

Jean-Marc Talbot

jtalbot@cmi.univ-mrs.fr



Mémoires

Mémoires

mémoire : dispositif permettant d'enregistrer, de conserver et de restituer de l'information.

- unité de base : 1 **bit** (0 ou 1)
- **octet** (byte) = groupe de 8 bits
- **mot** = regroupement d'octets (8 bits, 16 bits, 32 bits, ...)
 - ▶ unité d'information adressable en mémoire
- un **KiloOctet** = 2^{10} octets = 1024 octes = 1 Ko
- un **MegaOctet** = 2^{10} Ko = 1 Mo
- un **GigaOctet** = 2^{10} Mo = 1 Go
- un **TeraOctet** = 2^{10} Go = 1 To

Mémoires (II)

- différentes technologies :

Electronique - Magnétique - Optique

- différentes caractéristiques :

- ▶ capacité
- ▶ temps d'accès
- ▶ débit
- ▶ volatilité

Performances des mémoires

- **temps d'accès** : temps qui sépare une demande de lecture/écriture et sa réalisation t_a
- **temps de cycle** : temps minimum entre deux accès à la mémoire t_c

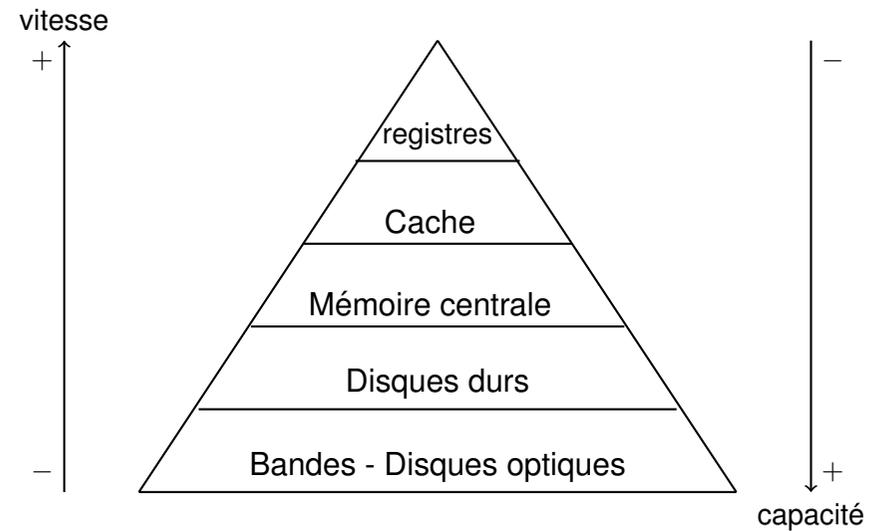
On a $t_a < t_c$ (stabilisation des signaux, synchronisation, ...)

- **débit** (ou **bande passante**) : nombre de bits maximum transmis par seconde

en cas d'accès en temps uniforme au données

$$B = \frac{n}{t_c} \quad n \text{ est le nombre de bits transférés par cycle}$$

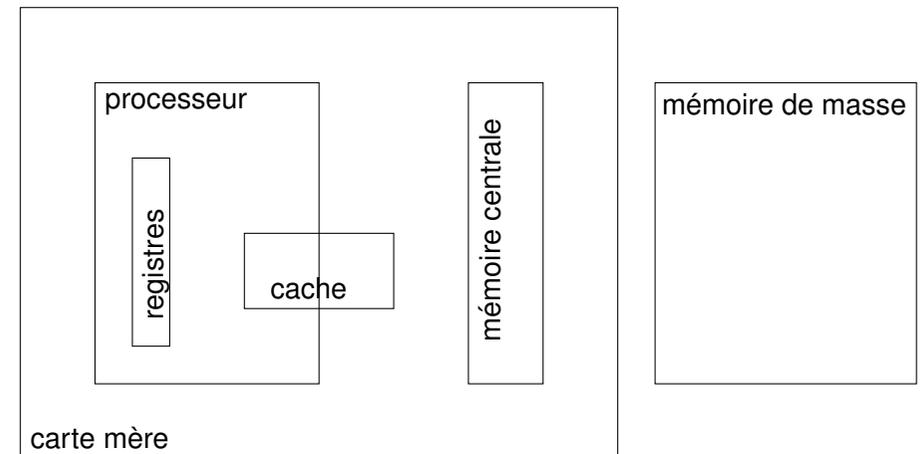
Types de mémoires (I)



Types de mémoires (II)

	vitesse (temps d'accès)	vitesse (débit)	capacité
registres	< 1 ns	> 50 Go/s	< 100 octets
cache	2 - 5 ns	5 - 20 Go/s	100 Ko - 1 Mo
mémoire centrale	20 ns	1 Go/s	256 Mo - 4 Go
disque dur	1-10 ms	300 Mo/s	50 Go - 500 Go

Mémoires : localisations



Méthodes d'accès

- Accès séquentiel
 - ▶ pour accéder à une information, il faut parcourir toutes les informations qui la précède
 - ▶ exemple : bandes magnétiques
- Accès direct
 - ▶ chaque information possède une adresse propre, à laquelle on peut accéder directement
 - ▶ exemple : mémoire centrale de l'ordinateur
- Accès semi-séquentiel
 - ▶ intermédiaire entre séquentiel et direct
 - ▶ exemple : disque dur
 - ★ accès direct au cylindre
 - ★ accès séquentiel au secteur sur un cylindre
- Accès associatif
 - ▶ une information est identifiée par sa clé
 - ▶ on accède à une information via sa clé
 - ▶ exemple : mémoire cache

Mémoires mortes (ROM)

- **ROM (Read Only Memory)** : information stockée au moment de la conception du circuit.
- **PROM (Programmable Read Only Memory)** : mémoire programmable par un utilisateur final mais une seule fois
- **EPROM (Erasable Programmable Read Only Memory)** : mémoire (re)programmable et effaçable par ultraviolet.
- **EEPROM (Electrically Erasable Programmable Read Only Memory)** : mémoire (re)programmable et effaçable électriquement.
 - ▶ exemple : Bios ("flashable") - lecteur MP3

Types de mémoire

Sur la carte mère cohabite

- Mémoires persistantes (mémoires mortes - ROM)
 - ▶ leur contenu est fixé (ou presque)
 - ▶ et conservé en permanence même hors alimentation électrique
- Mémoires volatiles (mémoires vives - RAM)
 - ▶ leur contenu est modifiable
 - ▶ et perdu hors alimentation électrique

ROM = Read Only Memory

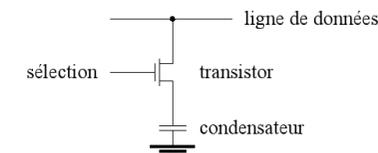
RAM = Random Access Memory

Mémoires volatiles (RAM) (I)

2 types de RAM

DRAM : Dynamic RAM

- Dynamique : l'information doit être périodiquement rafraîchie
- réalisation :
 - ▶ 1 bit = 1 transistor + 1 condensateur
 - ▶ le condensateur stocke l'information
 - ▶ Doit être rafraîchi régulièrement (pour conserver la valeur stockée dans le condensateur)
 - ★ ralentit la vitesse d'accès à la mémoire
- peu coûteuse



Mémoires volatiles (RAM) (II)

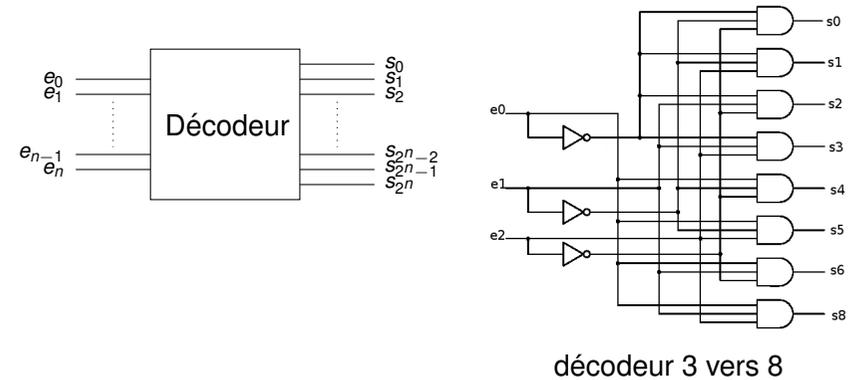
2 types de RAM

SRAM : Static RAM

- Statique : l'information n'a pas besoin d'être rafraîchie
- réalisation :
 - ▶ 1 bit = 4 transistors = 2 portes NOR
 - ▶ Bascule RS (ou D) qui stocke l'information
 - ★ beaucoup plus **rapide** que la DRAM
- beaucoup plus **cher** que la DRAM

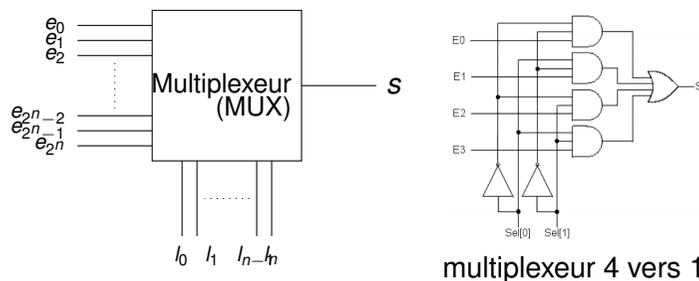
Décodeurs - multiplexeurs : rappels (I)

- Un **décodeur** permet de **décoder un mot binaire** : il comprend n entrées et 2^n sorties.
- la i ème sortie de décodeur vaut 1 si les n entrées forment l'entier binaire i .



Décodeurs - multiplexeurs : rappels (II)

- Un **multiplexeur** comporte 2^n entrées, 1 sortie et n lignes de sélection (entrées).
- la sortie du multiplexeur vaut la valeur de la i ème entrée si l'entier i est codé sur les lignes de sélection.



Registres (I)

- mémoire de type SRAM
- intégrés au cœur du processeur
- un registre stocke les informations relatives à une instruction
 - ▶ opérandes nécessaires à l'instruction
 - ▶ résultats produits par l'instruction
- très peu nombreux (< 20)
- très rapides (cadencés à la vitesse du processeur)

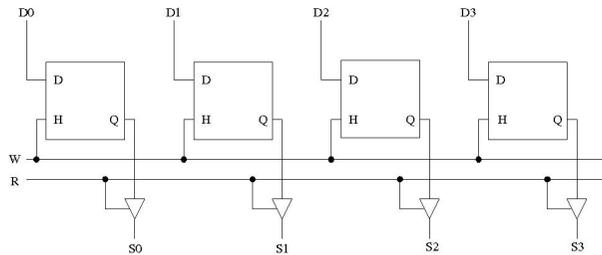
Nous en reparlerons lors de l'étude du processeur

Registres (II)

Réalisation :

- registre 1-bit = 1 bascule RS (ou D)
- registre n -bits = n bascules RS (ou D) en parallèle

registre 4 bits



Bancs de registres (I)

Un **banc de registres** de hauteur n et de largeur k est

- un ensemble de n registres de k bits
- une mémoire de n adresses

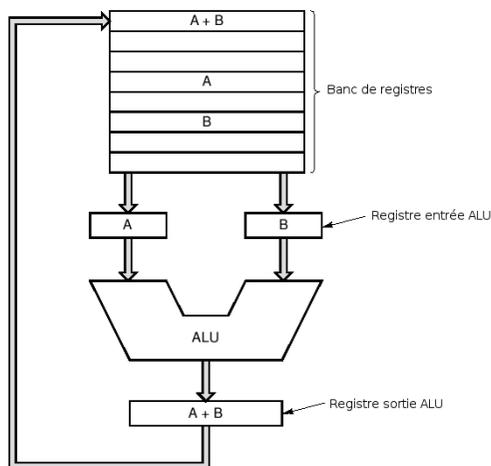
Exemple :

Une banc de n registres de 3 bits avec

- un port d'écriture
- deux ports de lecture (lors d'une lecture le contenu de deux registres est lu même temps.)

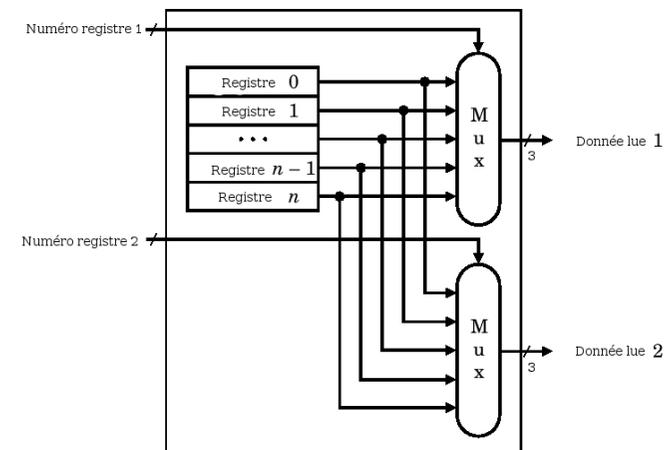
Bancs de registres (II)

Exemple d'utilisation :



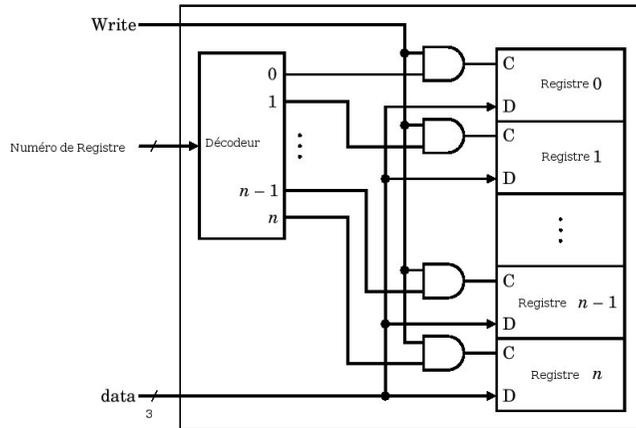
Bancs de registres (III)

Fonctionnement en lecture : (2 registres en parallèle)

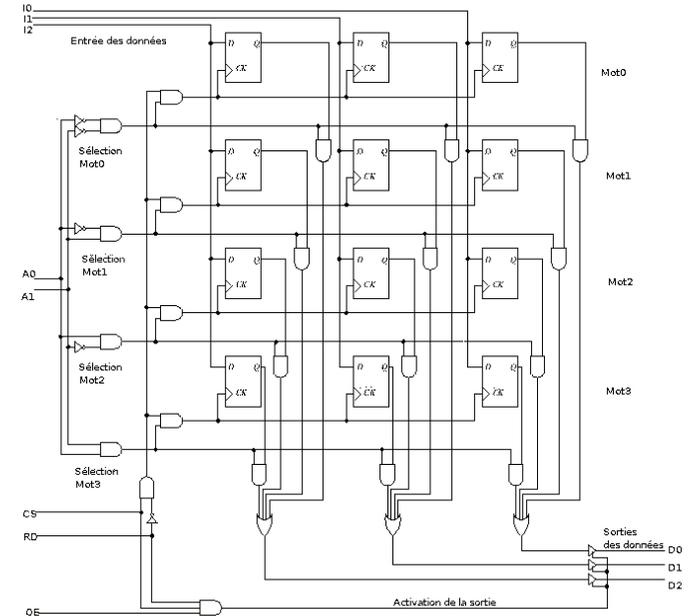


Bancs de registres (IV)

Fonctionnement en écriture :



Bancs de registres (V)



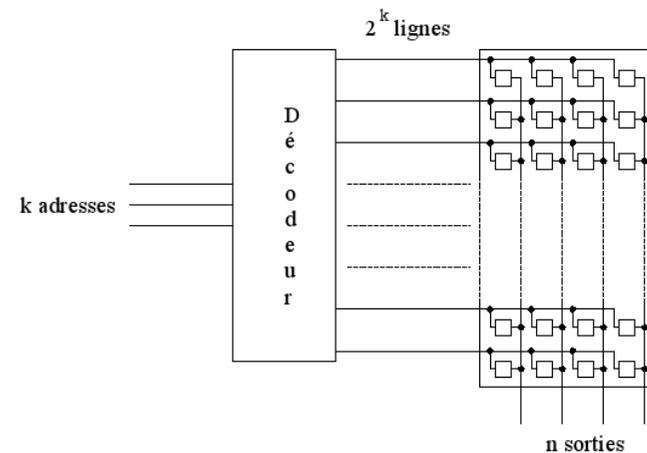
Mémoire centrale

Rappels :

- Mémoire de type DRAM
- l'information y est stockée comme des mots (mémoire) d'un certain nombre de bits (8, 16, 32, 64 bits) : la longueur des mots est le **format** de la mémoire.
- vitesse relativement lente (comparée à celle du processeur)

Organisation de la Mémoire centrale

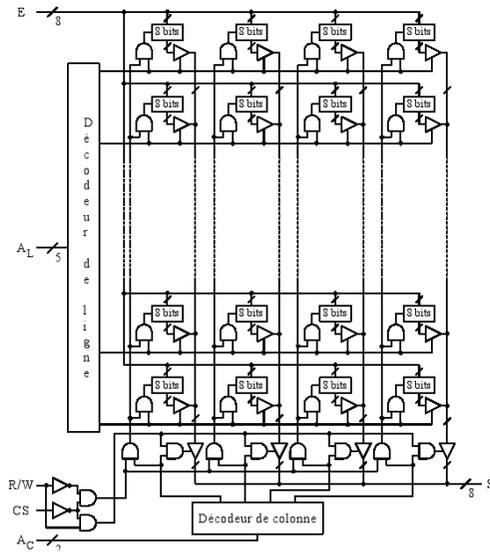
Mémoire unidimensionnelle



Nombre de portes dans le décodeur trop important

Organisation de la Mémoire centrale (II)

Mémoire bidimensionnelle utilisation d'un décodeur pour les lignes et un pour les colonnes

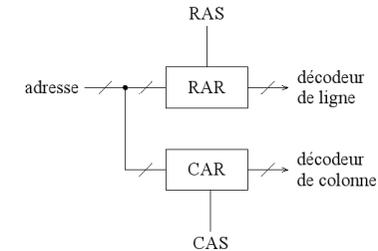


Organisation de la Mémoire centrale (III)

Mémoire matricielle (organisée comme une matrice carrée)

Utilisation des mêmes lignes d'adresses utilisées pour les lignes et les colonnes.

Chargement de l'adresse en deux temps et stocké dans deux registres

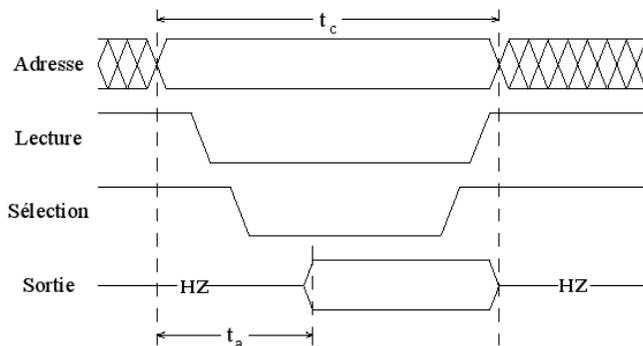


- **CAS** : Column Address Strobe
- **RAS** : Row Address Strobe

Fonctionnement de la mémoire (I)

Cycle de lecture

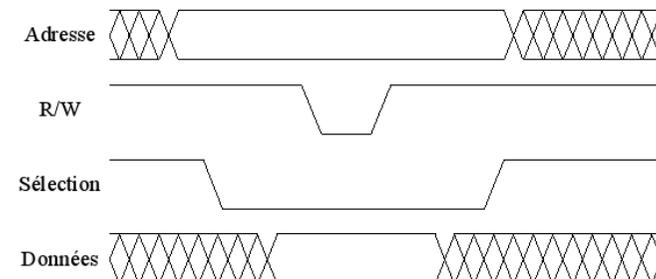
- établissement de l'adresse
- signal de lecture (R/W=0 par exemple)
- sélection du boîtier (CS=0)
- Après un certain temps, l'information apparaît sur la sortie et reste présente jusqu'à la fin du cycle.



Fonctionnement de la mémoire (II)

Cycle d'écriture

- établissement de l'adresse
- sélection du boîtier (CS=0)
- établissement de la donnée sur l'entrée
- signal d'écriture (R/W=0 ci-dessous)



Fonctionnement de la mémoire (III)

Protocoles échanges processeur-mémoire

- **Synchrone** : au bout de k unités de temps, le processeur suppose que l'opération sur la mémoire a été réalisée (mot écrit en mémoire, mot lu disponible sur la sortie)
- **Asynchrone (handshaking)** : processeur et mémoire s'échangent des informations de contrôle (**request/acknowledgment**)

Assemblage de boîtiers mémoire (I)

Les mémoires sont regroupés dans des **boîtiers mémoire**.

Du aux limites technologiques d'intégration, pour obtenir des mémoires de grandes tailles, on associe plusieurs boîtiers mémoires.

Ces blocs sont assemblés :

- pour augmenter la taille des mots de la mémoire
- pour augmenter le nombre de mots dans la mémoire

Fonctionnement de la mémoire (IV)

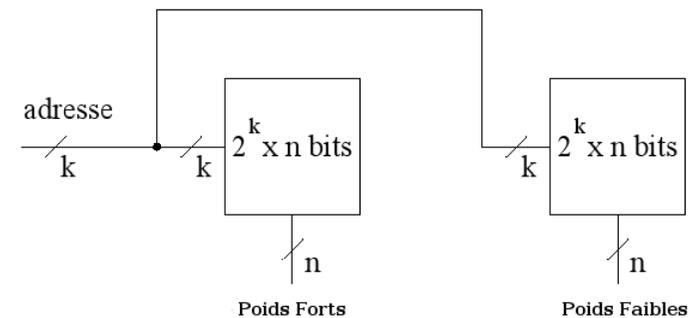
Optimisations

- mémoire synchrone (synchronisée avec le bus) : **SDRAM**
- Pour les mémoires matriciels, **accès en mode page** : on charge ligne et colonne, puis on ne change que les colonnes pour les accès suivants (localité des données) : **DRAM FPM**
- Pour les mémoires matriciels, **accès en rafale (burst)** : on charge ligne et colonne ainsi que le nombre de données à lire ; incrémentation dans la mémoire des colonnes pour les accès suivants (localité des données). **DDR-SDRAM**

Assemblage de boîtiers mémoire (II)

Augmentation de la taille des mots

2 boîtiers de 2^k mots de n bits \implies un bloc de 2^k mots de $2n$ bits

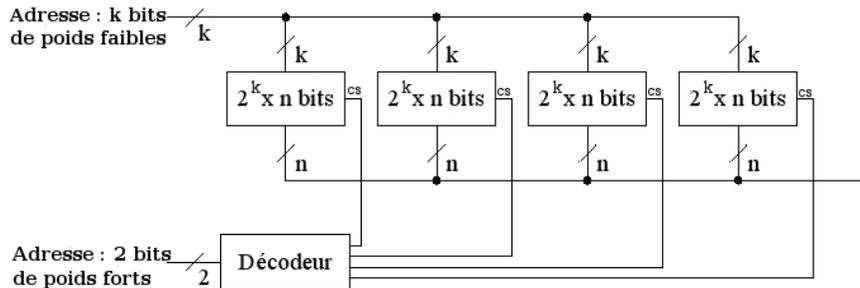


Assemblage de boitiers mémoire (III)

Augmentation du nombres de mots

4 boitiers mémoires de 2^k mots de n bits \implies un bloc de $4 * 2^k$ mots de n bits

adresse pour le nouveau bloc sur $k + 2$ bits



Mémoire et erreurs

Du fait de sa nature “physique”, les informations en mémoire peuvent comporter une ou des erreurs.

Pour détecter et corriger, on ajoute des **bits de contrôle**.

- **bit de parité** : 1 bit supplémentaire (en plus des bits de données) tel que le nombre de bits à 1 est pair
- **mémoire ECC** (Error Correction Coding) possède des bits supplémentaires pour détecter et corriger le(s) bit(s) erroné(s).

Mémoire Logique (I)

La mémoire logique est la façon dont le processeur (ou le programmeur) voit la mémoire (physique).

La mémoire est définie comme un ensemble de N octets consécutifs dont

- la première adresse est 0
- la dernière adresse est $N - 1$

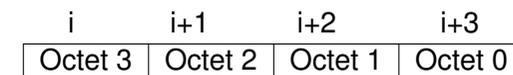
Adressage de la mémoire par des mots de : 8 (octet), 16, 32, 64, .. bits.

Un mot de 32 bits est constitué de 4 octets consécutifs

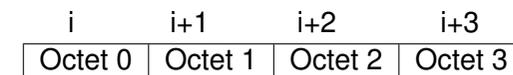
Mémoire Logique (II)

Pour un mot mémoire de 32 bits, il existe 2 façons de ranger les octets qui le compose :

- Le mot de poids fort est stocké en premier : **big-endian**



- Le mot de poids fort est stocké en dernier : **little-endian**



Un mot mémoire ne peut commencer n'importe où

- les mots de 16 bits commencent sur des adresses paires
- les mots de 32 bits commencent sur des adresses multiples de 4

Mémoire Logique (III)

Segmentation de la mémoire : découpage logique de la mémoire en un certain certains nombres de blocs (ou segments)

Une adresse est codée comme

- un numéro de blocs
- un déplacement dans le bloc (**offset**)

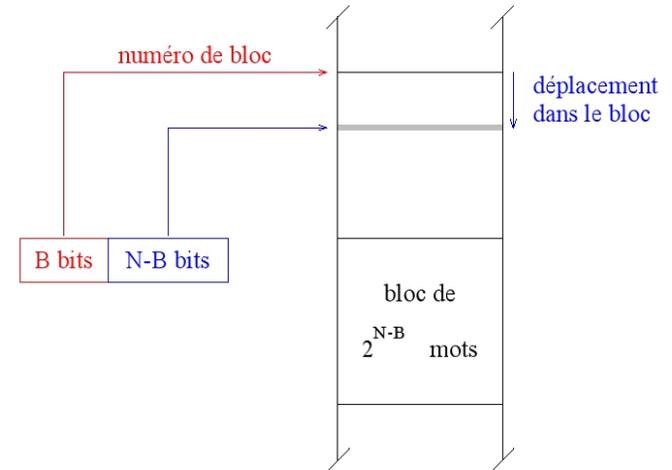
N bits d'adresses $\implies 2^N$ cases mémoire.

Les N bits sont séparés en deux :

- B bits (de poids fort) pour le numéro de bloc
- $N - B$ bits (de poids faible) pour le déplacement dans le bloc

Mémoire Logique (IV)

On a donc 2^B blocs ayant chacun 2^{N-B} cases.



Mémoire Virtuelle

La mémoire physique n'est qu'une partie de la mémoire disponible : le système permet l'utilisation de la mémoire de masse (disques durs) comme de la **mémoire virtuelle**.

Pagination de la mémoire virtuelle

Mécanisme de **swap**