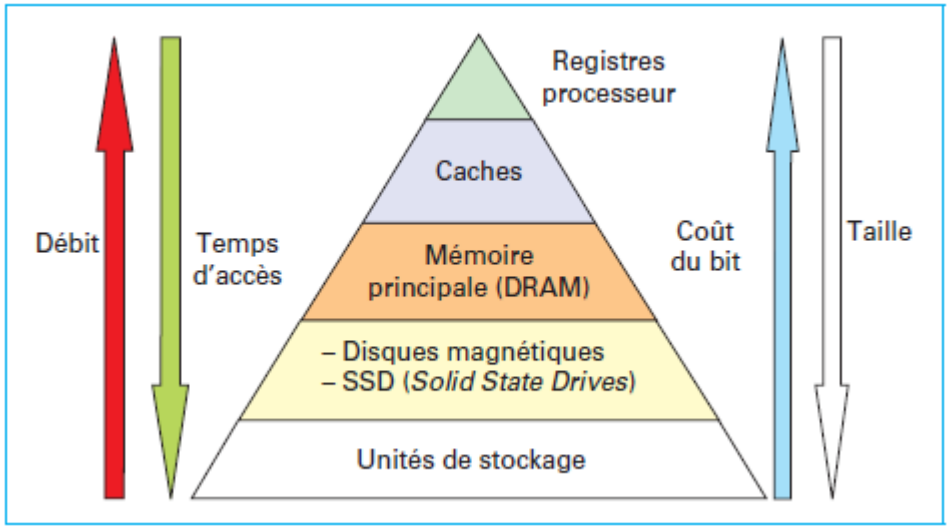
Le principe de hiérarchie mémoire : les caches
Les mémoires de l'ordinateur



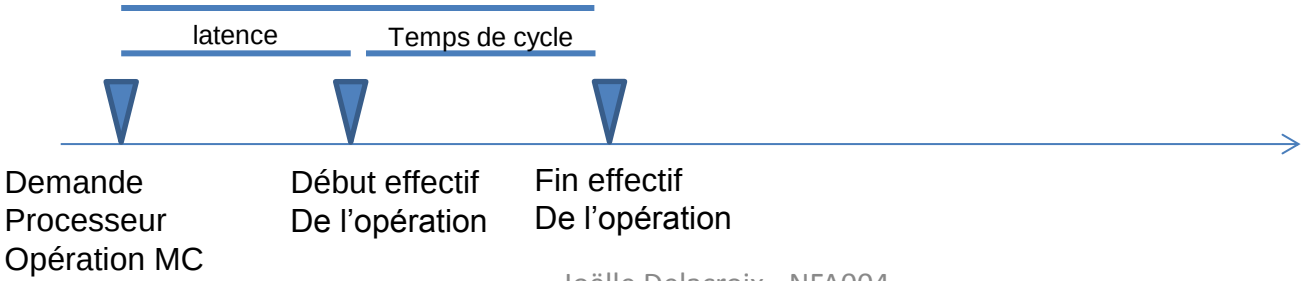
Techniques de l'ingénieur, H1002, hiérarchie mémoire : les caches

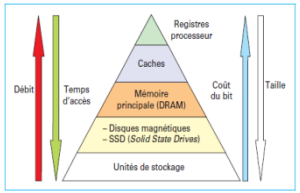
Les mémoires de l'ordinateur

- Une « **mémoire** » est un composant électronique capable de stocker temporairement des informations.

- Une mémoire est caractérisée par :
  - Sa **capacité**, représentant le volume global d'informations (en bits) que la mémoire peut stocker (par exemple 1 Goctets, soit 2^{30} octets, soit $2^{30} * 8$ bits.
  - Sa **latence** correspondant au temps qui sépare la demande de la donnée et le début effectif de l'opération.
  - son **temps de cycle** correspondant au temps qui sépare le début effectif de l'opération de lecture/ écriture et sa fin
  - Son **temps d'accès**, correspondant à l'intervalle de temps entre la demande de lecture/écriture et la disponibilité de la donnée

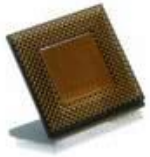
    Temps d'accès = latence + temps de cycle.





Les mémoires de l'ordinateu

L'ordinateur contient différents niveaux de mémoire, organisés selon une **hiérarchie mémoire**.



REGISTRES
N bits (32, 64)
1 nanoseconde

Mémoires Caches
Koctets
5 nanosecondes

Mémoires Centrales
Goctets
10 nanosecondes

Mémoires de masse
100 - 200 Goctets
5 millisecondes

Mémoires internes : mémoires **volatiles**

Mémoires de stockage : mémoires **permanentes**

PLUS GRANDE CAPACITE

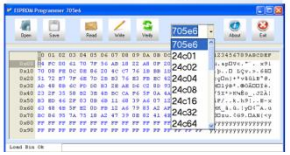
PLUS GRAND TEMPS D'ACCES

PLUS GRAND COUT

Les différents types de mémoire

SRAM

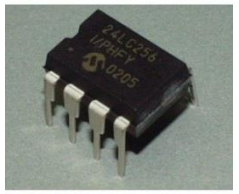

DRAM


Programmeur EPROM

- Mémoires vives : RAM (*Random Access Memory*)
  - Mémoire accessible en lecture et écriture
  - Mémoire volatile interne.
  - Compose la mémoire centrale et les caches
  - DRAM (Dynamic RAM) et SRAM (Static RAM) (60 à 5 ns)

- Mémoires mortes : ROM (*Read Only Memory*)
  - Mémoire accessible en lecture ; Mémoire non volatile interne.
  - *une fois l'information enregistrée, celle-ci ne peut pas (ou difficilement) être modifiée*.
    - PROM (programmable ROM) le contenu peut être modifié une fois par l'utilisateur, à l'aide d'un équipement spécialisé
    - EPROM (Erasable PROM): le contenu peut être effacé et modifié. L'effacement s'effectue par le biais d'une exposition àdes rayons ultra-violets (plusieurs minutes)
    - EEPROM (electrically EPROM) : le contenu est effacé électriquement (quelques millisecondes)
    - Flash : le contenu est effacé électriquement et plus rapidement que sur les EEPROM

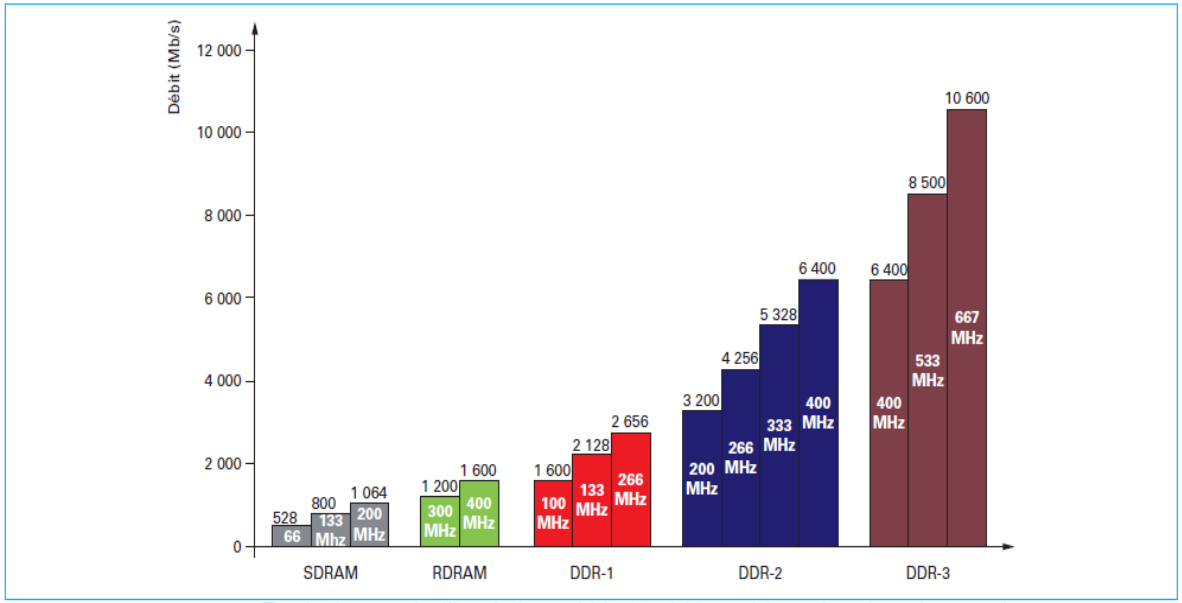

EPROM et effaceur UV


EEPROM

Mémoires vives : RAM (*Random Access Memory*)

- DRAM : mémoire dynamique. Peu couteuses, elles composent la mémoire centrale de l'ordinateur.
  - 1 cellule mémoire mémorise un bit et est constituée par un transistor et un condensateur
    - le condensateur se décharge dans le temps. Il convient de recharger chaque cellule périodiquement (1000 fois / s) : le rafraichissement de la mémoire.
  - Se présente sous la forme de barrette DIMM (*Dual Inline Memory Module*).
  - Temps d'accès : 60 ns (DRAM) à 10 ns (SDRAM)

le **débit** correspond au nombre d'octets que la mémoire peut fournir par unité de temps.



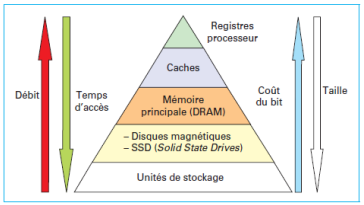
Techniques de l'ingénieur, H1002, hiérarchie mémoire : les caches

Mémoires vives : RAM (*Random Access Memory*)

- SRAM : mémoire statique. Plus couteuses et encombrantes, elles composent les caches du processeur.

  - 1 cellule mémoire mémorise un bit et est constituée par 4 à 6 transistors (*circuit de type bascule*)
  - Lecture non destructrice. Temps de cycle = temps de latence

  - Temps d'accès : 10 ns



Les différents types de mémoire

Contient les informations manipulées couramment par le processeur

registre

Mémoire SR A M

Mémoire DR A M

Mémoire ROM

Mémoire Vive, volatile (lecture/écriture)
Contient les informations
Les plus récemment accédées par le processeur
→ Un sous ensemble de la DRAM

Mémoire Vive, volatile (lecture/écriture)
Contient le code et les données
des programmes exécutés par le processeur

Mémoire Morte non volatile (lecture)
Contient l'amorce (boot) de l'ordinateur

Les mémoires de masse

- Conservation permanente des données. Grande capacité.

- L e disque dur : plateaux magnétiques qui tournent autour d'un axe. Ecriture par magnétisation de la surface

  Temps d'accès = 20 ms.

- Disque SSD (Solid Stat Disks) : à base de mémoire Flash

critère	Disque dur magnétique	Disque SSD
consommation	++	faible
bruit	++	nul
débit	100 Mo/s	2 à 4 fois plus
Prix 250 Go 1 To	Moyen 35 euros 70 euros	+++++ 170 euros 400 euros

Aussi, à l'heure actuelle, cette mémoire permanente est utilisée en remplacement d'un disque magnétique dans les mini-PC. Sur les PC classiques, la configuration de certains ordinateurs est la suivante :

- un disque SSD abrite le système d'exploitation, ce qui permet un boot très rapide de l'ordinateur ;
- un ou plusieurs disques magnétiques mémorisent les données des utilisateurs

Le principe de hiérarchie mémoire : les caches
Principe de fonctionnement

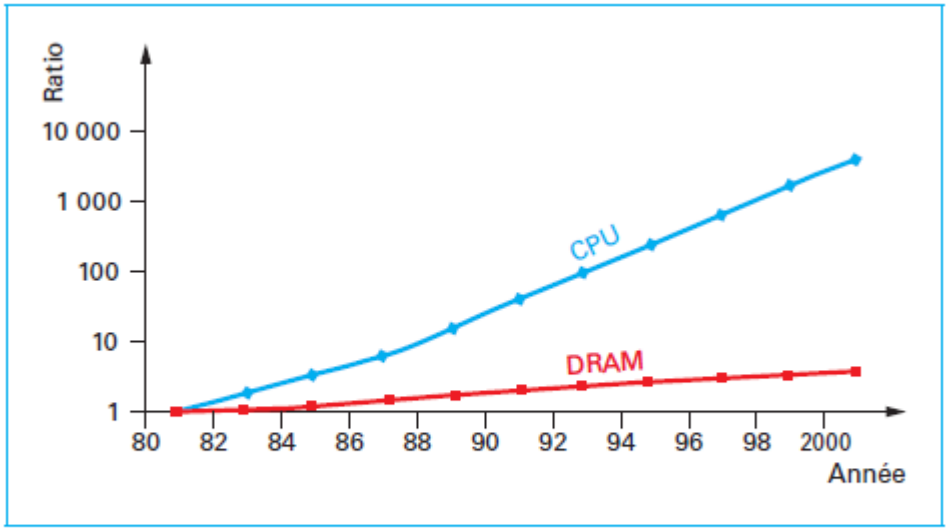


Figure 3 – Évolution des performances des processeurs (instructions exécutées par seconde) et des DRAM (1/temps d'accès) par rapport aux valeurs de 1981 [2]

Techniques de l'ingénieur, H1002, hiérarchie mémoire : les caches

Hiérarchie Mémoire

vitesse — 6-35 ns — 70 - 120 ns

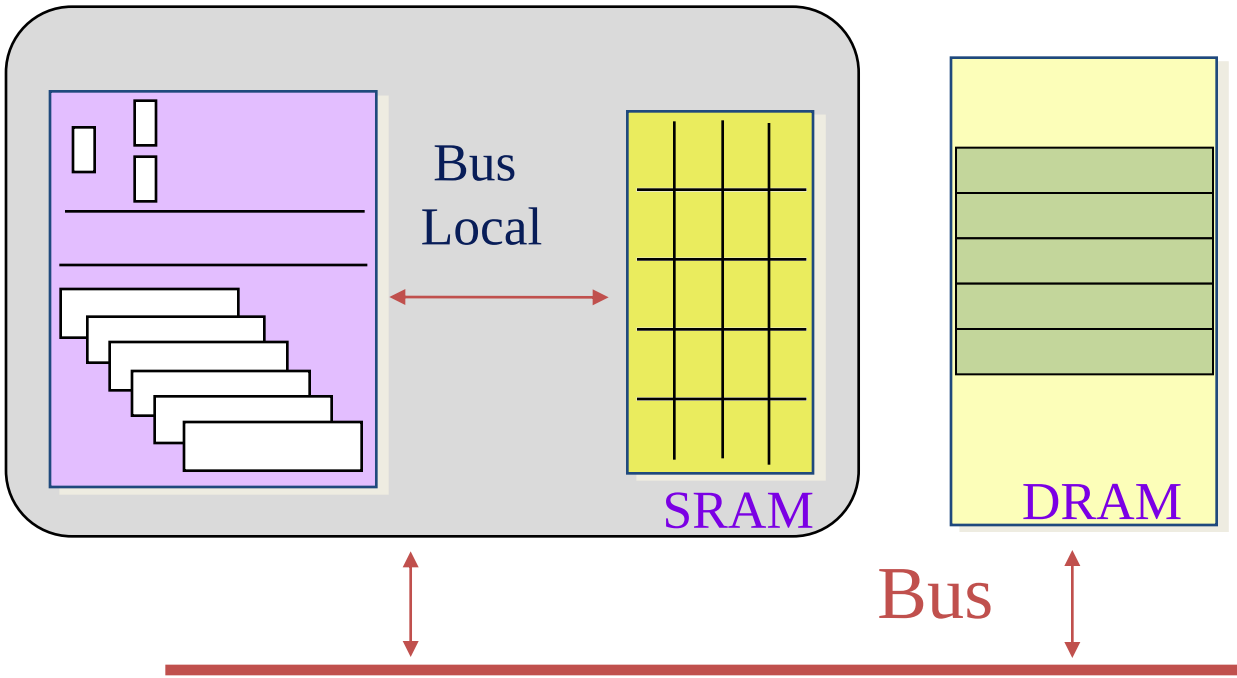
cout Le plus élevé — Le moins élevé

capacité La plus petite — La plus grande

Processeur Registres

Mémoire Cache

Mémoire centrale



La mémoire cache est une mémoire intermédiaire placée entre le processeur et la mémoire centrale dont le temps d'accès est de 4 à 20 fois inférieur à celui de la mémoire centrale.

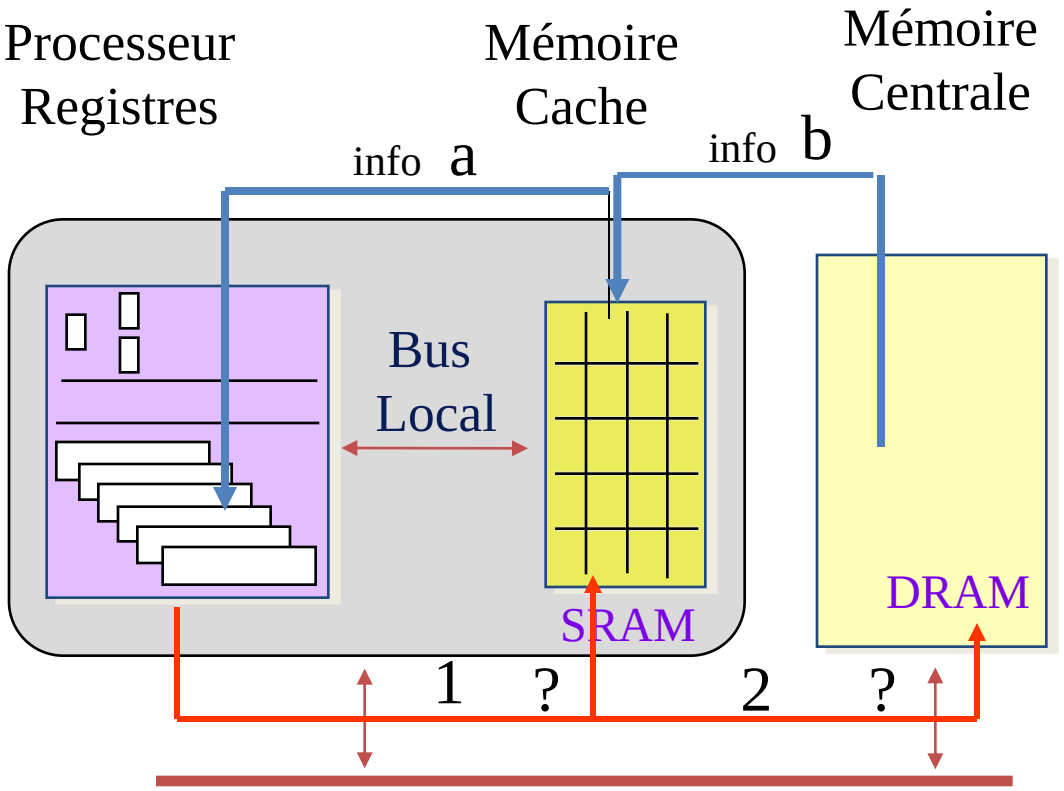
Elle comporte un nombre fini d'entrées. Une entrée contient n mot mémoire et s'appelle une ligne.

Tableau 2 – Temps d'accès pour un Pentium M

Élément	Temps d'accès
Registre	⩽ 1 cycle
Cache L1	~ 3 cycles
Cache L2	~ 14 cycles
Mémoire principale	~ 240 cycles

Mémoire cache : principe

La stratégie suivie s'appuie sur le principe de localité



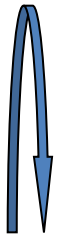
1. L'info cherchée est-elle dans le cache ?
OUI / Succès (a) : ramener l'info dans le processeur
NON / Défaut (2) : chercher l'info dans la mémoire centrale

2. L'info est-elle en mémoire centrale ?
OUI / Succès (b) : ramener l'info dans le cache , puis dans le processeur (a)
NON / Défaut

Mémoire cache
Principe de localité

- Localité temporelle : si une donnée d'adresse A est accédée à un temps t, la probabilité qu'elle soit de nouveau accédée aux temps t+1, t+2 est très forte.
  - ☞ La donnée est remontée dans le cache pour minimiser les temps d'accès suivants

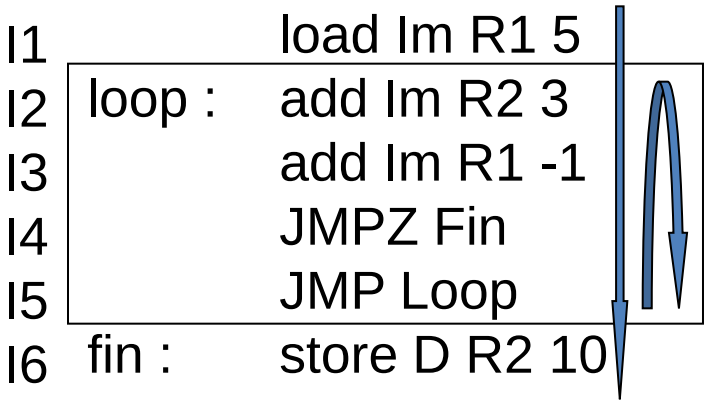
```
I1          load Im R1 5
I2  loop :  add Im R2 3
I3          add Im R1 -1
I4          JMPZ Fin
I5          JMP Loop
I6  fin :   store D R2 10
```



Localité temporelle
Premier accès aux instructions
I2, I3, I4, I5 → en MC
Les 4 accès suivants s'effectuent à partir du cache

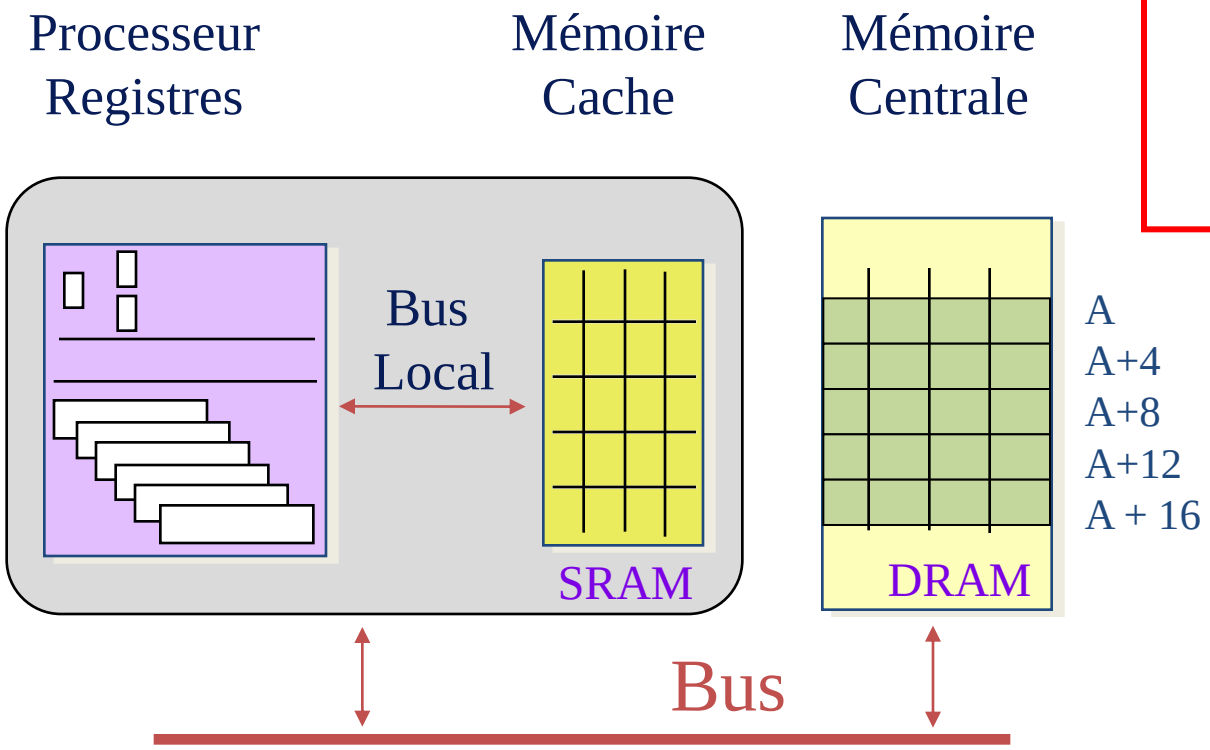
Mémoire cache
Principe de localité

- Localité spatiale : si une donnée d'adresse A est accédée à un temps t, la probabilité que les données d'adresses voisines soient accédées aux temps t+1, t+2 est très forte.
  - ☞ La donnée d'adresse A et également les données d'adresse voisines sont remontées dans le cache pour minimiser les temps d'accès suivants



Localité spatiale
Premier accès à l'instruction I1
→ en MC
Les accès à I2, I3, I4, I5, I6 s'effectuent à partir du cache

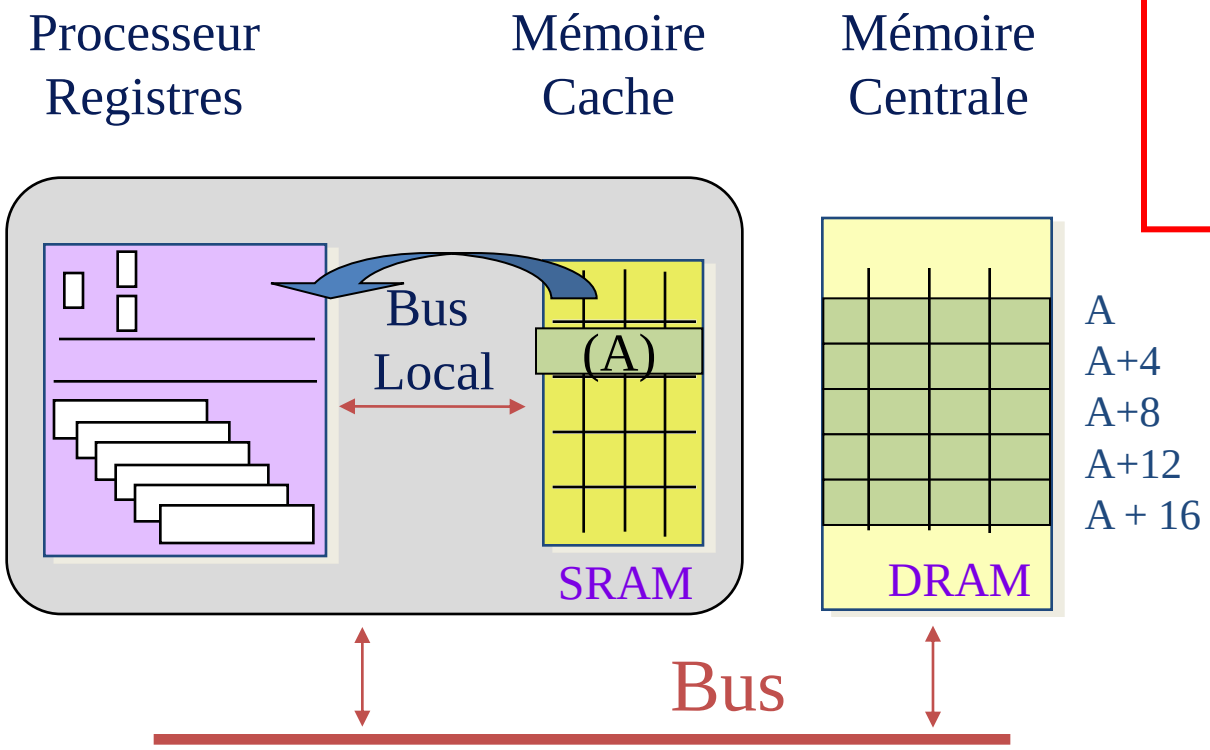
Mémoire cache
Le cache en lecture



Lecture
Load D R1 A

Si (A) présent
Alors Charger processeur avec (A)
Sinon
    Charger cache avec (A) et ses voisines
    Charger processeur avec (A)
FinSi
FinSi

Mémoire cache
Le cache en lecture



Si (A) présent Alors Charger processeur avec (A)

Sinon

Charger cache avec (A) et ses voisines

Charger processeur avec (A)

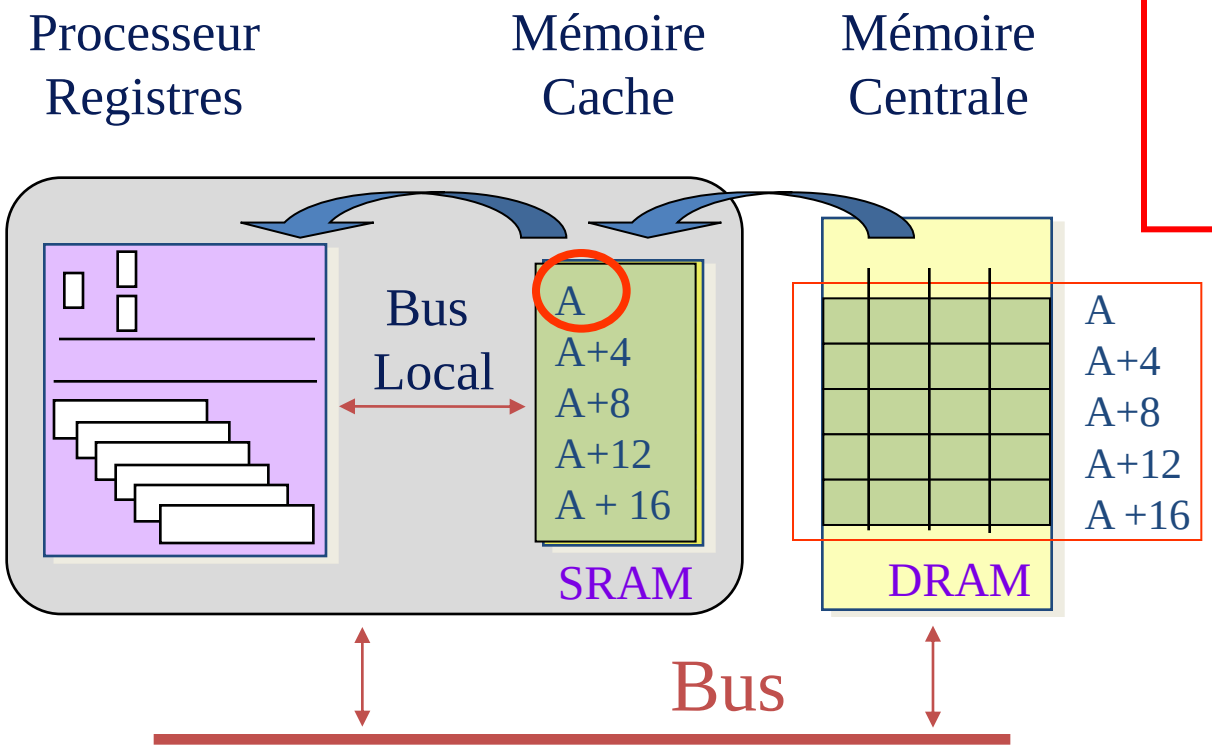
FinSi

Lecture

Load D R1 A

FinSi

Mémoire cache
Le cache en lecture



Lecture

Load D R1 A

Si (A) présent

Alors Charger processeur avec (A)

Sinon

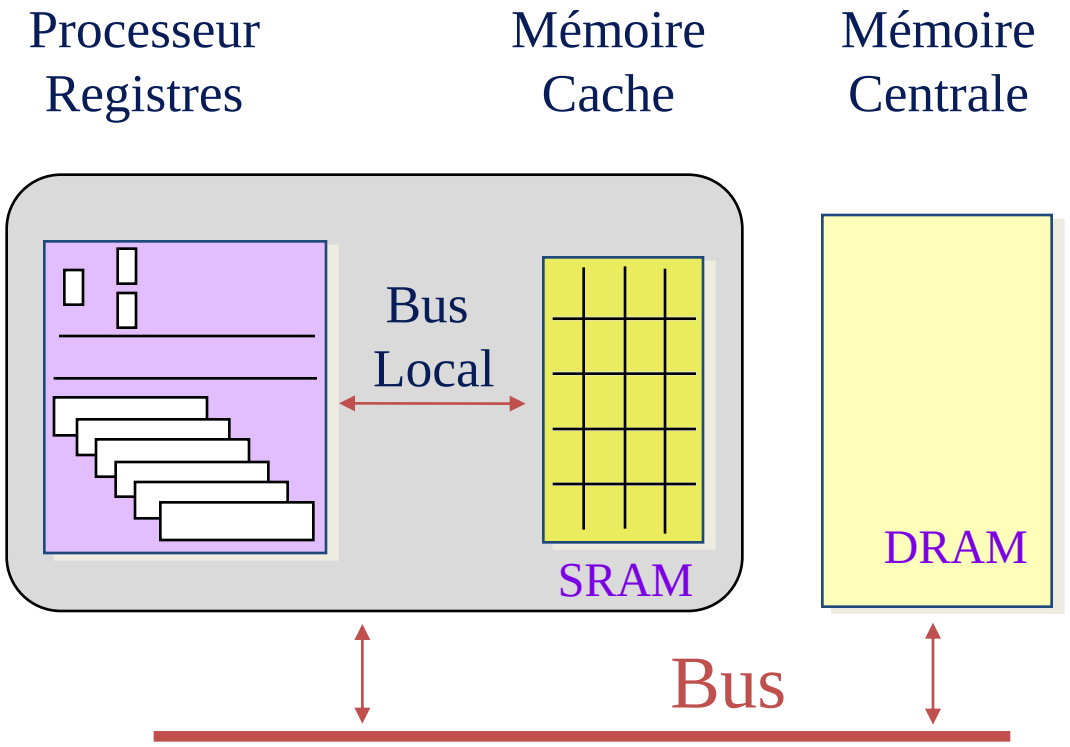
Charger cache avec (A) et ses voisines

Charger processeur avec (A)

FinSi

FinSi

Mémoire cache
Le cache en écriture



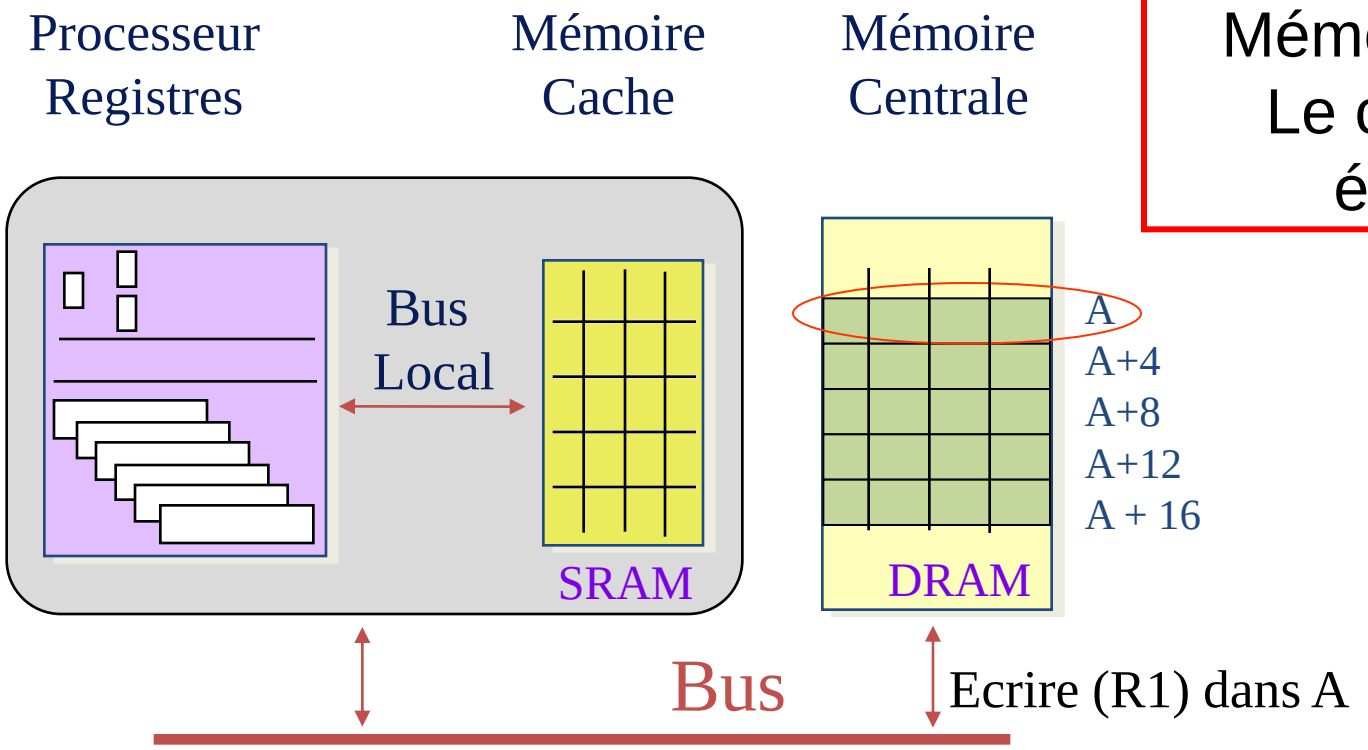
Ecriture

Store D R1 A

Si (A) présente Alors Modifier (A) dans le cache
Modifier (A) en mémoire principale
Sinon Modifier (A) en mémoire principale
FinSi

Write Through
Write Back

Mémoire cache
Le cache en écriture



Ecriture

Store D R1 A

Si (A) présente Alors Modifier (A) dans le cache

Modifier (A) en mémoire principale

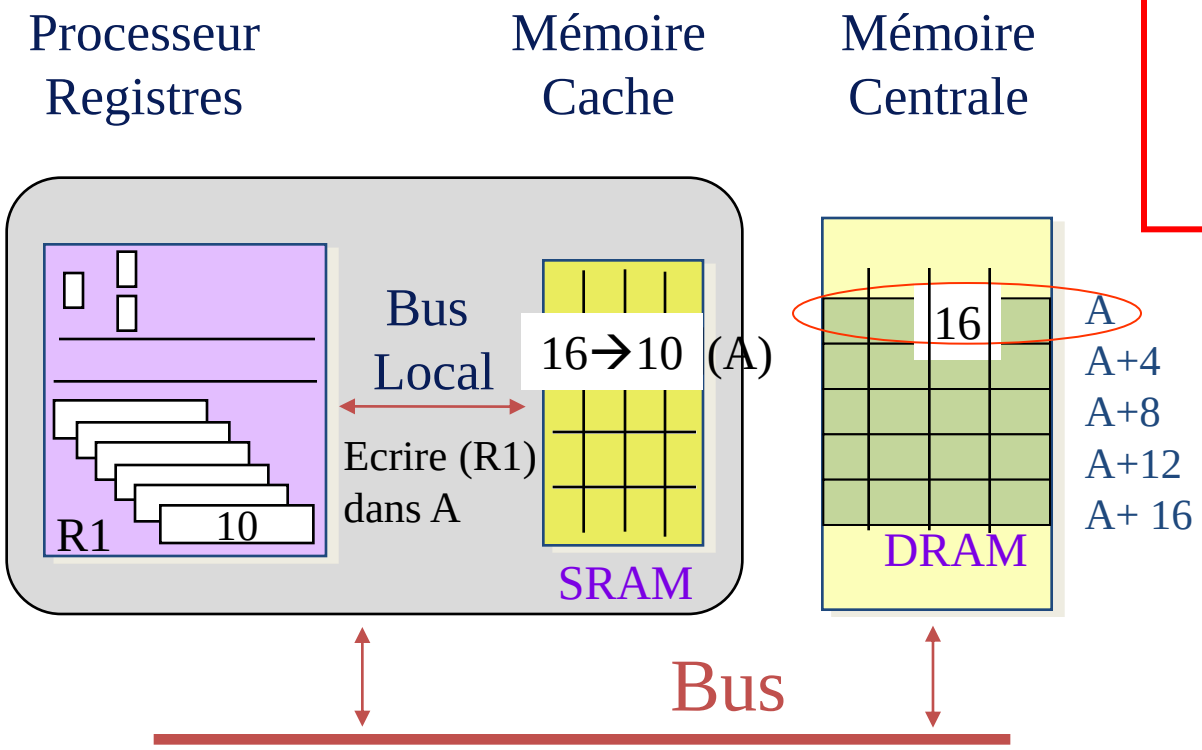
Sinon Modifier (A) en mémoire principale

FinSi

Write Through

Write Back

Mémoire cache
Le cache en écriture



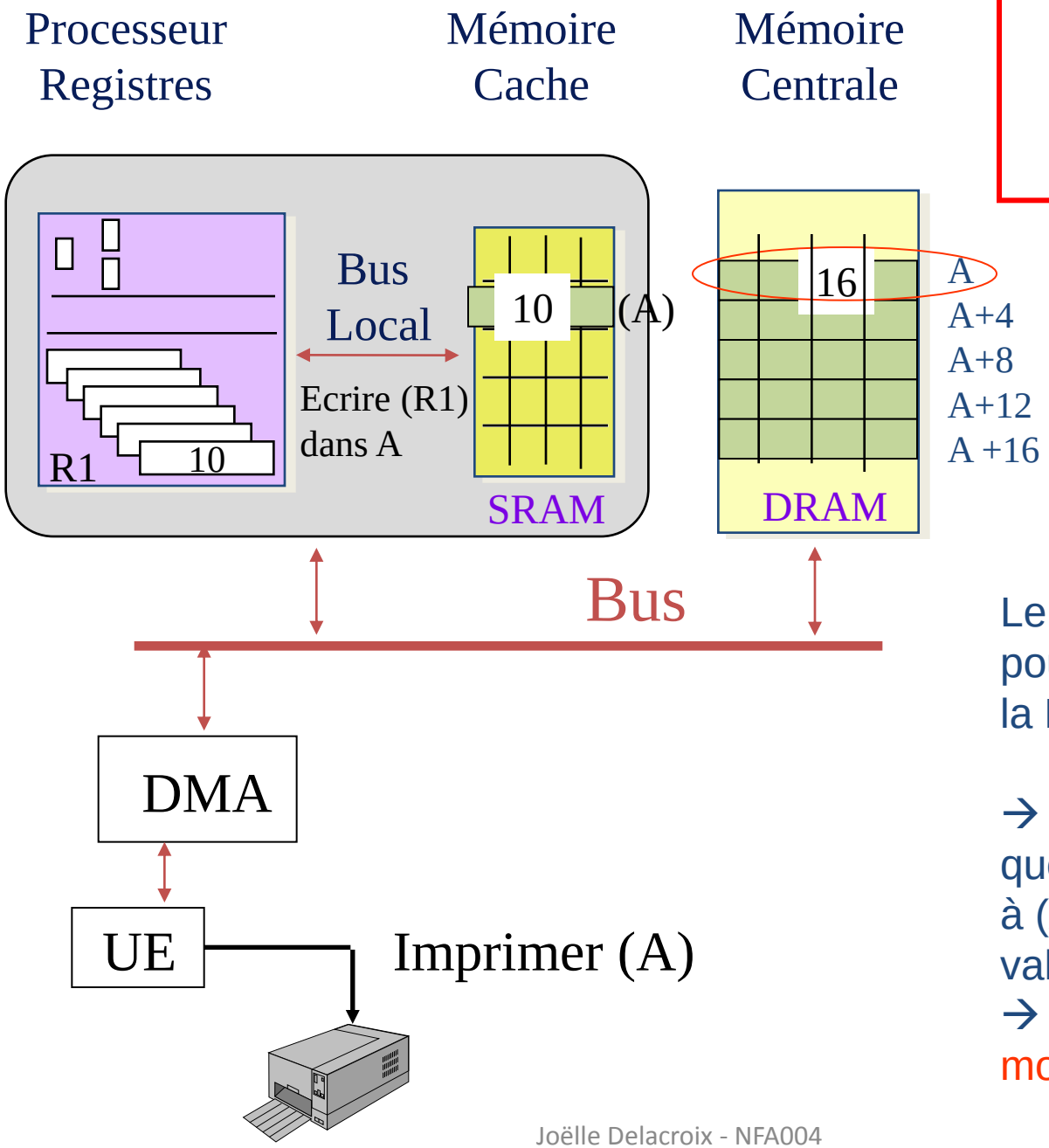
Ecriture

Store D R1 A

Si (A) présente Alors Modifier (A) dans le cache
Modifier (A) en mémoire principale
Sinon Modifier (A) en mémoire principale
FinSi

Write Through
Write Back

Mémoire cache
Le cache en écriture

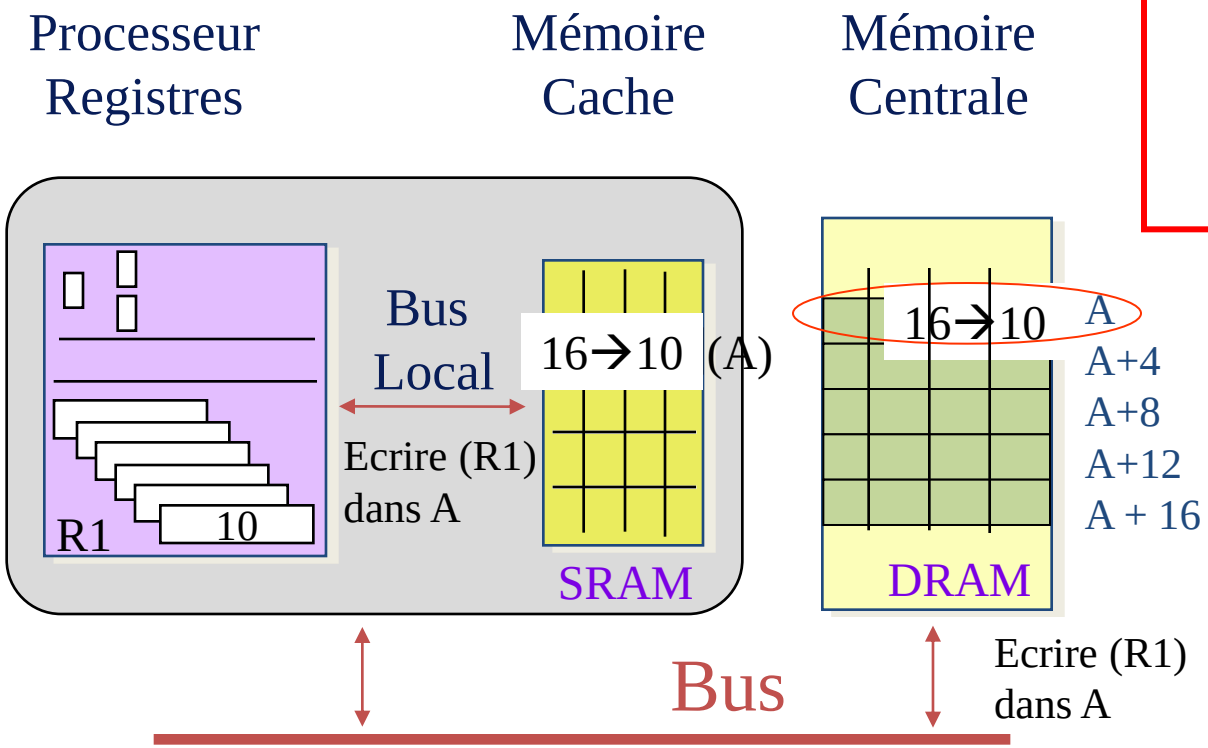


Le contenu du cache pour (A) est différent de la MC

→ Un autre dispositif tel que un DMA accédant à (A) ne voit pas la valeur modifiée

→ Politique de modification de la MC

Mémoire cache
Le cache en écriture



Ecriture

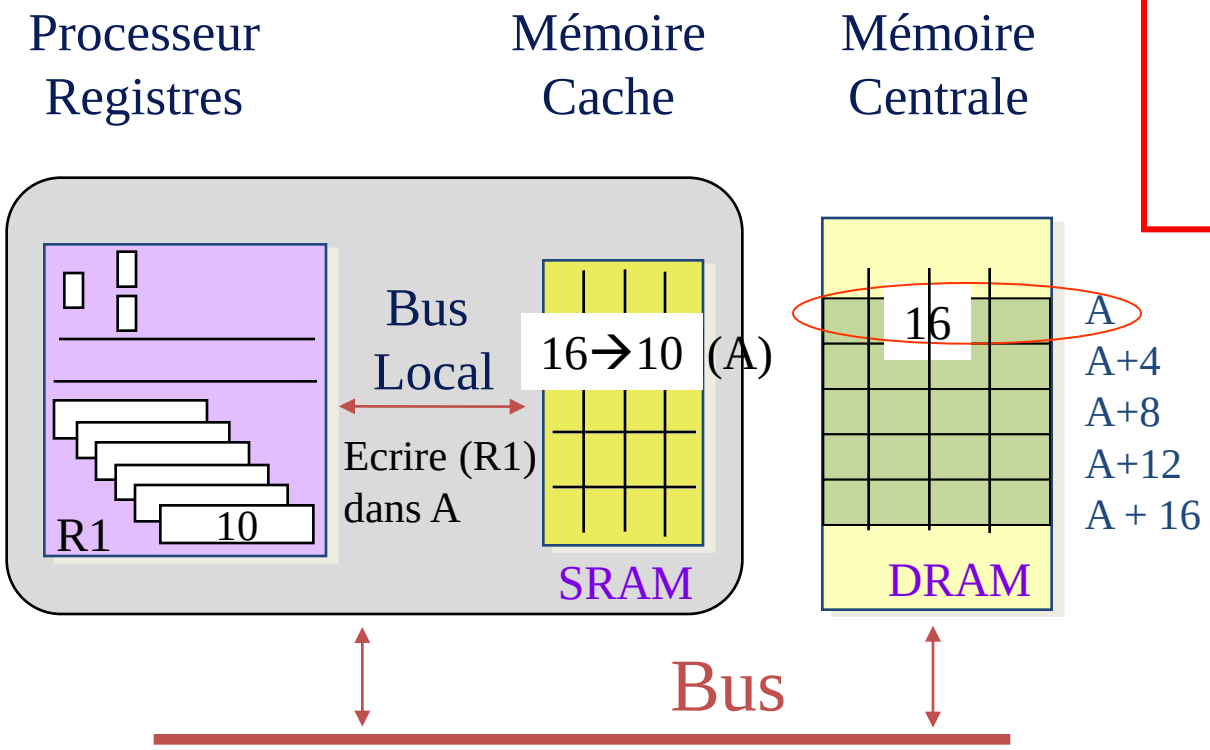
Store D R1 A

Si (A) présente Alors Modifier (A) dans le cache
Modifier (A) en mémoire principale

Write Through (écriture Immédiate)

L'écriture en mémoire centrale est effectuée en même temps que dans le cache
→ Cohérence maximale
→ Coût plus élevé d'une écriture

Mémoire cache
Le cache en écriture



Ecriture

Store D R1 A

Si (A) présente Alors Modifier (A) dans le cache

Modifier (A) en mémoire principale

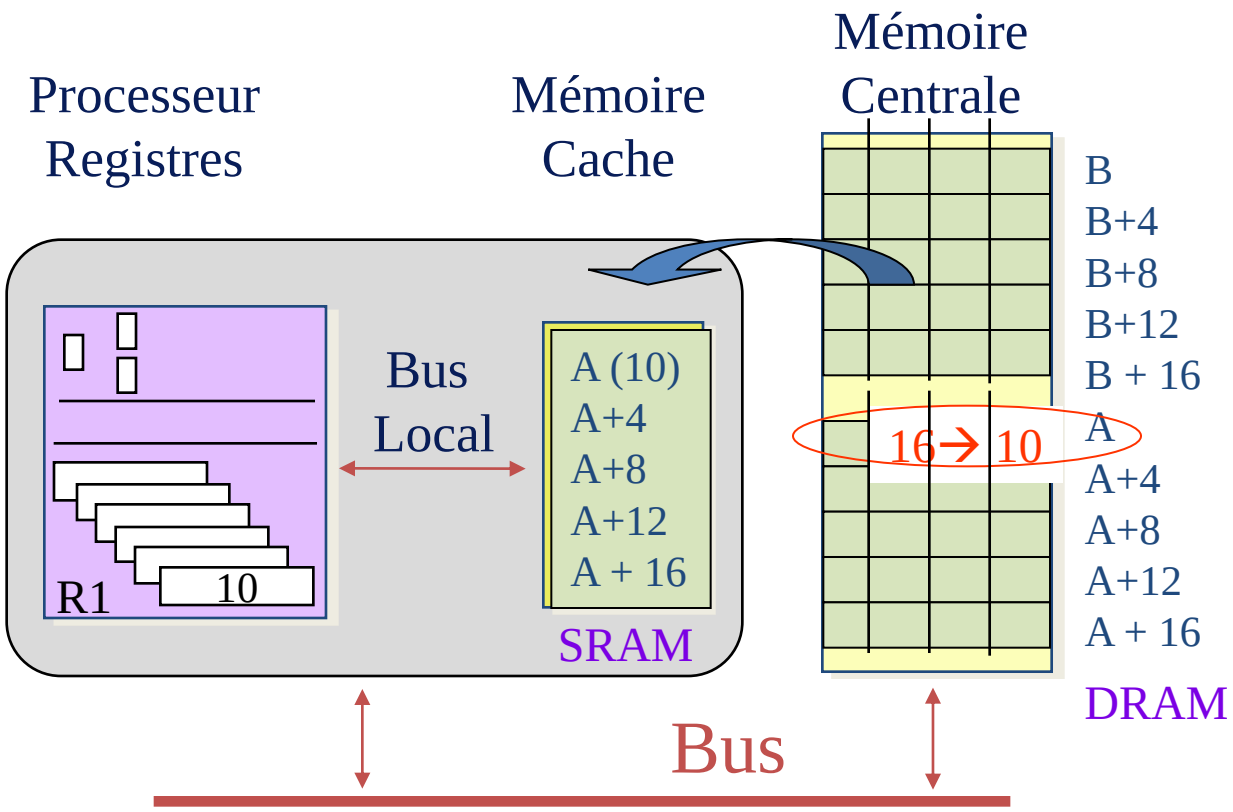
Write Back
(écriture différée)

L'écriture en mémoire centrale est effectuée plus tard lorsque l'entrée du cache doit être remplacée

→ Cohérence moindre

→ Coût moins élevé d'une écriture

Mémoire cache
Le cache en écriture



Ecriture

Load D R2 B

Ecrire l'entrée du cache modifiée dans A

Il y a défaut au niveau du cache. Le mot B et ses voisins doivent être chargés dans le cache.
Ils remplacent le mot A et ses voisins.
La modification de (A) est recopiée en mémoire centrale

Write Back (écriture différée)

L'écriture en mémoire centrale est effectuée plus tard lorsque l'entrée du cache doit être remplacée

→ Cohérence moindre

→ Coût moins élevé d'une écriture

Mémoire cache
Performances

Soient

h, la probabilité de succès

Tc le temps d'accès au cache

Tm, le temps de lecture d'un bloc de mots en mémoire centrale

Td, le temps d'accès à un mot en mémoire centrale

alors Teff, le temps effectif pour accéder à une information

$$Teff = h \times Tc + (1 - h) \times (Tm + Tc)$$

h	Tc (cycle)	Tm (cycle)	Td (cycle)	Teff
0,9	1	20	5	3
0,8	1	20	5	5
0,7	1	20	5	7

Mémoire cache
Architecture de caches

Tableau 2 – Temps d'accès pour un Pentium M	
Élément	Temps d'accès
Registre	⩽ 1 cycle
Cache L1	~ 3 cycles
Cache L2	~ 14 cycles
Mémoire principale	~ 240 cycles

Techniques de l'ingénieur, H1002, hiérarchie mémoire : les caches