ENICARTHAGE

المدرسة الوطنية للمهندسين بقرطاج

Ecole Nationale d'Ingénieurs de Carthage

MODULE: Architecture des systèmes à microprocesseurs

CHAPITRE II:

LES MEMOIRES

2018-2019

Caractéristiques d'une Mémoire

La capacité : c'est le nombre total de bits que contient la mémoire. Elle s'exprime aussi souvent en octet (byte).

Symbole	Préfixe	Décimal	Binaire
1 K	Kilo	10^3	2^{10}
1 M	Méga	10^6	2^{20}
1 G	Giga	10^9	2^{30}
1 T	Téra	10^{12}	2^{40}

Le format des données : c'est le nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du mot mémorisable.

Le temps d'accès : c'est le temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture en mémoire et l'instant où la première information est disponible sur le bus de données.

Le temps de cycle : il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture.

Symbole	Préfixe	Décimal
1 ms	Milli	10^{-3}
1 µs	Micro	10^{-6}
1 ns	nano	10^{-9}
1 ps	pico	10^{-12}

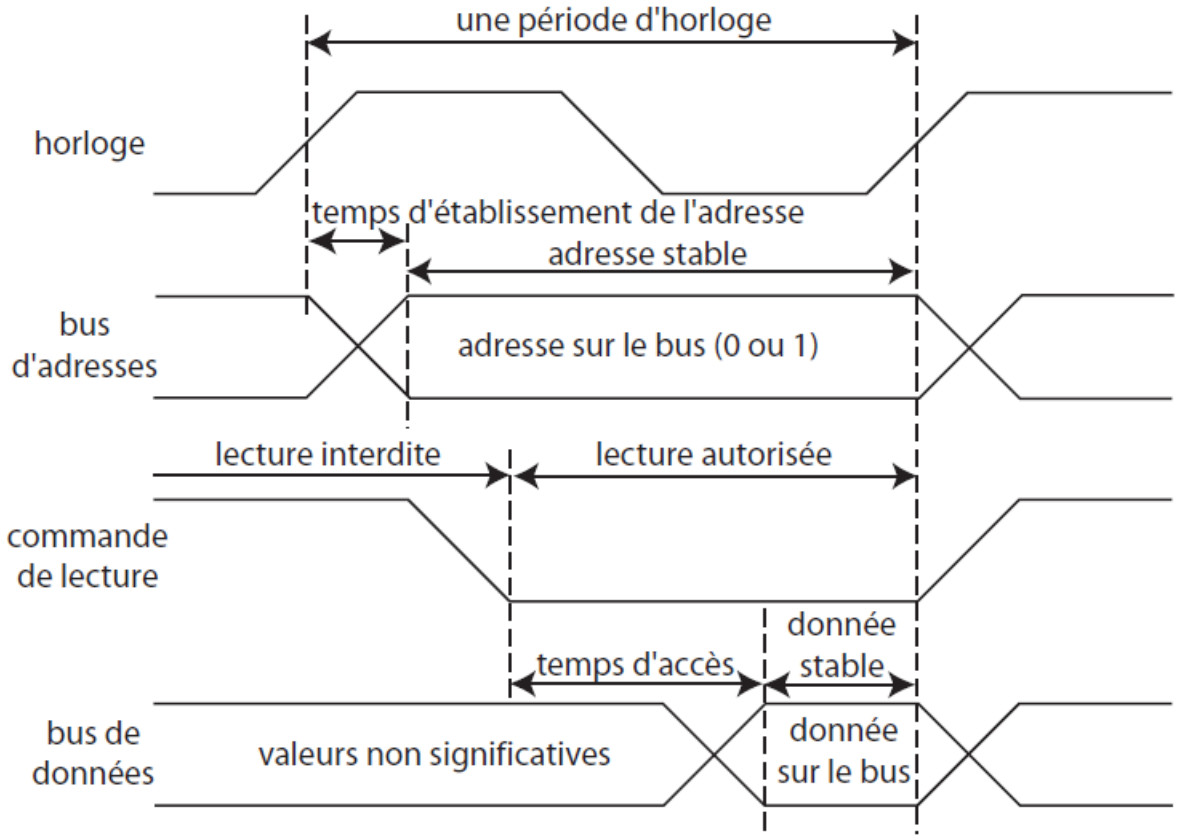
Le débit : c'est le nombre maximum d'informations lues ou écrites par seconde.

Volatilité : elle caractérise la permanence des informations dans la mémoire. L'information stockée est volatile si elle risque d'être altérée par un défaut d'alimentation électrique et non volatile dans le cas contraire.

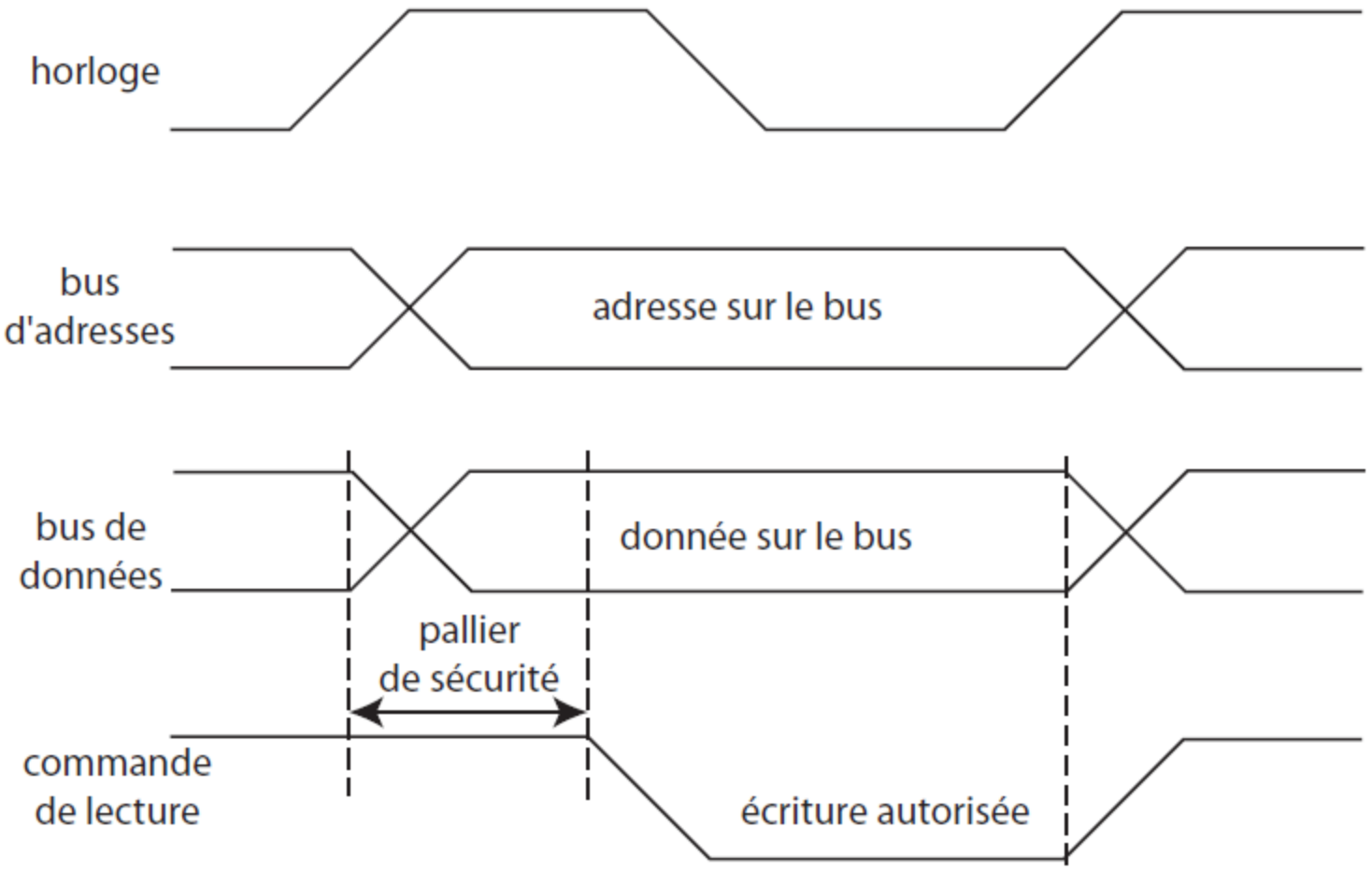
Chronogrammes de Lecture/Ecriture en Mémoire

Une caractéristique importante des mémoires est ***leur temps d'accès*** : c'est le temps qui s'écoule entre l'instant où l'adresse de la case mémoire est présentée sur le bus d'adresses et celui où la mémoire place la donnée demandée sur le bus de données. Ce temps varie entre 50 ns et 300 ns.

Chronogramme de lecture en mémoire

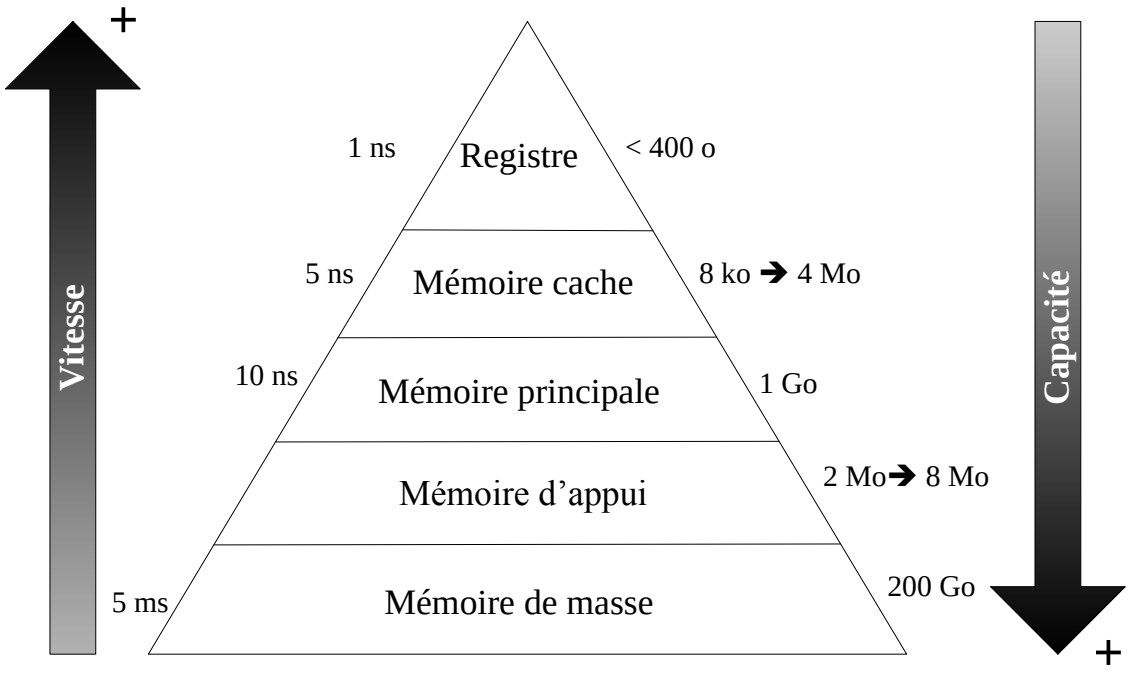


Chronogramme d'écriture en mémoire



Hiérarchie des Mémoires

Une mémoire idéale serait une mémoire de grande capacité, capable de stocker un maximum d'informations et possédant un temps d'accès très faible afin de pouvoir travailler rapidement sur ces informations. Mais il se trouve que les mémoires de grande capacité sont souvent très lente et que les mémoires rapides sont très chères. Et pourtant, la vitesse d'accès à la mémoire conditionne dans une large mesure les performances d'un système. En effet, c'est là que se trouve le goulot d'étranglement entre un microprocesseur capable de traiter des informations très rapidement et une mémoire beaucoup plus lente (ex : processeur à 3 Ghz et mémoire à 400 MHz). Or, on n'a jamais besoin de toutes les informations au même moment. Afin d'obtenir le meilleur compromis coût-performance, on définit donc une hiérarchie mémoire.



Types d'Accès aux Mémoires

Accès direct: Dans une mémoire à semi-conducteur, on accède directement à n'importe quelle information dont on connaît l'adresse, le temps pour obtenir l'information ne dépend pas de l'adresse. On dira que l'accès à une telle mémoire est **aléatoire**, **direct** ou encore **sélectif**. Exemple: mémoire DRAM.

Accès séquentiel: pour accéder à une information sur bande magnétique, il faut dérouler la bande en repérant tous les enregistrements jusqu'à ce que l'on trouve celui que l'on adresse. On dit alors que l'accès à l'information est séquentiel. Le temps d'accès est variable selon la position de l'information recherchée. Exemple: bande magnétique.

Accès semi-séquentiel: combinaison des accès direct et séquentiel. Pour un disque dur par exemple, l'accès à la piste est direct, puis l'accès au secteur est séquentiel.

Accès par le contenu: les informations sont identifiées par une clé et la recherche s'effectue sur cette clé de façon simultanée sur toutes les positions de la mémoire. Exemple: mémoire cache.

Mémoires Centrale: ROM et RAM

On distingue deux types de mémoires :

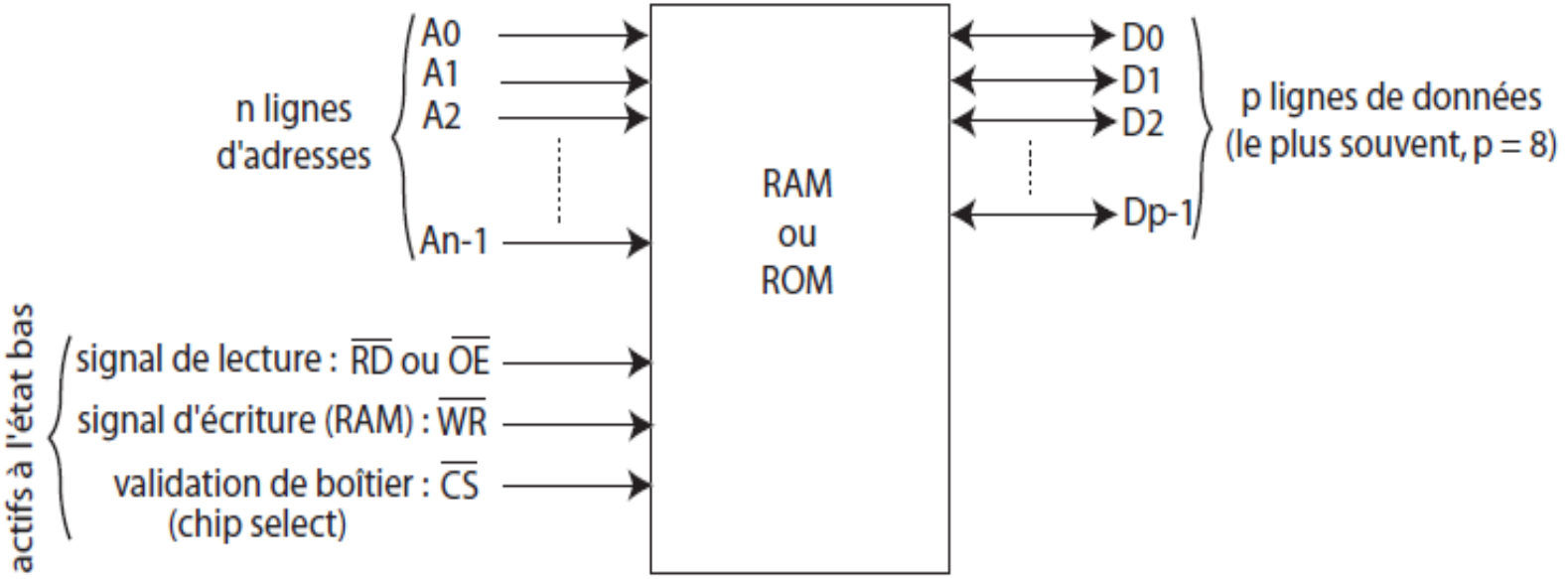
- **Les mémoires vives (RAM : Random Access Memory) ou mémoires volatiles:** Elles perdent leur contenu en cas de coupure d'alimentation. Elles sont utilisées pour stocker temporairement des données et des programmes. Elles peuvent être lues et écrites par le microprocesseur.

- **Les mémoires mortes (ROM : Read Only Memory) ou mémoires non volatiles:** Elles conservent leur contenu en cas de coupure d'alimentation. Elles ne peuvent être que lues par le microprocesseur (pas de possibilité d'écriture). On les utilise pour stocker des données et des programmes de manière définitive.

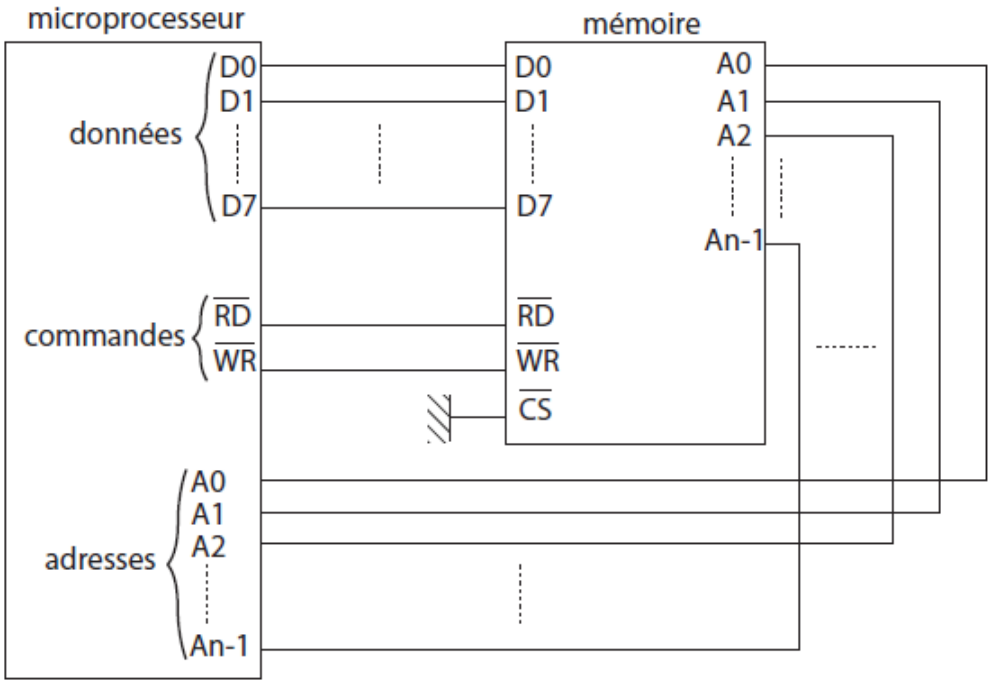
Schéma Fonctionnel de la Mémoire Centrale

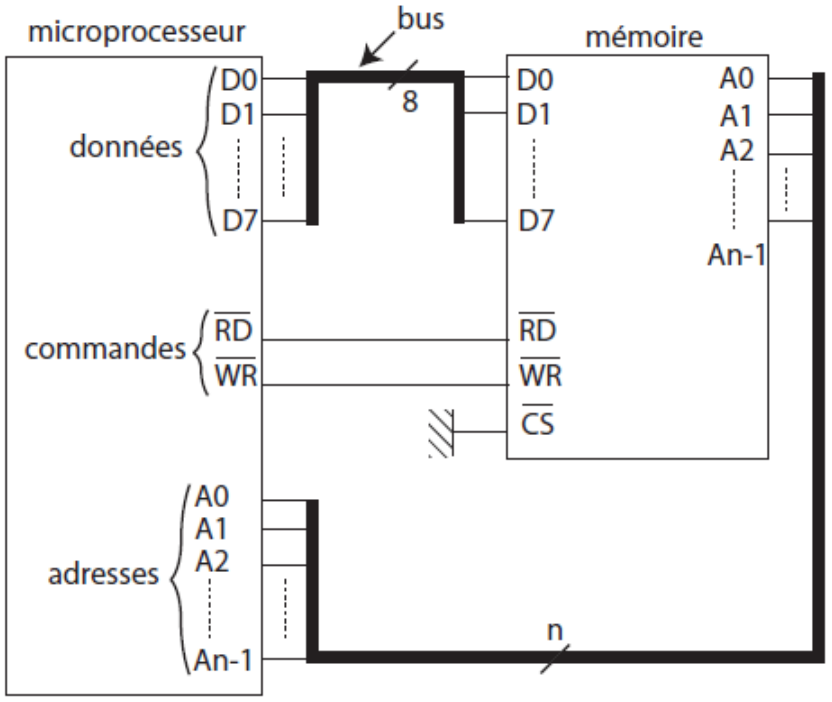
Le nombre de lignes d'adresses dépend de la capacité de la mémoire : n lignes d'adresses permettent d'adresser 2^n cases mémoire.
8 bits d'adresses permettent d'adresser 256 octets, 16 bits d'adresses permettent d'adresser 65536 octets (= 64 Ko), ...

Exemple : mémoire RAM 6264, capacité = 8K×8 bits : 13 broches d'adresses A0 à A12, 2^{13} = 8192 = 8 Ko.



Interfaçage Microprocesseur/Mémoire





Connexion de Plusieurs Boîtiers Mémoire sur le Bus d'un Microprocesseur

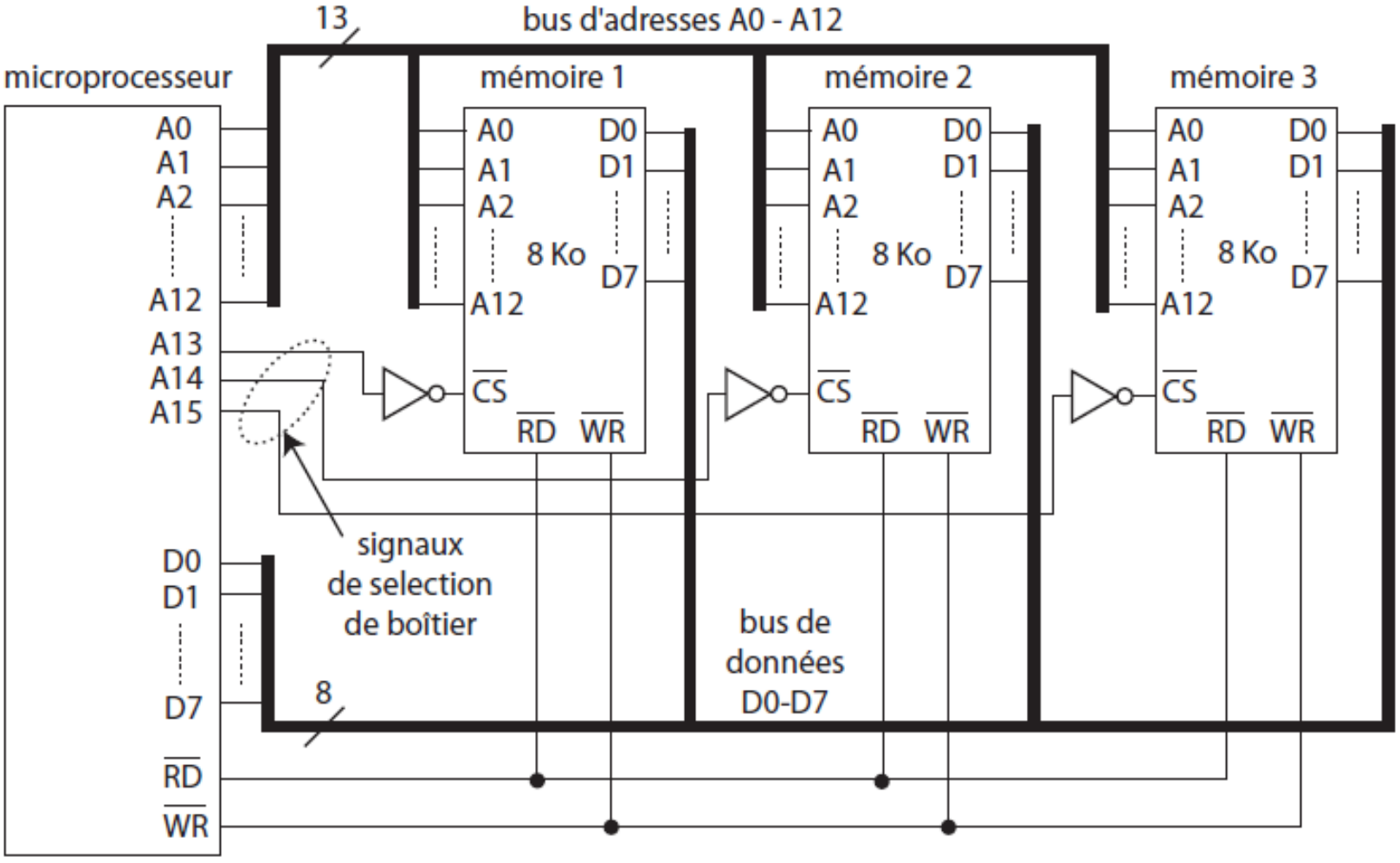
- Les boîtiers mémoire possèdent une broche notée ***CS : Chip Select.***

- Lorsque cette broche est active, le circuit peut être lu ou écrit.

- Lorsqu'elle est inactive, le circuit est exclu du service. Ses broches de données D0 à D7 passent à l'état de haute impédance : tout se passe comme si la mémoire était déconnectée du bus de données du microprocesseur, d'où la possibilité de connecter plusieurs boîtiers mémoire sur un même bus.

- Un seul signal CS doit être actif à un instant donné pour éviter les conflits entre les différents boîtiers.

Exemple : connexion de trois boitiers mémoire d'une capacité de 8 Ko chacun (13 lignes d'adresses) sur un bus d'adresse de 16 bits :



Dans un même boîtier, une case mémoire est désignée par les bits d'adresses A0 à A12 :

A12	A11	...	A1	A0
0	0	...	0	0

0000H

à

A12	A11	...	A1	A0
1	1	...	1	1

1FFFH

Pour atteindre la mémoire N°1, il faut mettre à 1 le bit A13 et à 0 les bits A14 et A15.
La plage d'adresses occupée par cette mémoire est donc :

A15	A14	A13	A12	...	A0
0	0	1	0	...	0

2000H

à

A15	A14	A13	A12	...	A0
0	0	1	1	...	1

3FFFH

De même, pour la mémoire N°2, on doit avoir A13 = 0, A14 = 1 et A15 = 0 d'où la plage d'adresses occupée cette mémoire :

A15	A14	A13	A12	...	A0
0	1	0	0	...	0

4000H

à

A15	A14	A13	A12	...	A0
0	1	0	1	...	1

5FFFH

Pour la mémoire N°3, on doit avoir A13 = 0, A14 = 0 et A15 = 1 d'où la plage d'adresses occupée cette mémoire :

A15	A14	A13	A12	...	A0
1	0	0	0	...	0

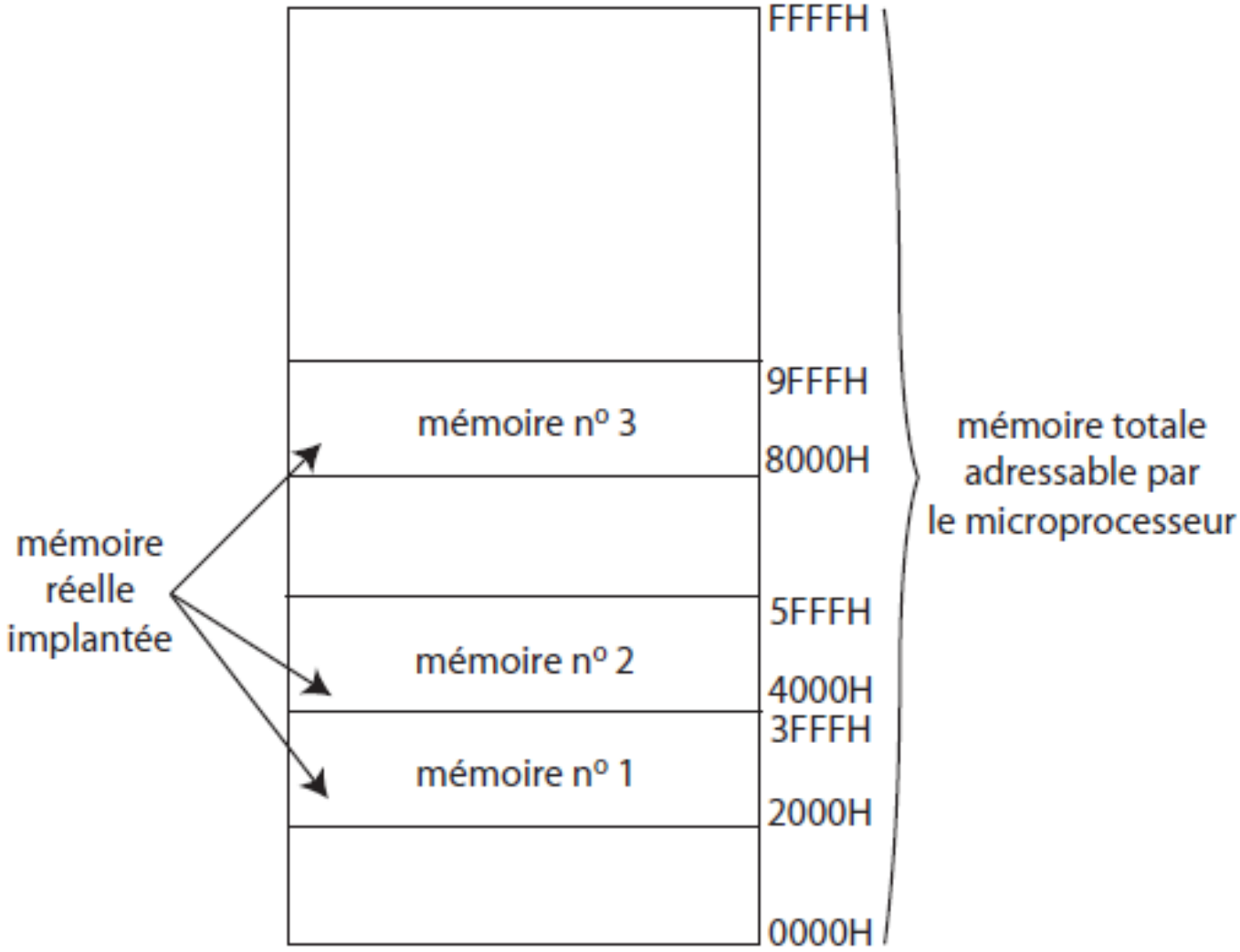
8000H

à

A15	A14	A13	A12	...	A0
1	0	0	1	...	1

9FFFH

On en déduit *la cartographie ou mapping de la mémoire visible* par le microprocesseur :



Décodage d'Adresses

Les trois bits A13, A14 et A15 fournissent en fait 8 combinaisons, d'où la possibilité de connecter jusqu'à 8 boîtiers mémoire de 8 Ko sur le bus.

La mémoire totale implantée devient donc de 8 × 8 Ko = 64 Ko : *valeur maximale* possible avec 16 bits d'adresses.

Pour cela, il faut utiliser un circuit de décodage d'adresses : un décodeur 3 vers 8.

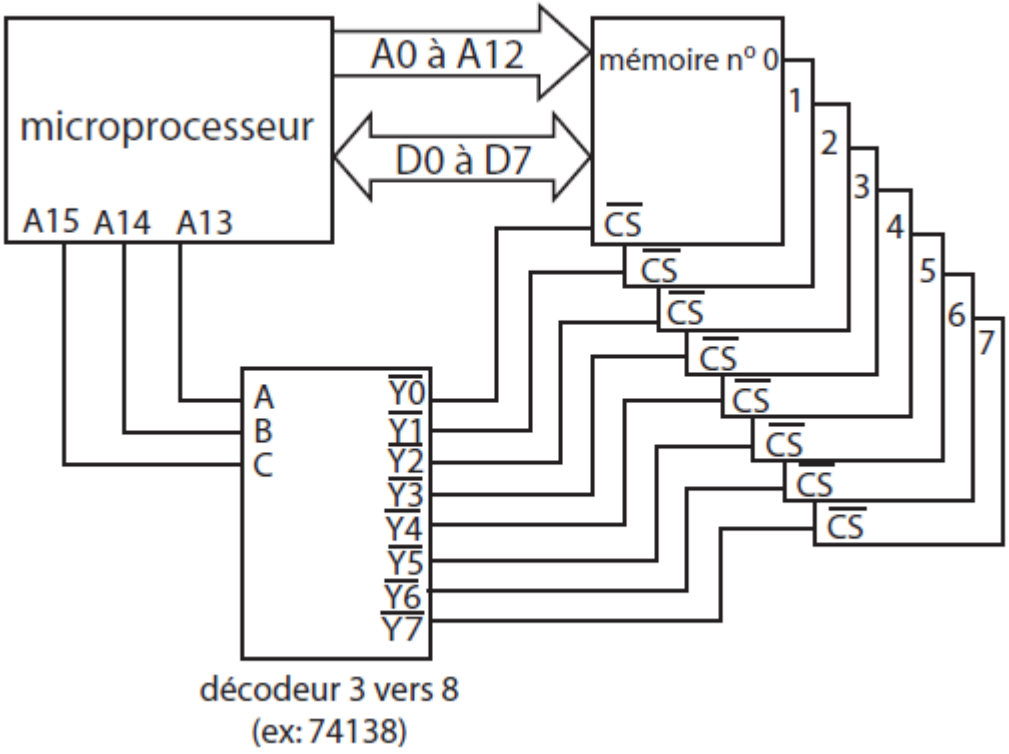
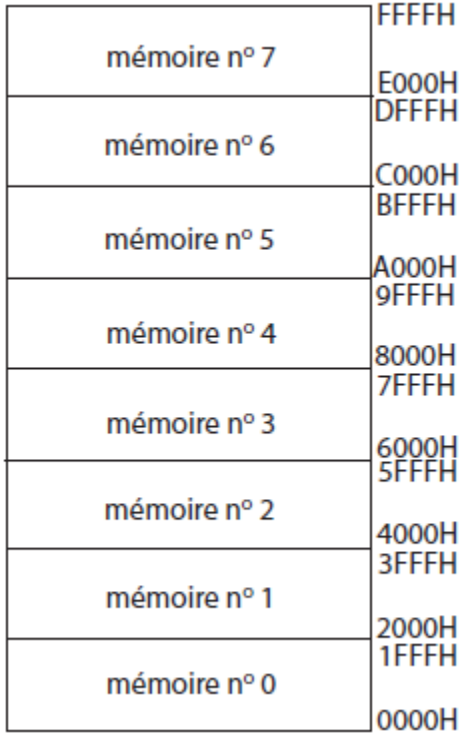


Table de vérité du décodeur d'adresses :

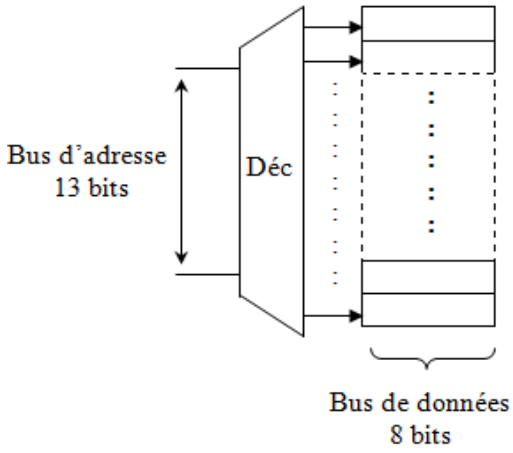
C	B	A	$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$	$\overline{Y4}$	$\overline{Y5}$	$\overline{Y6}$	$\overline{Y7}$
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

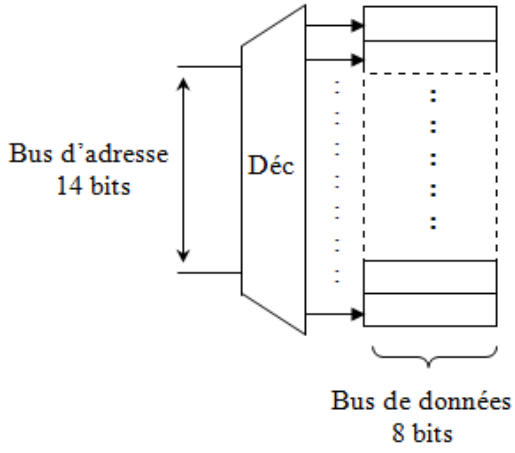
Le mapping de la mémoire :

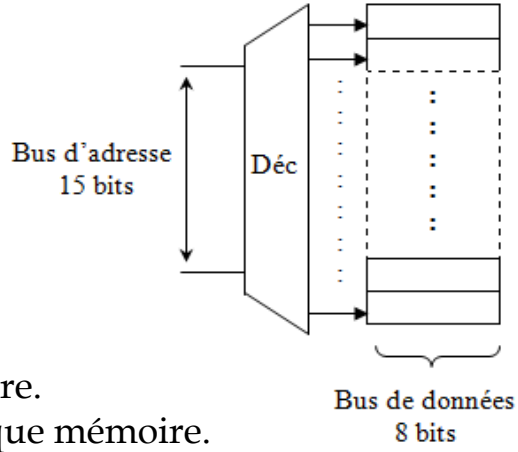


Exercice d'application:

Soient les circuits RAM suivants :

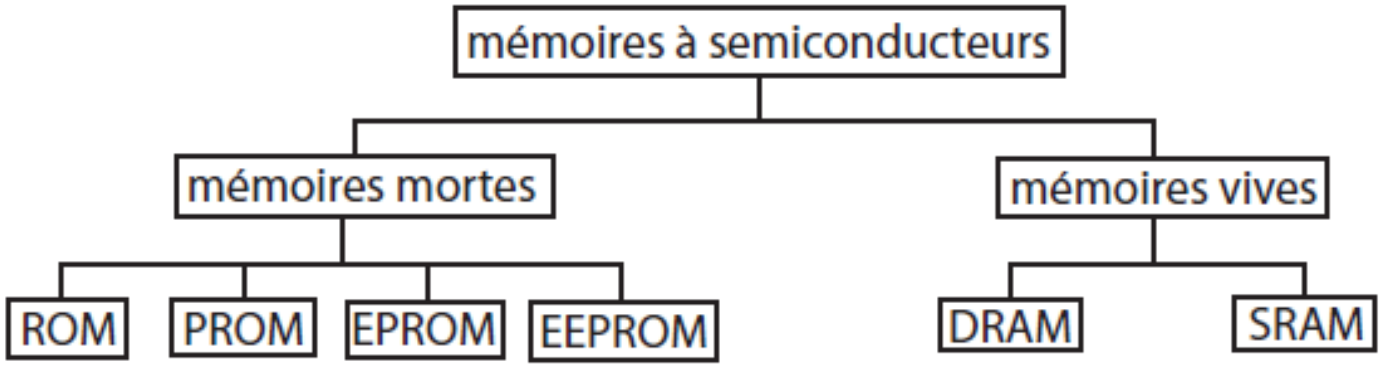






- Calculer la capacité totale de chaque mémoire.
- Calculer l'adresse de début et de fin de chaque mémoire.
- On veut associer à ces mémoires un processeur admettant 16 bits d'adresse et 8 bits de données. Donner la capacité totale adressable par le processeur.
- Dresser la table des adresses, sachant que les mémoires sont disposées successivement à partir de la première adresse (adresse 0) selon l'ordre suivant : Mémoire 1, Mémoire 2 et Mémoire 3.
- Calculer les équations de sélection CS_1, CS_2 et CS_3.
- Expliquer pourquoi la capacité totale adressable par le processeur permet d'introduire une quatrième puce mémoire. Calculer la taille maximale de la puce qu'on peut y ajouter.

Classification des Mémoires (1/2)



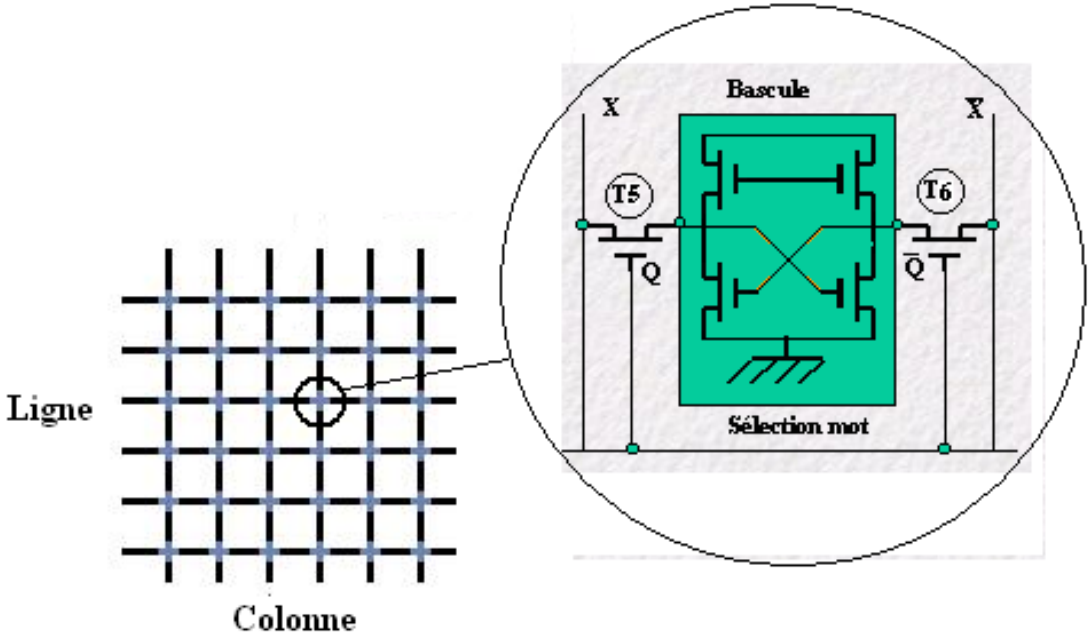
Mémoires vives

Une mémoire vive sert au stockage temporaire des données. Elle doit avoir un temps de cycle très court pour ne pas ralentir le microprocesseur. Les mémoires vives sont en général volatiles : elles perdent leurs informations en cas de coupure d'alimentation. Certaines d'entre elles, ayant une faible consommation, peuvent être rendues non volatiles par l'adjonction d'une batterie. Il existe deux grandes familles de mémoires RAM:

- ✓ Les RAM statiques
- ✓ Les RAM dynamiques

SRAM

Le bit mémoire d'une RAM statique (SRAM) est composé d'une bascule. Chaque bascule contient entre 4 et 6 transistors.



Avantage : très rapide, simple d'utilisation ➔ utilisation pour les mémoires caches.

Inconvénient : compliquée à réaliser, coûteuse.

DRAM

Dans les RAM dynamiques (DRAM), l'information est mémorisée sous la forme d'une charge électrique stockée dans un condensateur (capacité grille substrat d'un transistor MOS).

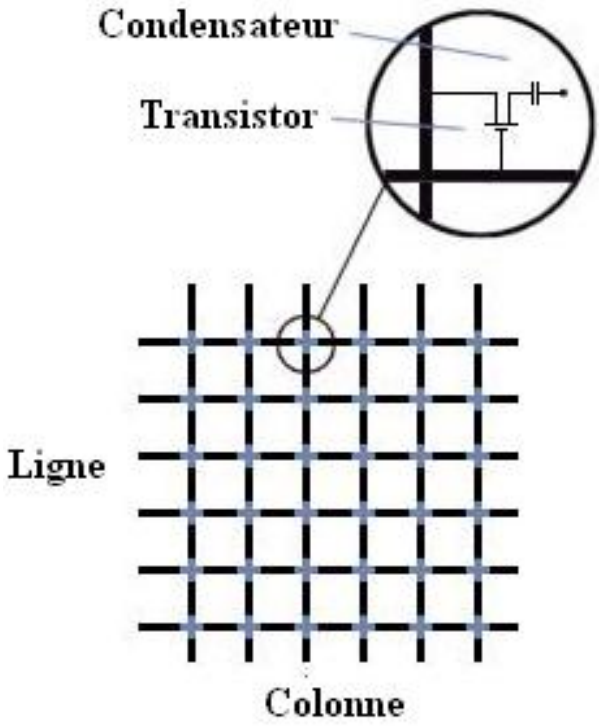
Avantages

Cette technique permet une plus grande densité d'intégration, car un point mémoire nécessite environ quatre fois moins de transistors que dans une mémoire statique. Sa consommation s'en retrouve donc aussi très réduite.

Inconvénients

La présence de courants de fuite dans le condensateur contribue à sa décharge. Ainsi, l'information est perdue si on ne la régénère pas périodiquement (charge du condensateur). Les RAM dynamiques doivent donc être rafraîchies régulièrement pour entretenir la mémorisation : il s'agit de lire l'information et de la recharger.

Ce rafraîchissement indispensable a plusieurs conséquences :
Il complique la gestion des mémoires dynamiques car il faut tenir compte des actions de rafraîchissement qui sont prioritaires.
La durée de ces actions augmente le temps d'accès aux informations.
D'autre part, la lecture de l'information est destructive. En effet, elle se fait par décharge de la capacité du point mémoire lorsque celle-ci est chargée. Donc toute lecture doit être suivie d'une réécriture.



Evolution de la DRAM (1/2)

La DRAM FPM (Fast Page Mode, 1987): Elle permet d'accéder plus rapidement à des données en introduisant la notion de page mémoire. (33 à 50 Mhz)

La DRAM EDO (Extended Data Out, 1995): Les composants de cette mémoire permettent de conserver plus longtemps l'information, on peut donc ainsi espacer les cycles de rafraîchissement. Elle apporte aussi la possibilité d'anticiper sur le prochain cycle mémoire. (33 à 50 Mhz)

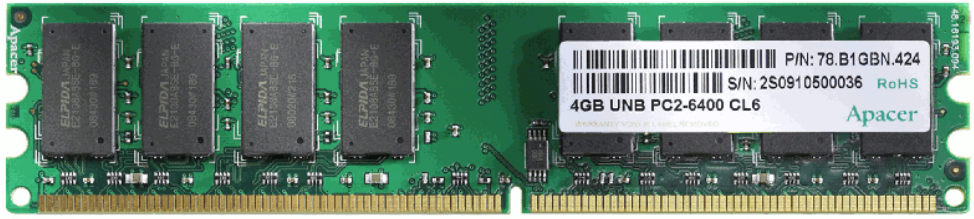
La DRAM BEDO (Bursted EDO, 1996): On n'adresse plus chaque unité de mémoire individuellement lorsqu'il faut y lire ou y écrire des données. On se contente de transmettre l'adresse de départ du processus de lecture/écriture et la longueur du bloc de données (Burst). Ce procédé permet de gagner beaucoup de temps, notamment avec les grands paquets de données tels qu'on en manipule avec les applications modernes. (66 MHz)

La Synchronous DRAM (SDRAM, 1997): La mémoire SDRAM a pour particularité de se synchroniser sur une horloge. Les mémoires FPM, EDO étaient des mémoires asynchrones et elles induisaient des temps d'attentes lors de la synchronisation. Elle se compose en interne de deux bancs de mémoire et des données peuvent être lues alternativement sur l'un puis sur l'autre de ces bancs grâce à un procédé d'entrelacement spécial. Le protocole d'attente devient donc tout à fait inutile. Cela lui permet de supporter des fréquences plus élevées qu'avant (100 Mhz).

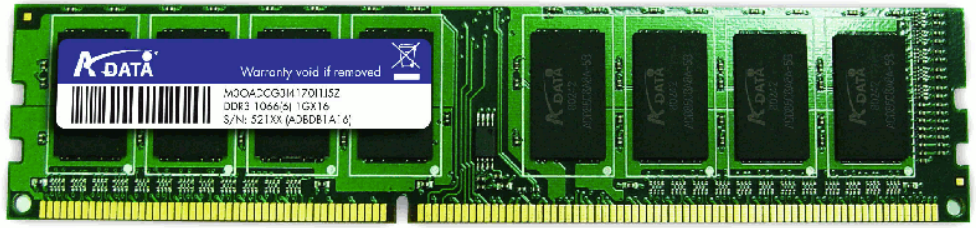
Evolution de la DRAM (2/2)

La DDR-I ou DDR-SDRAM (Double Data Rate Synchronous DRAM, 2000): La DDR-SDRAM permet de recevoir ou d'envoyer des données lors du front montant et du front descendant de l'horloge. (133 à 200 MHz).

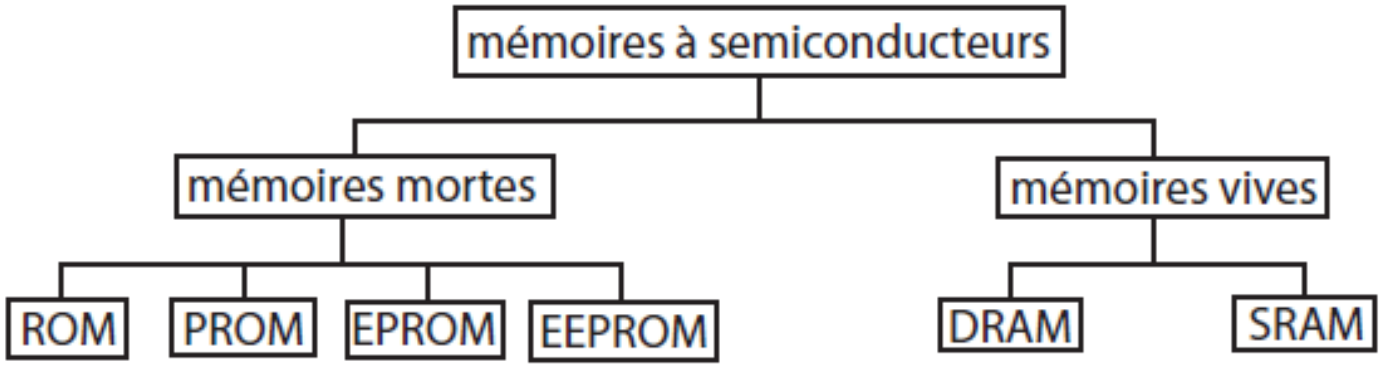
La **DDR2 SDRAM** ou **DDR2** (***Double Data Rate two Synchronous DRAM, 2003***): La différence majeure entre la DDR et la DDR2 est que la fréquence du bus est double de celle du groupe de cellules mémoires. Quatre mots de données peuvent ainsi être transférés par cycle des cellules mémoires. À fréquence des cellules mémoires égale, la DDR2 a un débit deux fois plus élevé que celui de la DDR.



DDR3 SDRAM ou DDR3 (*Double Data Rate 3rd generation Synchronous DRAM*, 2007*):* Le standard DDR3 a été élaboré dans le but de succéder au standard DDR2, en offrant des améliorations de performances tout en diminuant la consommation électrique.



Classification des Mémoires (2/2)



Mémoires Mortes

Pour certaines applications, il est nécessaire de pouvoir conserver des informations de façon permanente même lorsque l'alimentation électrique est interrompue. On utilise alors des mémoires ROM.

Ces mémoires, contrairement aux RAM, ne peuvent être que lue. L'inscription en mémoire des données reste possible mais est appelée programmation. Suivant le type de ROM, la méthode de programmation change. Il existe donc plusieurs types de ROM :

- ✓ ROM
- ✓ PROM
- ✓ EPROM
- ✓ EEPROM
- ✓ FLASH EPROM.

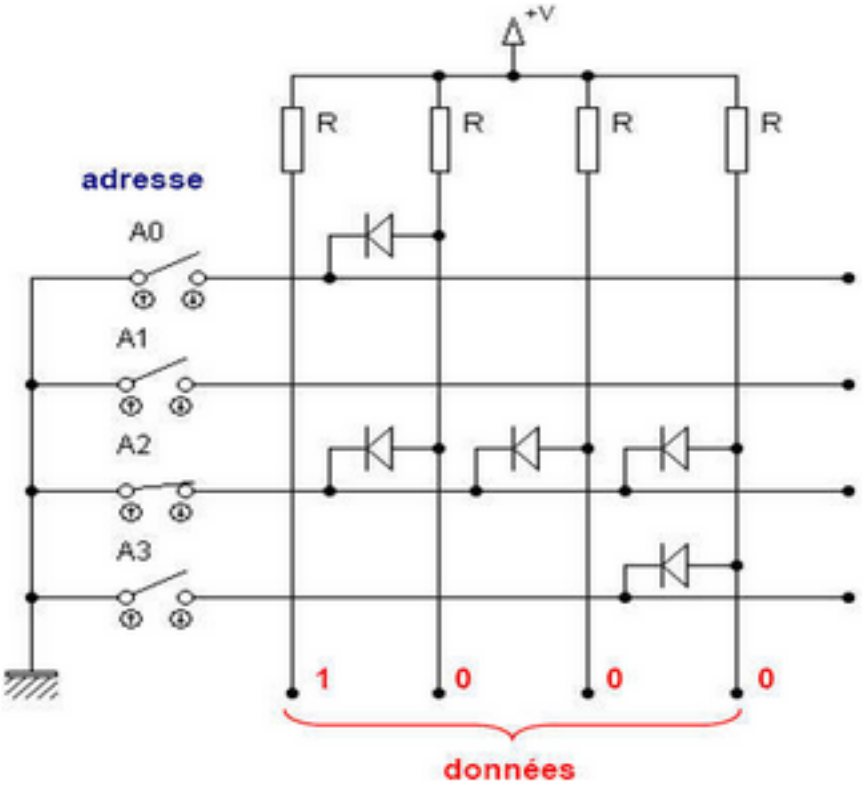
ROM

- **Structure**

Cette mémoire est composée d'une matrice dont la programmation s'effectue en reliant les lignes aux colonnes par des diodes. L'adresse permet de sélectionner une ligne de la matrice et les données sont alors reçues sur les colonnes (le nombre de colonnes fixant la taille de mots mémoire).

* **Programmation**

L'utilisateur doit fournir au constructeur un masque indiquant les emplacements des diodes dans la matrice.



Avantages	Inconvénients
- Densité élevée - Non volatile - Mémoire rapide	- Écriture impossible - Modification impossible (toute erreur est fatale). - Obligation de grandes quantités en raison du coût élevé qu'entraîne la production du masque et le processus de fabrication.

PROM

C'est une ROM qui peut être programmée une seule fois par l'utilisateur (Programmable ROM). La programmation est réalisée à partir d'un programmateur spécifique.

*** Structure**

Les liaisons à diodes de la ROM sont remplacées par des fusibles pouvant être détruits ou des jonctions pouvant être court-circuitées.

*** Programmation**

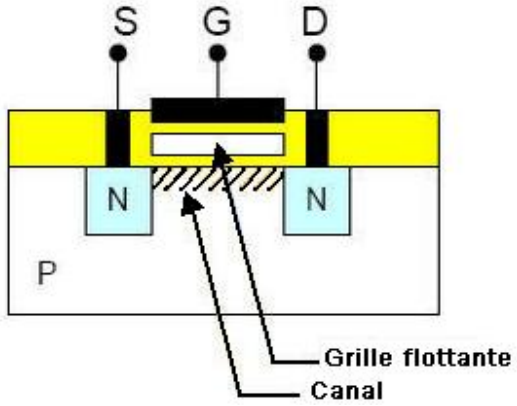
Les PROM à fusible sont livrées avec toutes les lignes connectées aux colonnes (0 en chaque point mémoire). Le processus de programmation consiste donc à programmer les emplacements des ''1'' en générant des impulsions de courants par l'intermédiaire du programmateur ; les fusibles situés aux points mémoires sélectionnés se retrouvant donc détruits. Le principe est identique dans les PROM à jonctions sauf que les lignes et les colonnes sont déconnectées (1 en chaque point mémoire). Le processus de programmation consiste donc à programmer les emplacements des ''0'' en générant des impulsions de courants par l'intermédiaire du programmateur ; les jonctions situées aux points mémoires sélectionnés se retrouvant court-circuitées par effet d'avalanche.

Avantages	Inconvénients
- Idem ROM - Claquage en quelques minutes - Coût relativement faible	- Modification impossible (toute erreur est fatale).

EPROM

*** Structure**

Dans une EPROM, le point mémoire est réalisé à partir d'un transistor FAMOS (Floating gate Avalanche injection Metal Oxyde Silicium). Ce transistor MOS a été introduit par Intel en 1971 et a la particularité de posséder une grille flottante.



*** Programmation**

La programmation consiste à piéger des charges dans la grille flottante. Pour cela, il faut tout d'abord appliquer une très forte tension entre Grille et Source. Si l'on applique ensuite une tension entre D et S, le canal devient conducteur. Mais comme la tension Grille-Source est très importante, les électrons sont déviés du canal vers la grille flottante et capturés par celle-ci. Cette charge se maintient une dizaine d'années en condition normale.

L'exposition d'une vingtaine de minutes à un rayonnement ultraviolet permet d'annuler la charge stockée dans la grille flottante. Cet effacement est reproductible plus d'un millier de fois. Les boîtiers des EPROM se caractérisent donc par la présence d'une petite fenêtre transparente en quartz qui assure le passage des rayons UV. Afin d'éviter toute perte accidentelle de l'information, il faut obturer la fenêtre d'effacement lors de l'utilisation.

Avantages	Inconvénients
- Reprogrammable et non Volatile	- Impossible de sélectionner une seule cellule à effacer - Impossible d'effacer la mémoire in-situ. - l'écriture est beaucoup plus lente que sur une RAM. (environ 1000x)

EEPROM

L'EEPROM (Electrically EPROM) est une mémoire programmable et effaçable électriquement. Elle répond ainsi à l'inconvénient principal de l'EPROM et peut être programmée in situ.

- **Structure**

Dans une EEPROM, le point mémoire est réalisé à partir d'un transistor SAMOS reprenant le même principe que le FAMOS sauf que l'épaisseur entre les deux grilles est beaucoup plus faible.

- **Programmation**

Une forte tension électrique appliquée entre grille et source conduit à la programmation de la mémoire. Une forte tension inverse provoquera la libération des électrons et donc l'effacement de la mémoire.

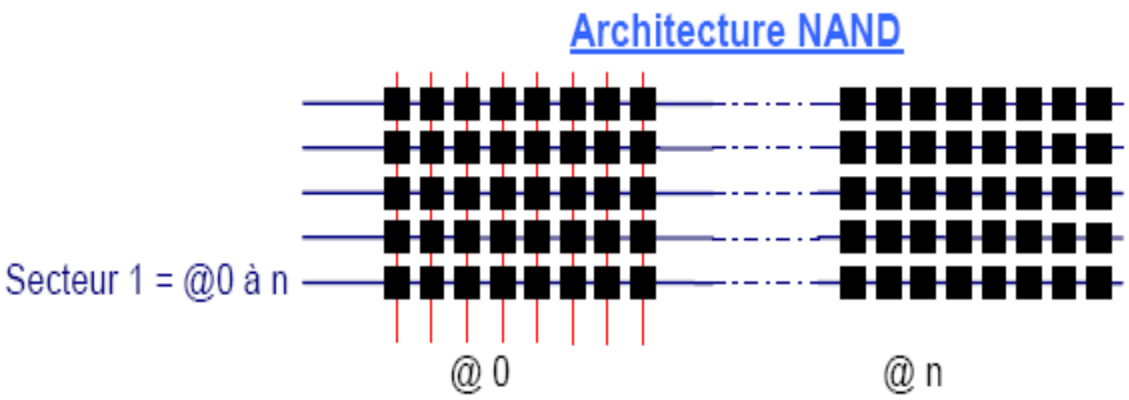
Avantages	Inconvénients
- Comportement d'une RAM non Volatile. - Programmation et effacement mot par mot possible.	- Très lente pour une utilisation en RAM. - Coût de réalisation.

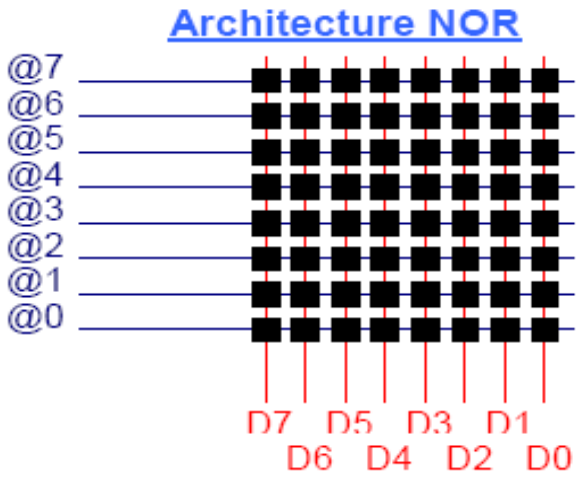
FLASH EPROM (1/2)

La mémoire Flash s'apparente à la technologie de l'EEPROM. Elle est programmable et effaçable électriquement comme les EEPROM.

*** Structure**

Il existe deux technologies différentes qui se différencient par l'organisation de leurs réseaux mémoire : L'architecture NOR et NAND.
L'architecture NOR propose un assemblage des cellules élémentaires de mémorisation en parallèle avec les lignes de sélection comme dans une EEPROM classique. L'architecture NAND propose un assemblage en série de ces mêmes cellules avec les lignes de sélection. D'un point de vue pratique, la différence majeure entre NOR et NAND tient à leurs interfaces. Alors qu'une NOR dispose de bus d'adresses et de données dédiés, la NAND est dotée d'une interface d'E/S indirecte. Par contre, la structure NAND autorise une implantation plus dense grâce à une taille de cellule approximativement 40 % plus peti





FLASH EPROM (2/2)

*** Programmation**

Si NOR et NAND exploitent toutes les deux le même principe de stockage de charges dans la grille flottante d'un transistor, l'organisation de leur réseau mémoire n'offre pas la même souplesse d'utilisation.
Les Flash NOR autorisent un adressage aléatoire qui permet de la programmer octet par octet alors que la Flash NAND autorise un accès séquentiel aux données et permettra seulement une programmation par secteur comme sur un disque dur.

	Avantages	Inconvénients
Flash NOR	- Comportement d'une RAM non Volatile. - Programmation et effacement mot par mot possible. - Temps d'accès faible.	- Lenteur de l'écriture/lecture par paquet. - Coût.
Flash NAND	- Comportement d'une RAM non Volatile. - Forte densité d'intégration - Coût réduit. - Rapidité de l'écriture/lecture par paquet - Consommation réduite.	- Ecriture/lecture par octet impossible. - Interface E/S indirecte

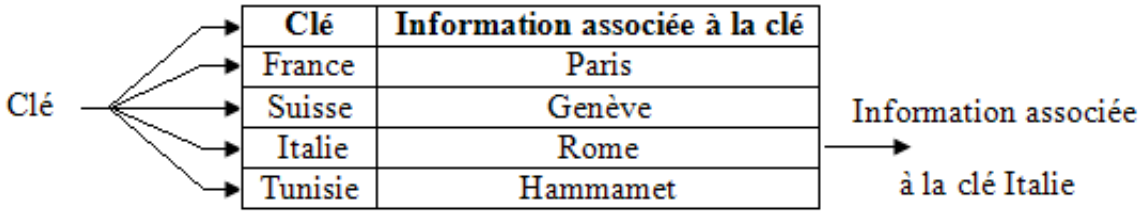
Les Mémoires Caches

Besoin

Pour résoudre le problème de la trop grande différence de vitesse entre le CPU et la mémoire centrale, on a recours aux mémoires caches ou antémémoires : la solution consiste à insérer entre les deux une mémoire très rapide (vitesse d'accès de 15 ns au moins, de type SRAM) mais pas très grande qui va contenir les informations (instructions, données) dont a besoin le CPU. **Cette mémoire ne fait pas partie de la mémoire centrale**.

Types d'accès

L'antémémoire est une mémoire associative, ce qui signifie que les informations ne sont pas accessibles par une adresse, ce qui est le cas dans la mémoire centrale, mais sont adressables par le contenu. Chaque case d'une mémoire associative comprend deux champs correspondant à la clé et à l'information associée à cette clé.



Dans le cas de l'antémémoire, la clé est constituée par **l'adresse en mémoire centrale** de l'instruction ou de la donnée cherchée, et l'information associée est constituée de **l'instruction ou de la donnée elle-même**.

La recherche par clé dans la mémoire associative ne s'effectue que de manière séquentielle, mais en parallèle sur toutes les cases de la mémoire associative. En un seul accès, on sait si l'instruction cherchée se trouve ou non dans l'antémémoire.

Le Principe de Localité

Le bon fonctionnement des caches est basé sur le ***principe de localité*** qui dit que le code et les données des programmes ne sont pas utilisées de manière uniforme. On constate souvent que **10%** du code d'un programme contribue à 90% des instructions exécutées.

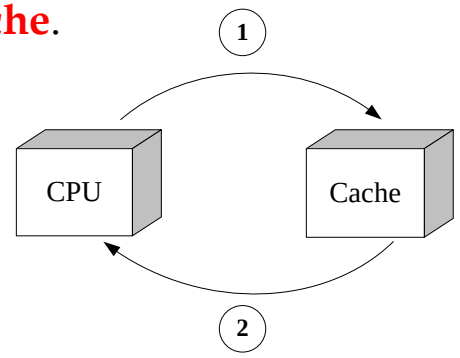
On distingue deux types de localité:

- La ***localité temporelle*** indique que des éléments auxquels on a eu accès récemment seront probablement utilisés dans un futur proche (il y a plus de chances d'accéder à une position de mémoire utilisée il y a 10 cycles qu'à une autre utilisée il y a 10000 cycles).

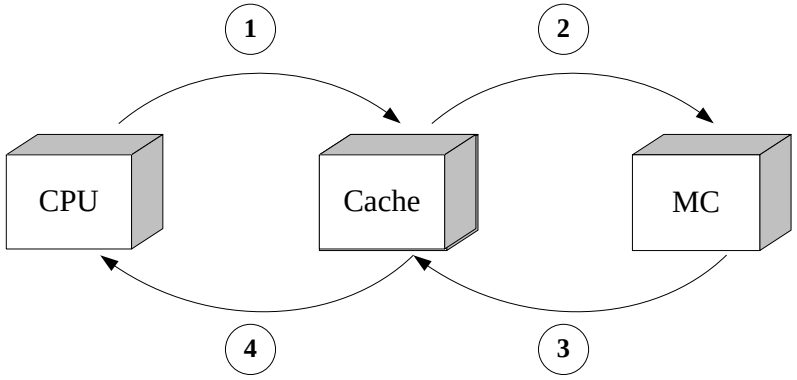
- La ***localité spatiale*** indique que des éléments proches ont tendances à être référencés à des instants proches (comme des blocs correspondant à des boucles ou/et des sous-programmes).

Succès et Défaut de Cache

En pratique, le processeur ne peut pas savoir à l'avance si la donnée qu'il cherche se trouve ou non dans la mémoire cache et la copie de la donnée à partir de la MC ne peut donc pas être effectuée de façon préventive. Une stratégie de tentative/échec est alors envisagée :
Soit la donnée ou l'instruction est présente dans le cache et elle est alors envoyée directement au CPU ➔ on parle de **succès de cache**.



Soit la donnée ou instruction n'est pas présente dans le cache➔ un **défaut de cache** a lieu. Ce défaut de cache déclenche alors une action prédéterminée, qui consiste à charger une page mémoire, dont le rôle est de faire la copie d'une partie de données (page) de la mémoire principale vers la mémoire cache, par la suite la donnée peut être lue par le processeur.



Caches L1, L2 et L3

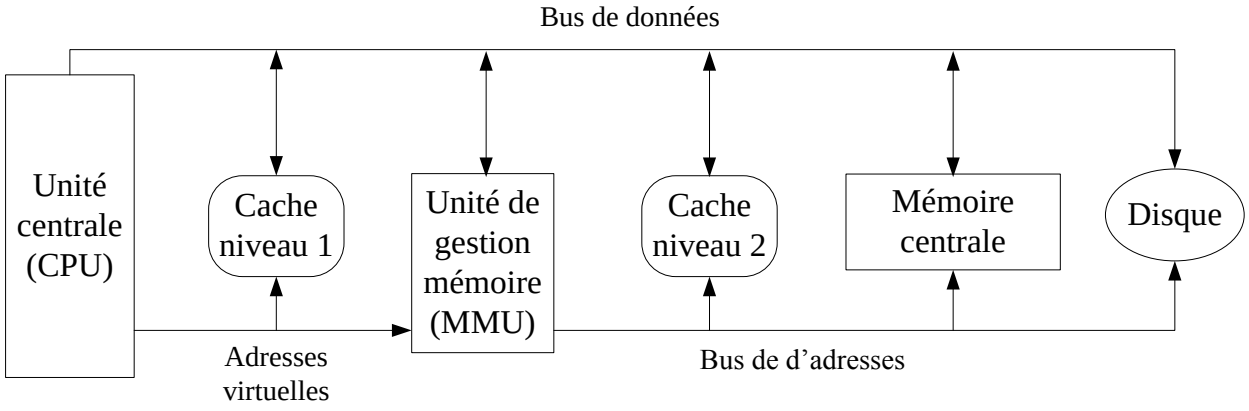
Les ordinateurs récents possèdent plusieurs niveaux de mémoire cache :

- ✓ La **mémoire cache de premier niveau** (appelée **L1 Cache**, pour **Level 1 Cache**) est directement intégrée dans le processeur. Elle se subdivise en 2 parties :
  - La première est le **cache d'instructions**, qui contient les instructions issues de la mémoire vive.
  - La seconde est le **cache de données**, qui contient des données issues de la mémoire vive et les données récemment utilisées lors des opérations du processeur.

  Les caches du premier niveau sont très rapides d'accès. Leur délai d'accès tend à s'approcher de celui des registres internes aux processeurs.

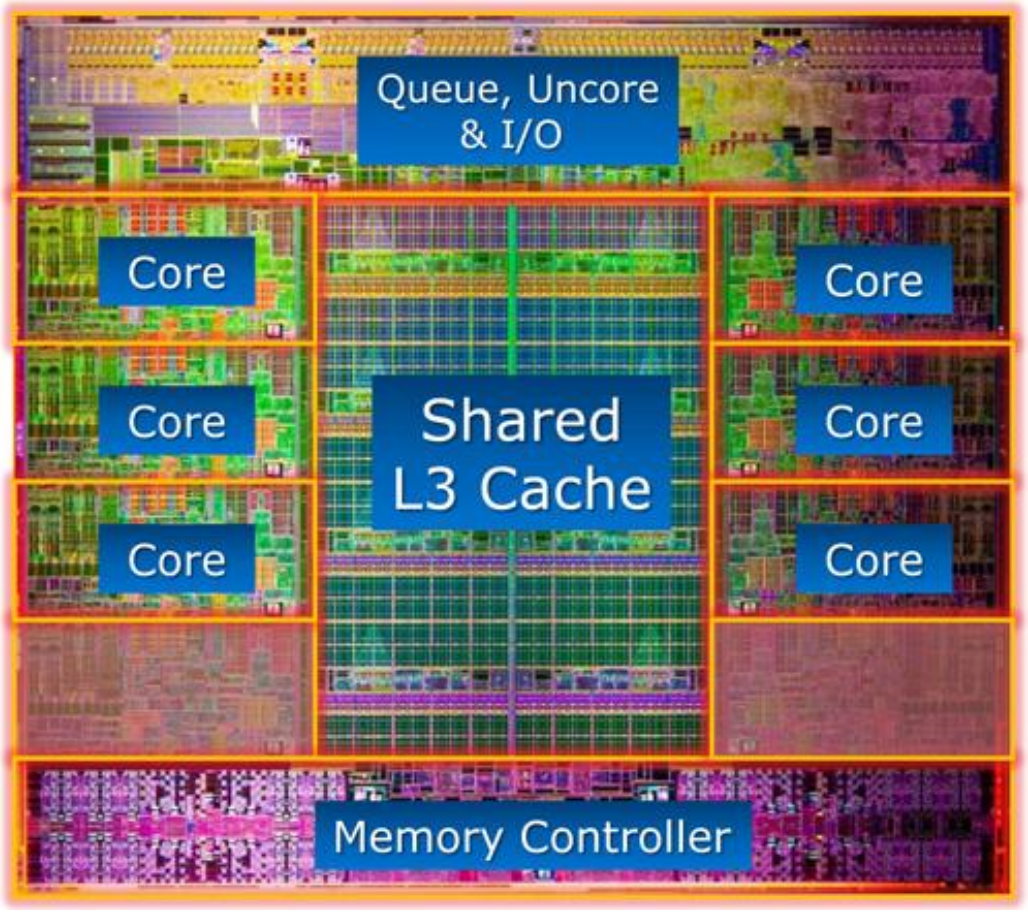
- ✓ La **mémoire cache de second niveau** (appelée **L2 Cache**, pour **Level 2 Cache**) est située au niveau du boîtier contenant le processeur (dans la puce). Le cache de second niveau vient s'intercaler entre le processeur avec son cache interne et la mémoire vive. Il est plus rapide d'accès que cette dernière mais moins rapide que le cache de premier niveau.

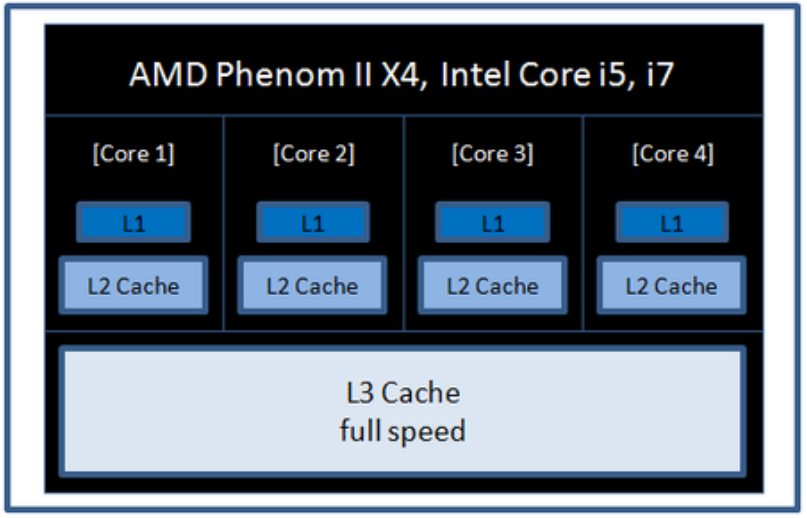
- ✓ La **mémoire cache de troisième niveau** (appelée **L3 Cache**, pour **Level 3 Cache**) est située au niveau de la carte mère.



Caches L1, L2 et L3

Intel ® Core ™ i7-3960x





Cache L1: 6 × 64 KO
Cache L2: 6 × 256 KO
Cache L3 de taille 16 MO

Les Mémoires Auxiliaires

Les mémoires auxiliaires, appelés aussi **secondaires** ou **périphériques**, sont des mémoires permettant le stockage permanant d'un très grand nombre d'informations. Elles sont composées principalement de disques (magnétiques, magnéto-optique, optiques), de bandes magnétiques ou de cartouches magnétiques. On trouve deux catégories de mémoires auxiliaires : les mémoires fixes et les mémoires amovibles. Les mémoires fixes sont les disques durs magnétiques fixes. Les dispositifs amovibles (bandes, disques) servent généralement de mémoire d'archivage.

Les disques magnétiques servent de mémoire pour le support des fichiers et n'offrent qu'une capacité limitée de quelques Gbytes, chaque unité de disque contenant un ou plusieurs disques fixes. Les mémoires d'archivage amovibles, telles que le **disque dur**, **bandes**, **cartouches magnétique** ou **disques optiques**, offrent de plus grandes capacités de stockage, chaque unité n'étant pas liée à un seul disque ou cartouche. On peut les changer à volonté.

L'accès à ces mémoires est plus long que l'accès aux disques fixes. Elles sont généralement utilisées pour sauvegarder le contenu des disques.