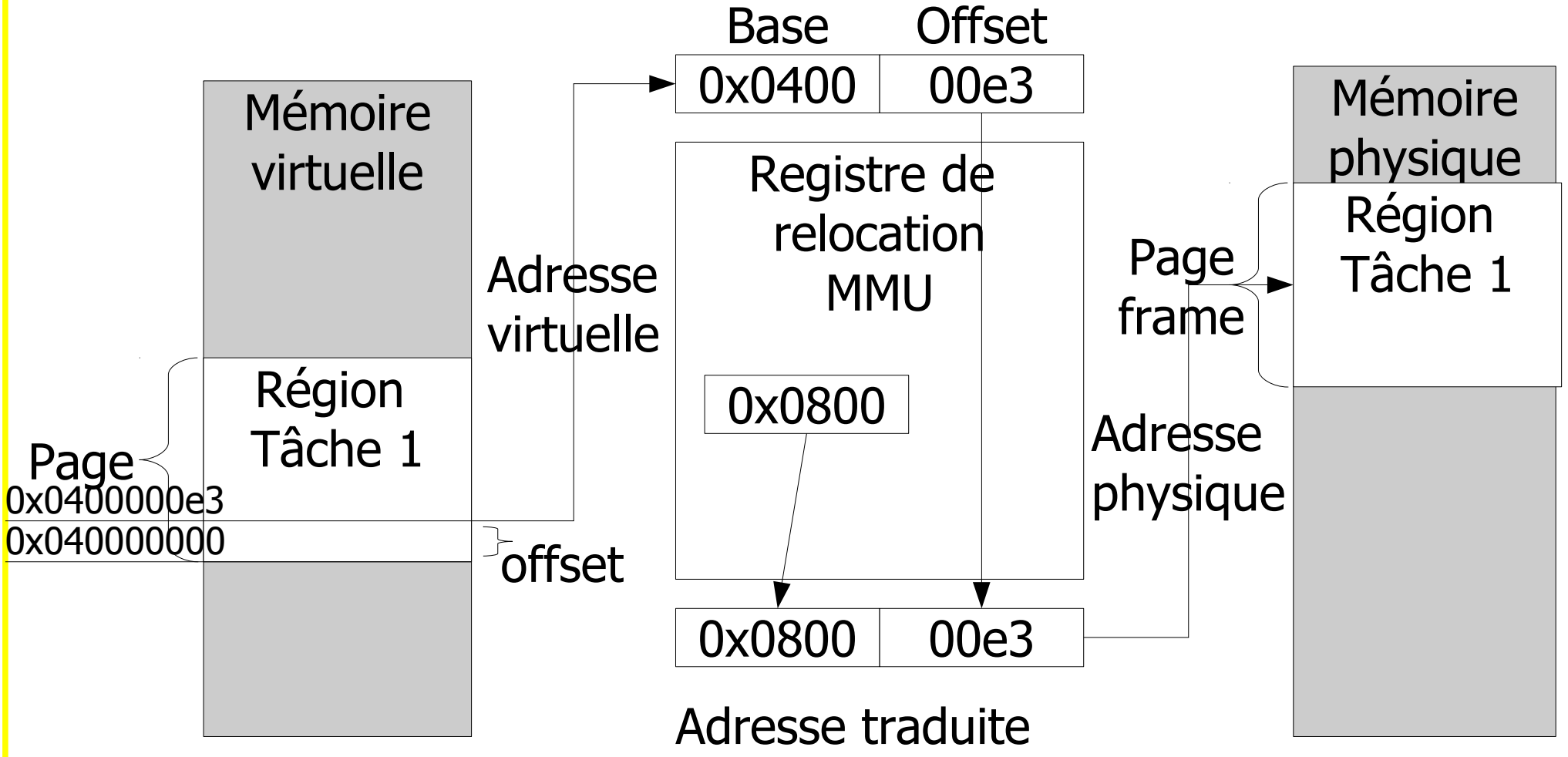
- Régions peuvent se recouvrir
- A chaque région est attribué une priorité, indépendante de ses privilèges
- En cas de recouvrement les droits de la région de plus haute priorité sont appliqués
- L'adresse de départ d'une région est un multiple de sa taille
- La taille d'une région est une puissance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

Exemple régions MPU



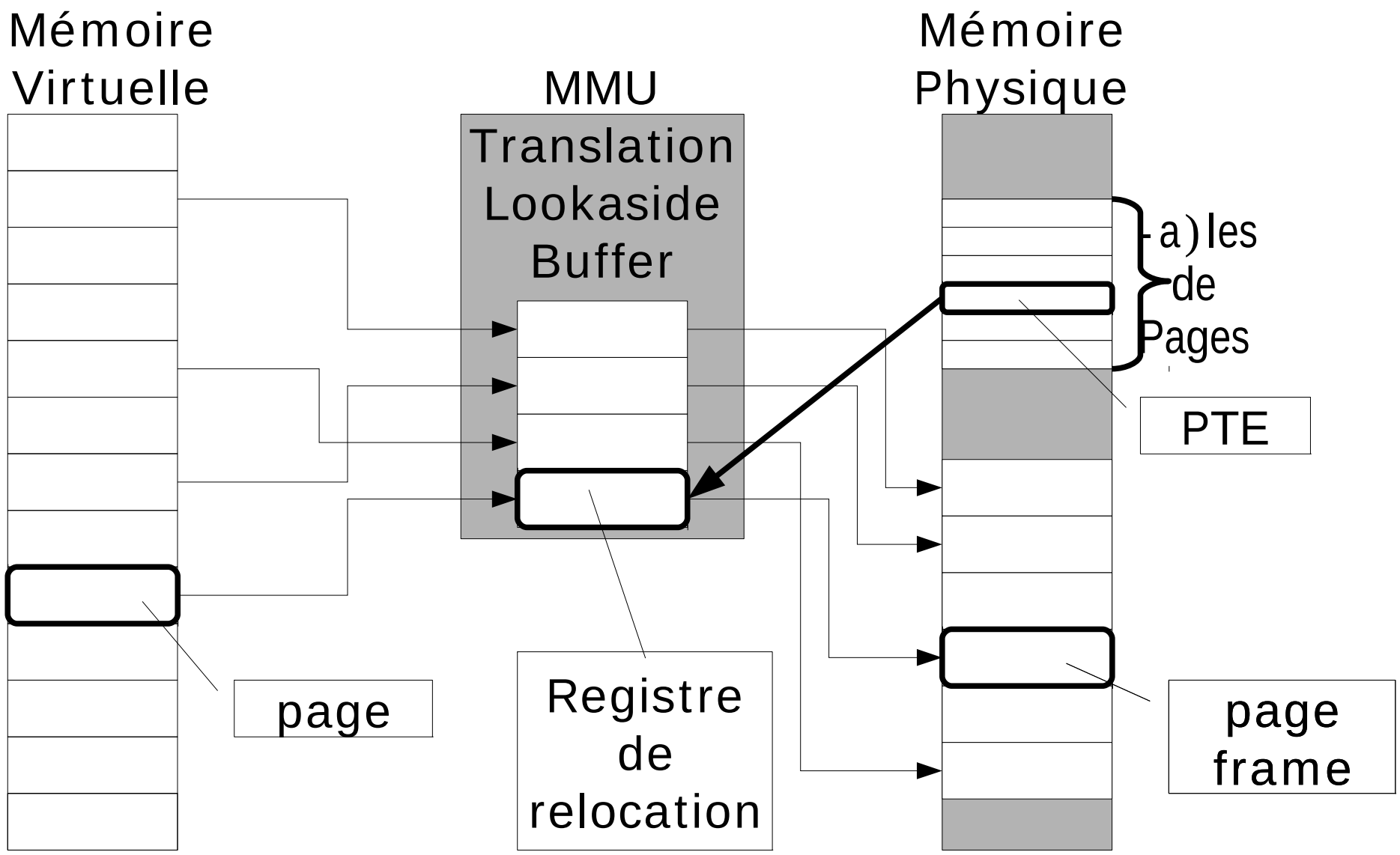
- Région 0: accès privilégié seulement
- Région 3: redéfinie à chaque changement de tâche, accès en mode utilisateur:
  - une tâche active ne peut pas corrompre les régions dont elle n'a pas besoin

Memory Management Unit

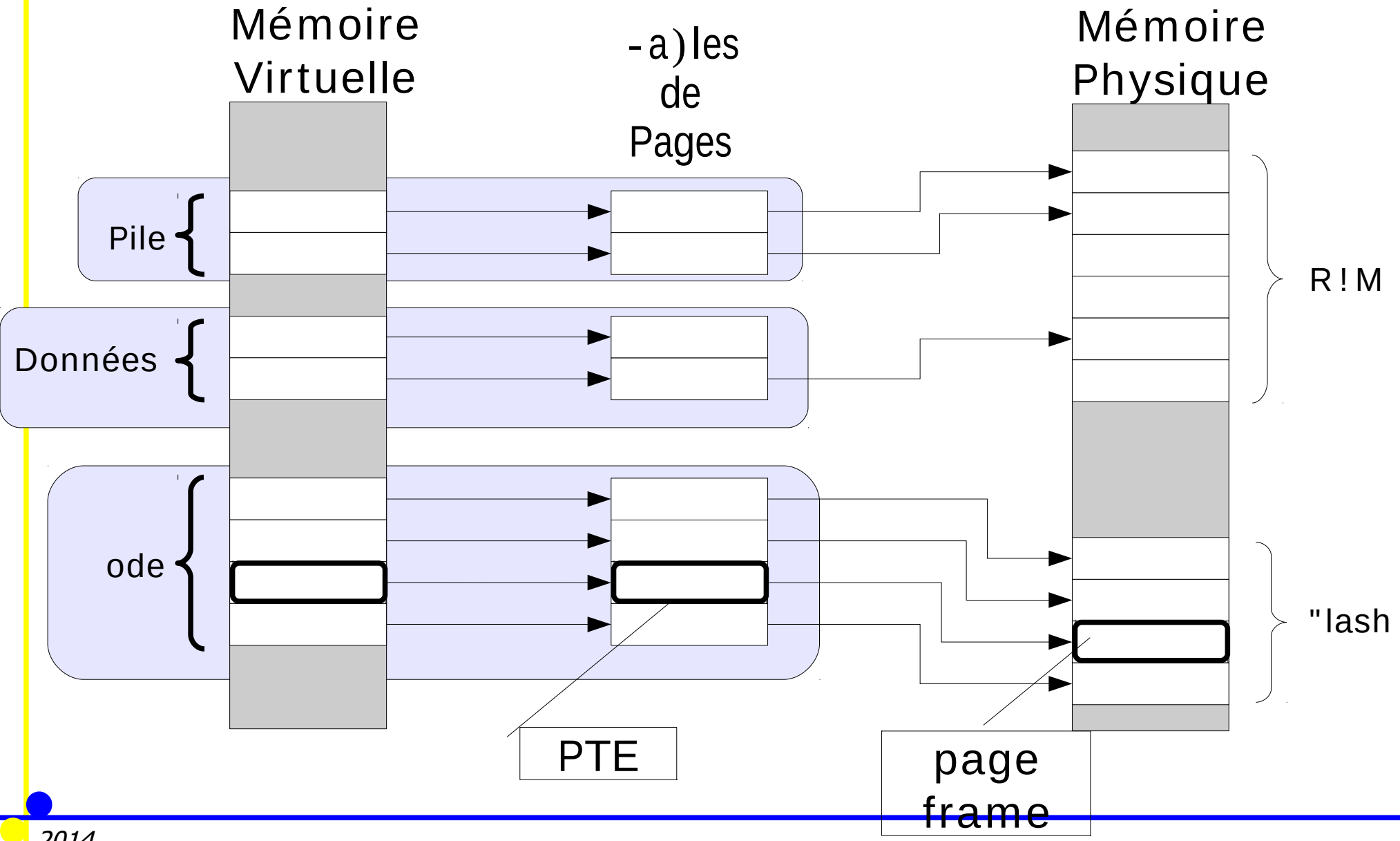


- Une autre tâche avec la même adresse virtuelle accédera une page physique différente

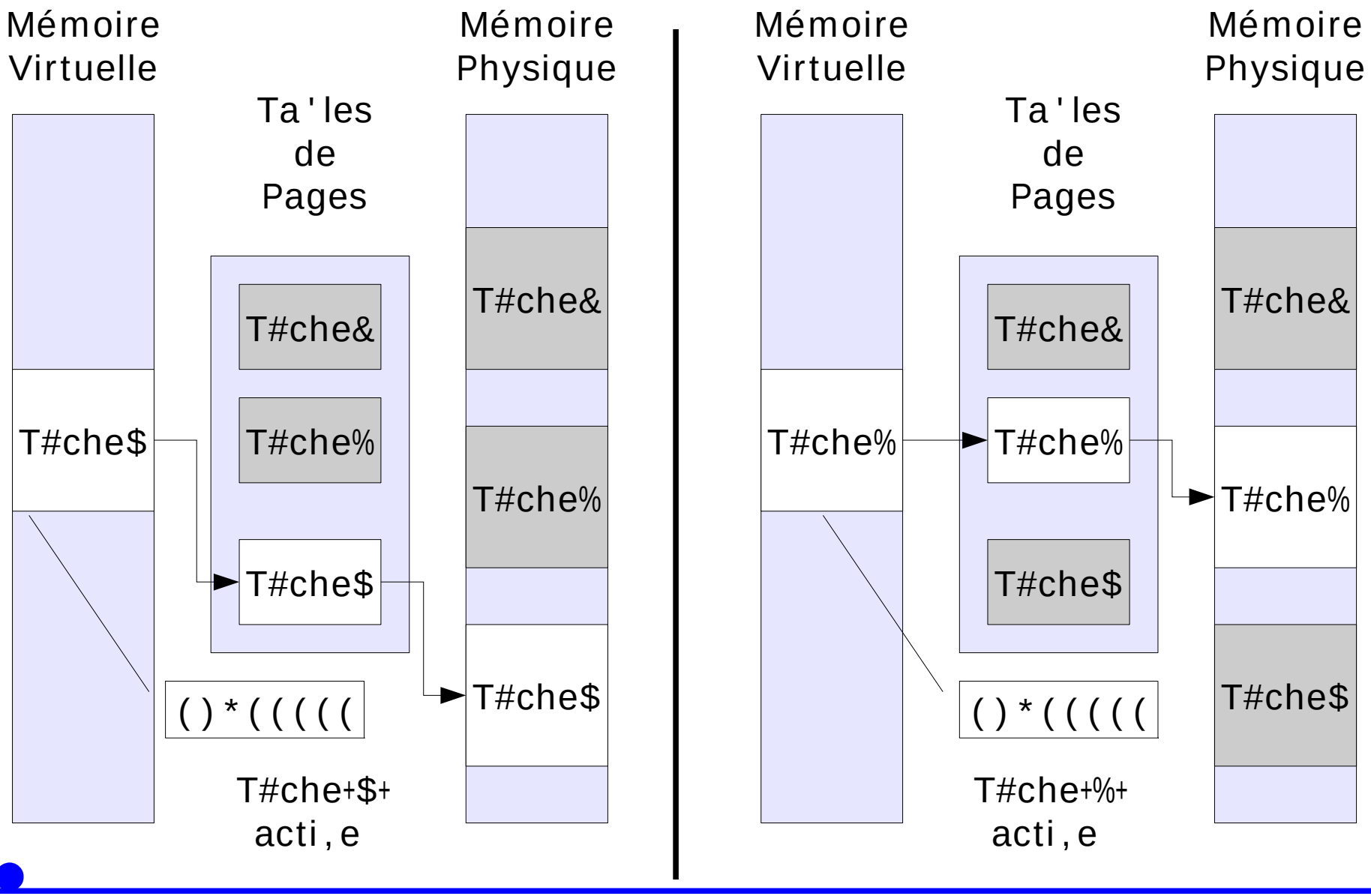
Mémoire Virtuelle: composants



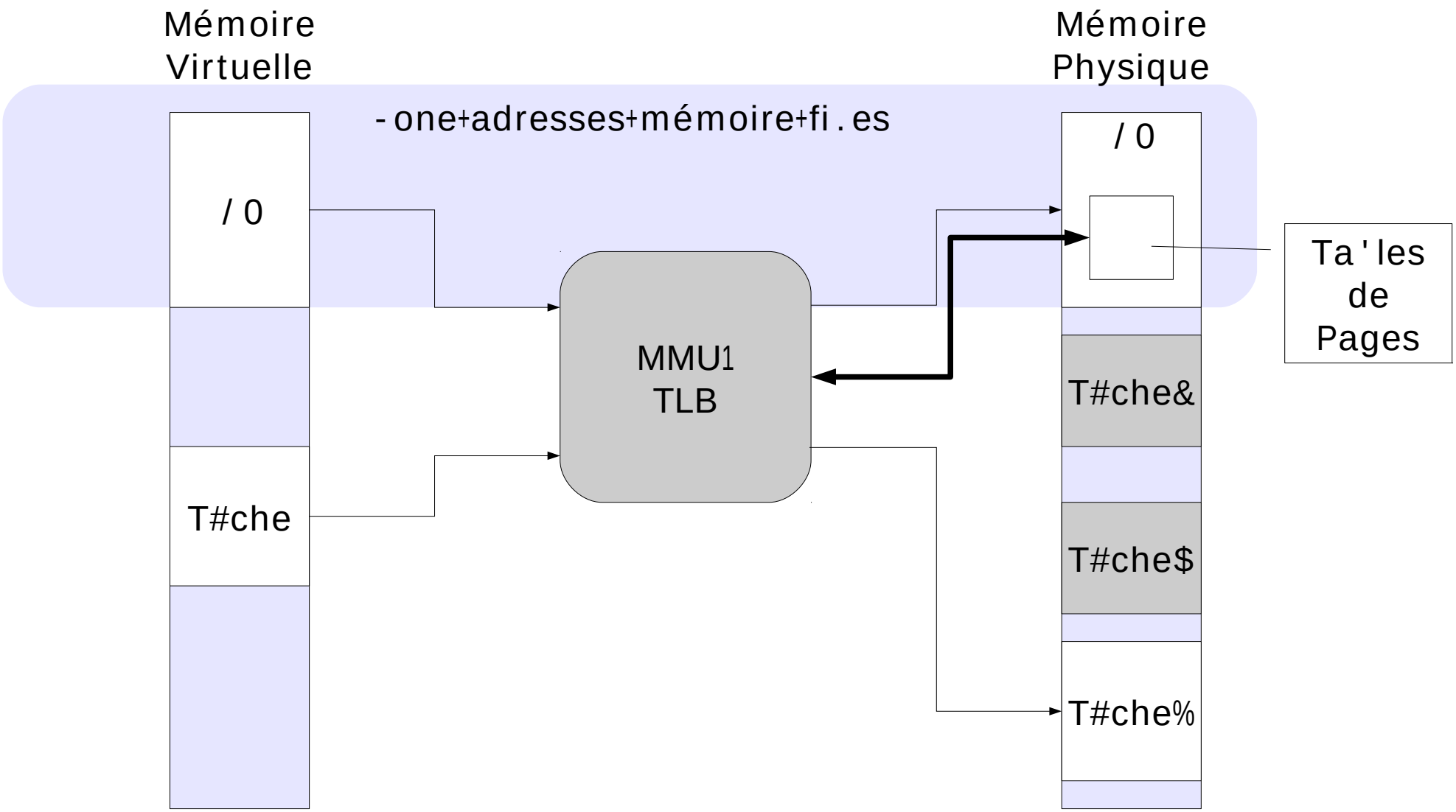
Exemple de "Mapping"



MMU et multi-tâches



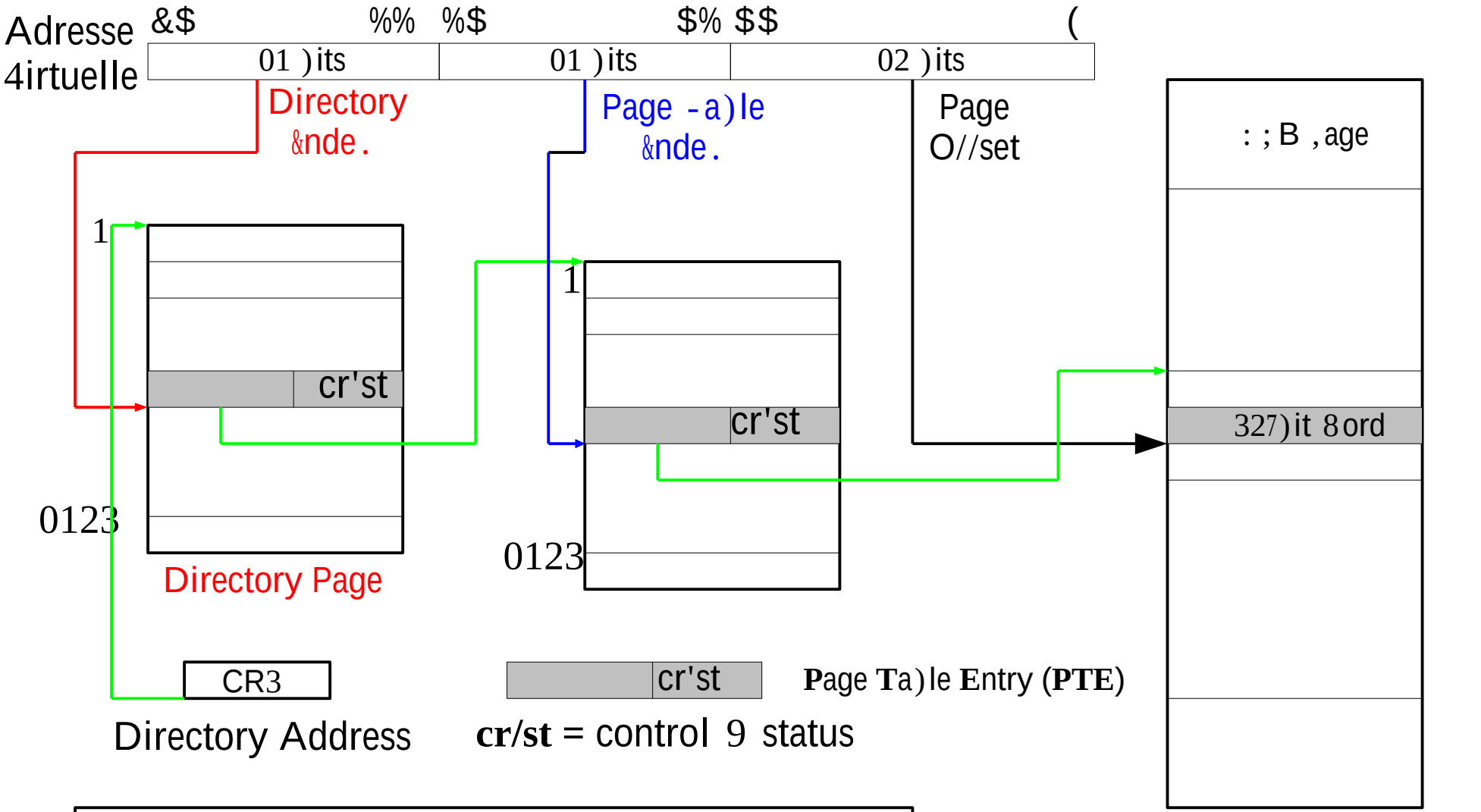
MMU, système et applications



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 4irtuelle
&$ %% %$ $% $$ (
01)its
01)its
02)its
Directory &nde.
Page -a)le &nde.
Page O//set
: ; B ,age
1
cr'st
1
cr'st
327)it 8ord
0123
0123
Directory Page
CR3
cr'st
Page Ta)le Entry (PTE)
Directory Address
cr/st = control 9 status
Translation Loo<aside Bu//er (TLB) = cac5e /or P- s
Mémoire ,5ysi6ue

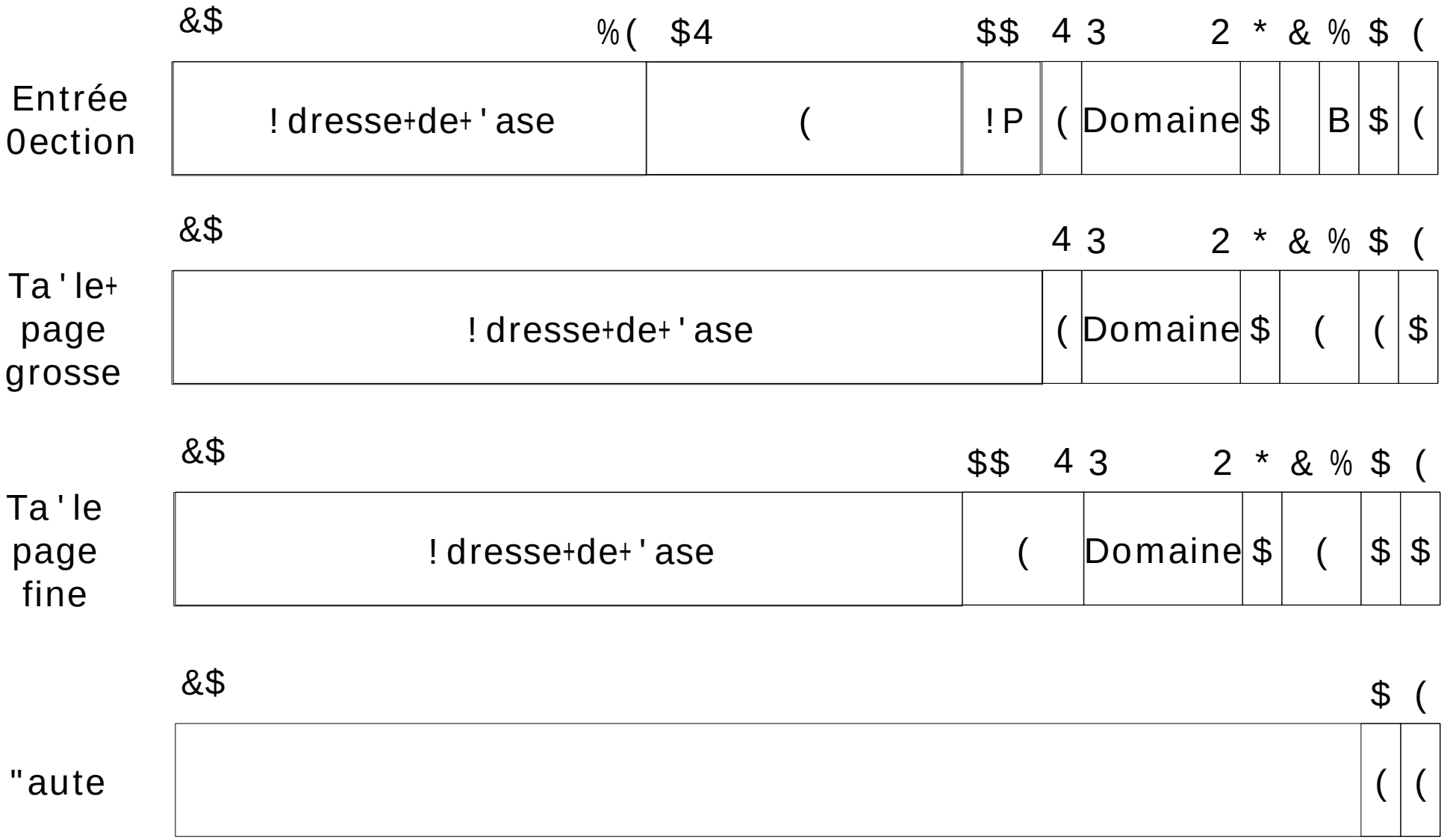
Page Table Entry

- 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

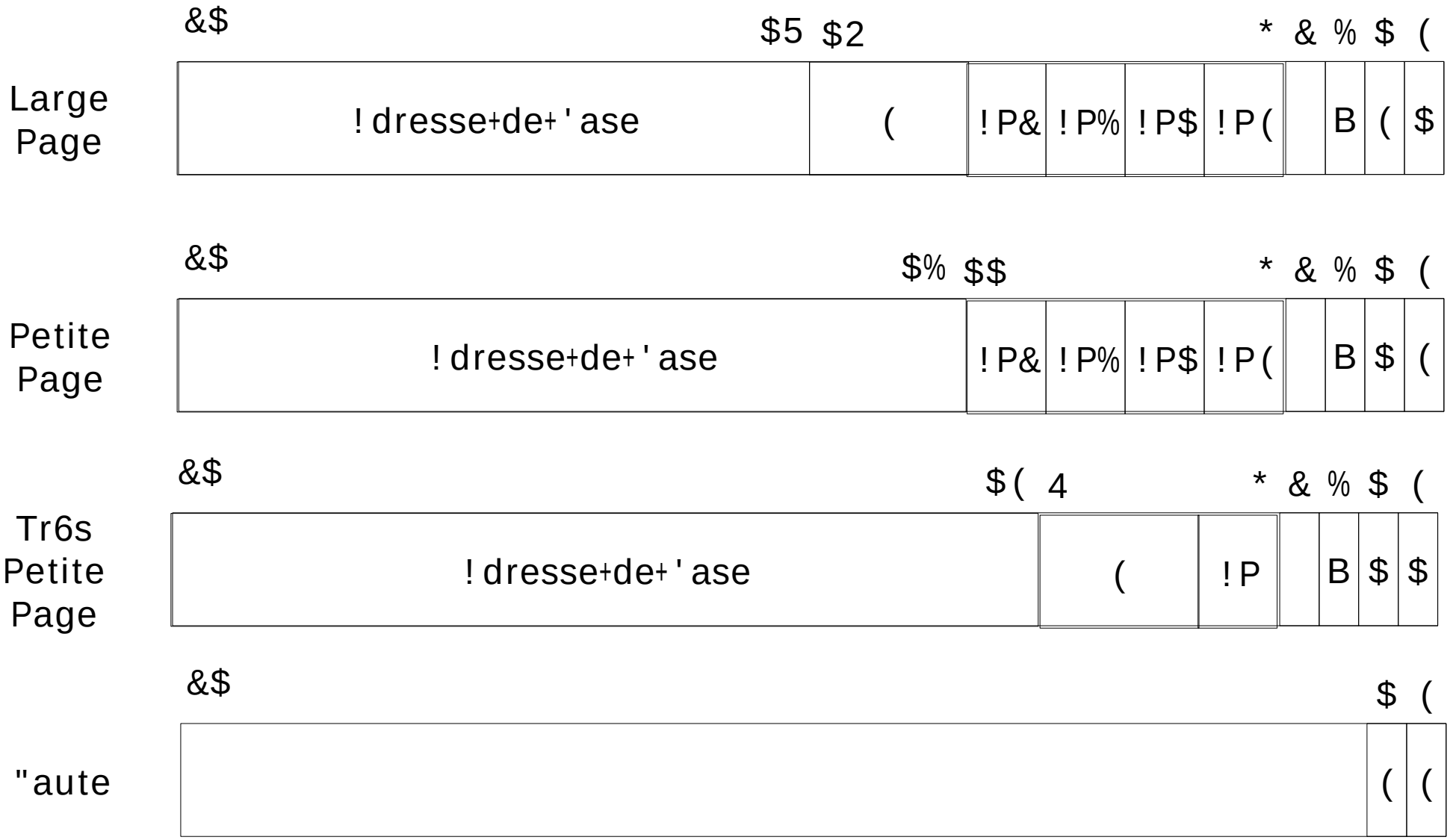
Level 1 PTE

- 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

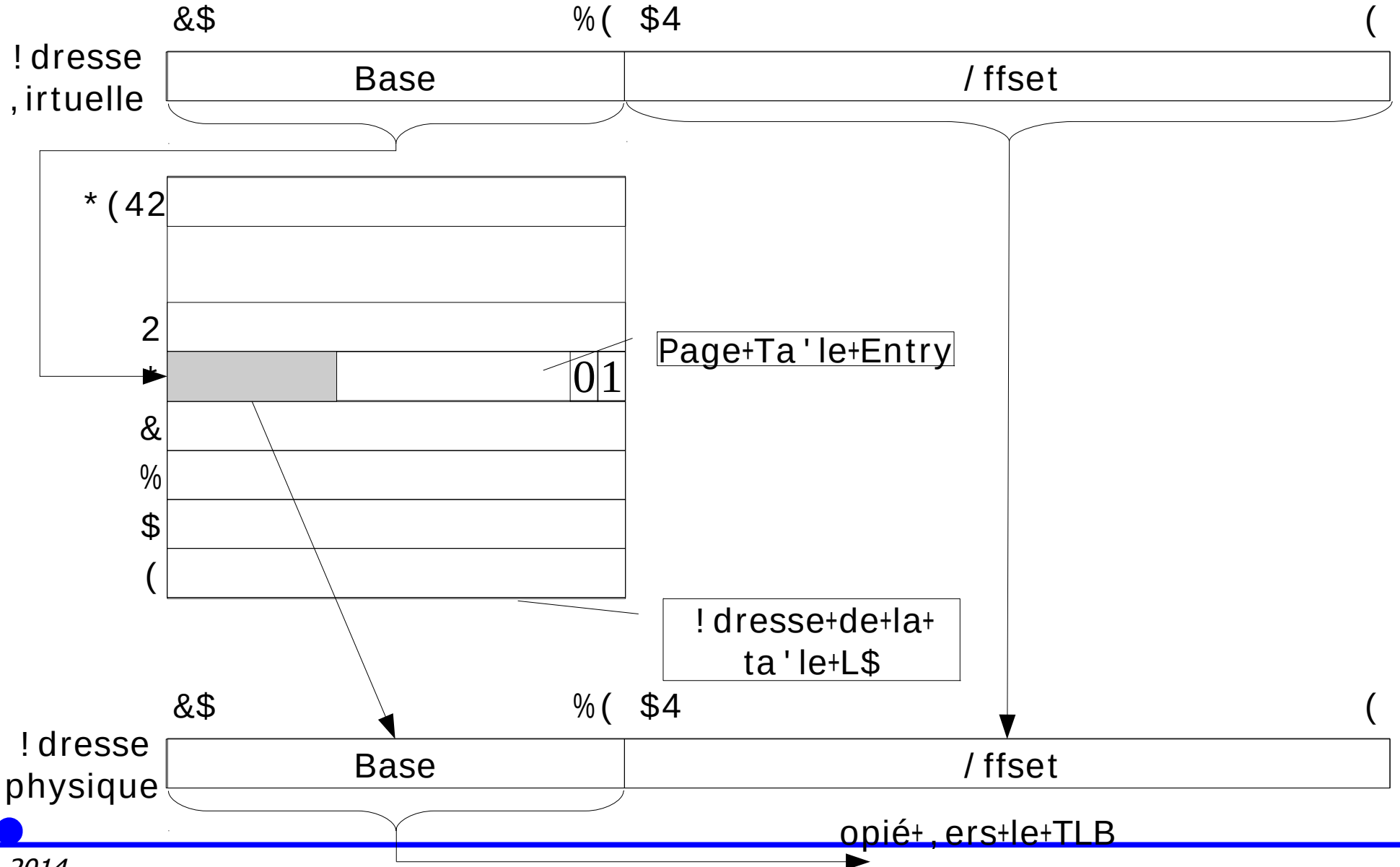
Level 1 PTE


&$
%($4
$$ 4 3 2 * & % $ (
Entrée 0ection
! dresse+de+ ' ase
(
! P
(
Domaine
$
B
$
(
&$
4 3 2 * & % $ (
Ta ' le+ page grosse
! dresse+de+ ' ase
(
Domaine
$
(
(
$
&$
$$ 4 3 2 * & % $ (
Ta ' le page fine
! dresse+de+ ' ase
(
Domaine
$
(
$
$
&$
$ (
"aute
((

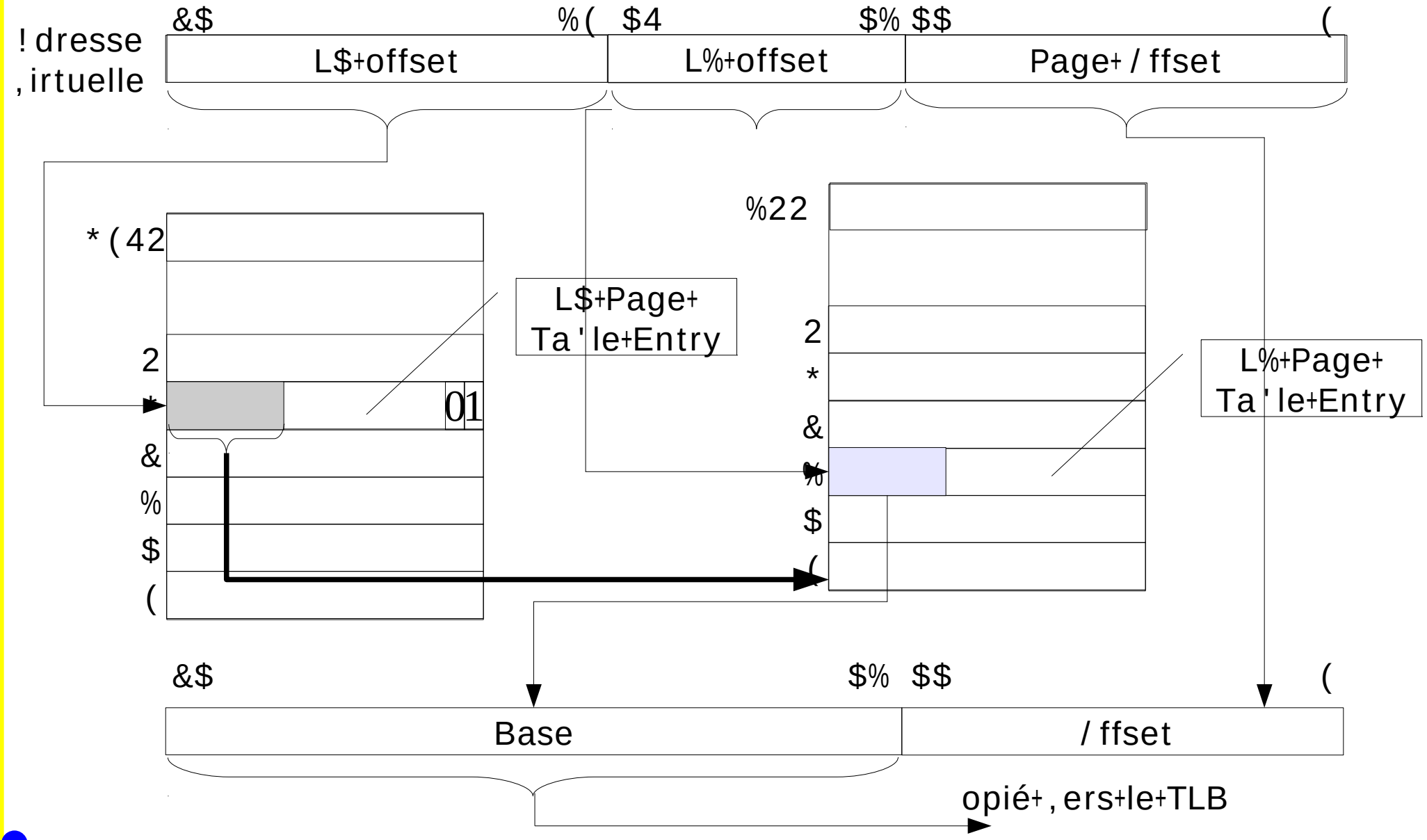
Level 2 PTE


&$
$5 $2
* & % $ (
Large Page
!dresse+de+'ase
(
!P&
!P%
!P$
!P(
B
($
&$
$% $$
* & % $ (
Petite Page
!dresse+de+'ase
!P&
!P%
!P$
!P(
B
$ (
&$
$(4
* & % $ (
Tr6s Petite Page
!dresse+de+'ase
(
!P
B
$ $
&$
$ (
"aute
((

Conversion avec table L1

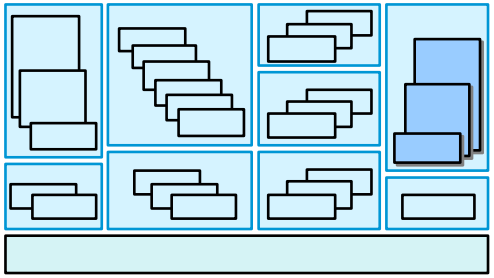


Conversion avec table L2



Memory Management

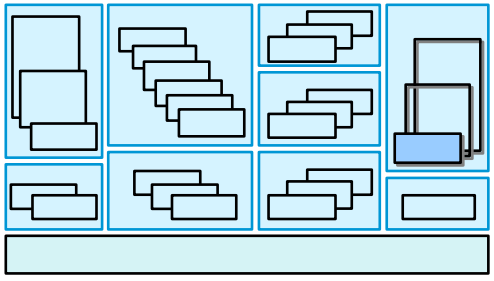
- Based+on+notion+of+**region**
  - address1+si7e1+access+modes+8R1 9 1) :
  - inheritance+properties
- onfigura ' le+memory+management+support
  - **FL**at **M**emory+8"LM:
  - **PR**otected+**M**emory+8PRM:
  - **V**irtual+**M**emory+8VM:



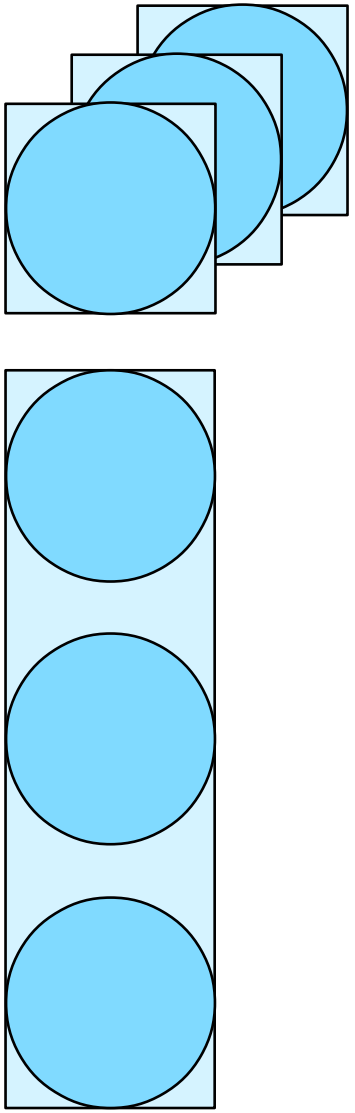
- Depends+upon+hard ; are+support+8MMU:
- Tradeoff+performances<protection

Flat Memory (FLM)

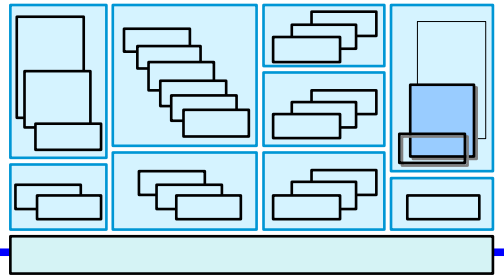
- 0ingle+0uper,isor+Memory+0pace
- Unprotected
- 0hared+'y+microkernel+and+all+actors
- Basic+MMU+support
  - to+ena'le+memory+cache8s:
  - to+setup+non=cached+memory+regions+8DM!:
  - $=to=$+mapping
  - in,alid+address
    >?+unreco,era'le+system+error



Protected Memory (PRM)



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



Virtual Memory (VM)

- @ncludes+PRM+features
- Dynamic+la7y+physical+page+allocation
  - fill=7ero+option+8A'ssB+region:
- opy=on= ; rite+optimi7ation
  - page+inheritance+8Ainit+dataB+region:
- / ptional+page+s ; apping
  - e.ternal+s ; apper
  - s ; ap+space+accounted in+a,aila'le+memory