



Architecture des Ordinateurs

Chapitre II

(Mémoires)

Mokrani Hocine
dr.mokrani@gmail.com

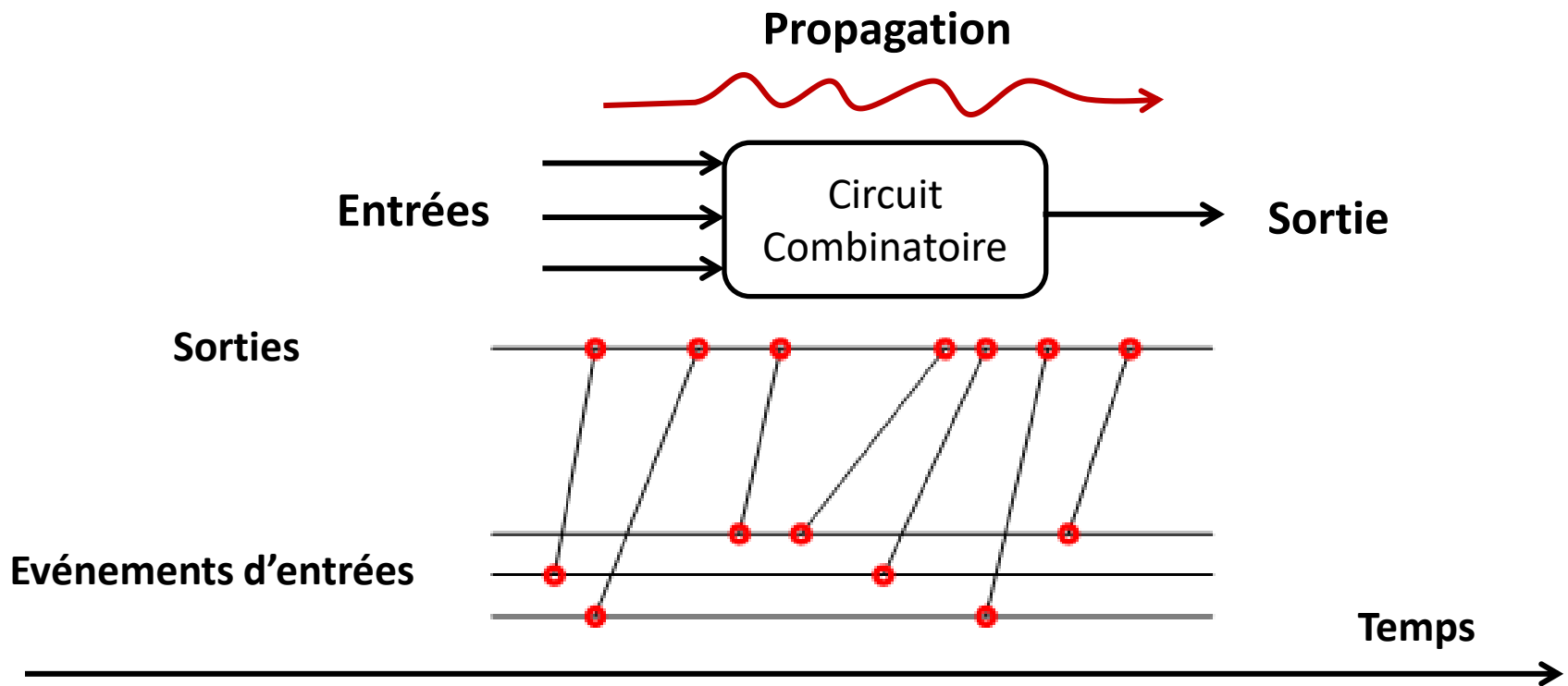
Chapitre II (Mémoires)

- ❑ Logique combinatoire & logique séquentielle.
- ❑ Mémoires.

Logique combinatoire & Logique séquentielle

Logique combinatoire

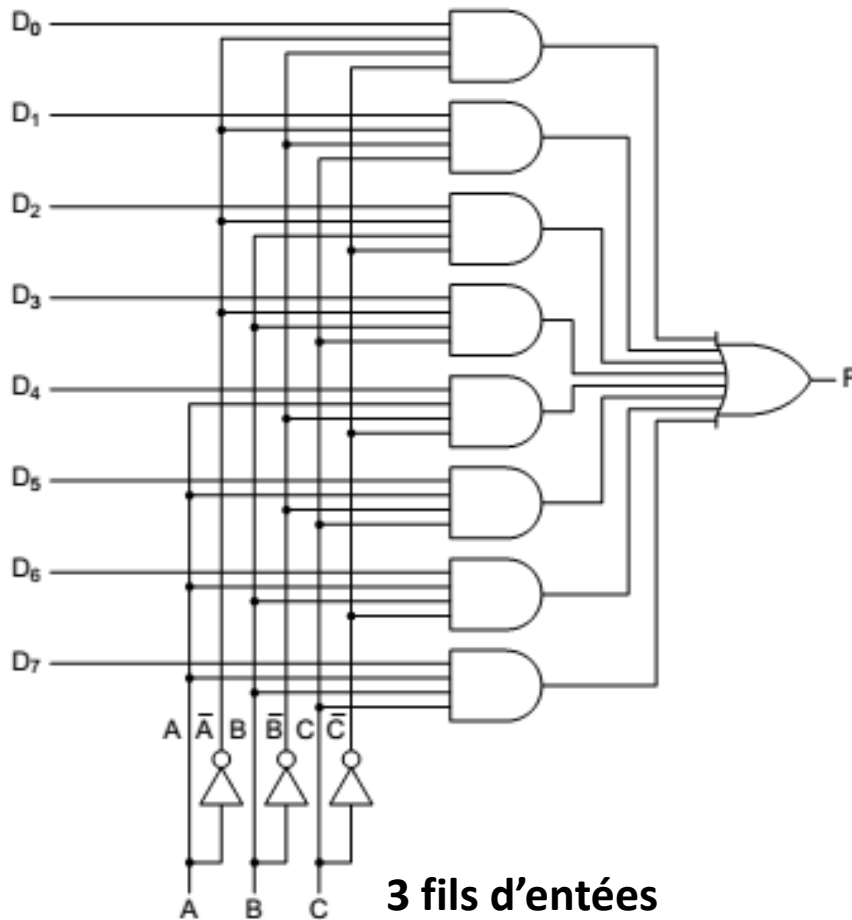
Les échanges des entrées sont directement propagées vers les sorties.



Point important: Attendre que le signal soit stable

Exemples de circuit combinatoire

8 entrées possibles



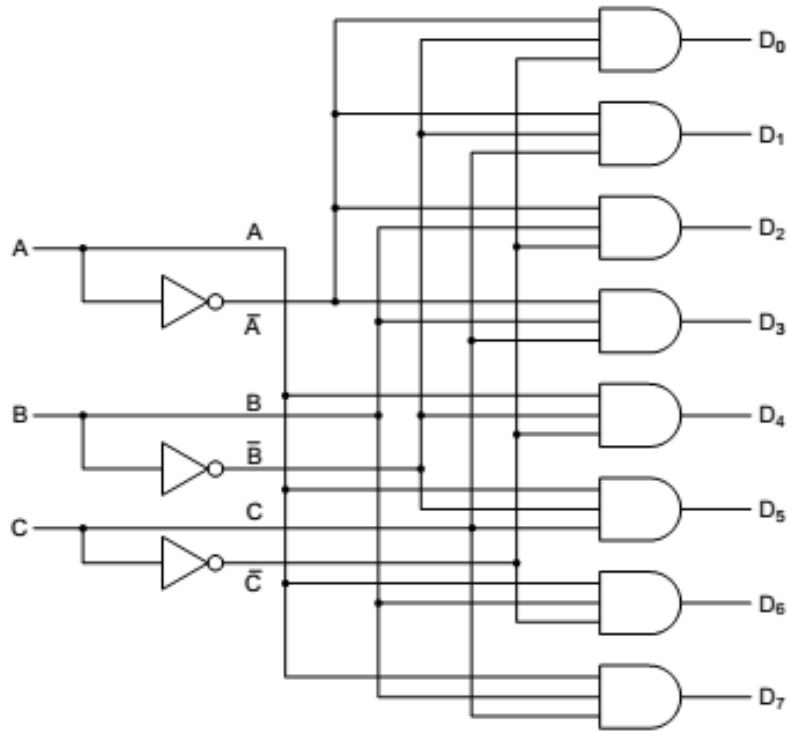
Une Entrée sélectionnée comme sortie

Sélection d'un signal:
Contrôler le cheminement des données

Multiplexeur: sélectionnée une entrée parmi plusieurs.

Exemples de circuit combinatoire

3 Fils d'entrée, 8 possibilités

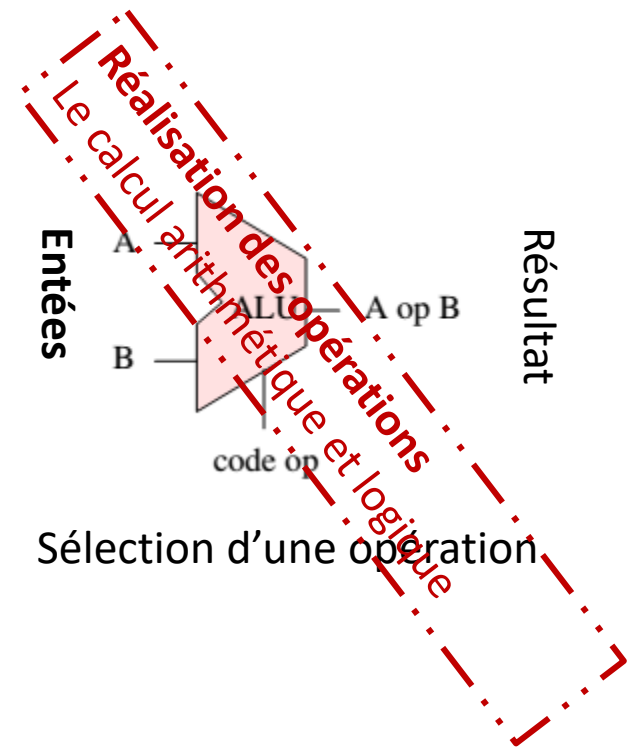
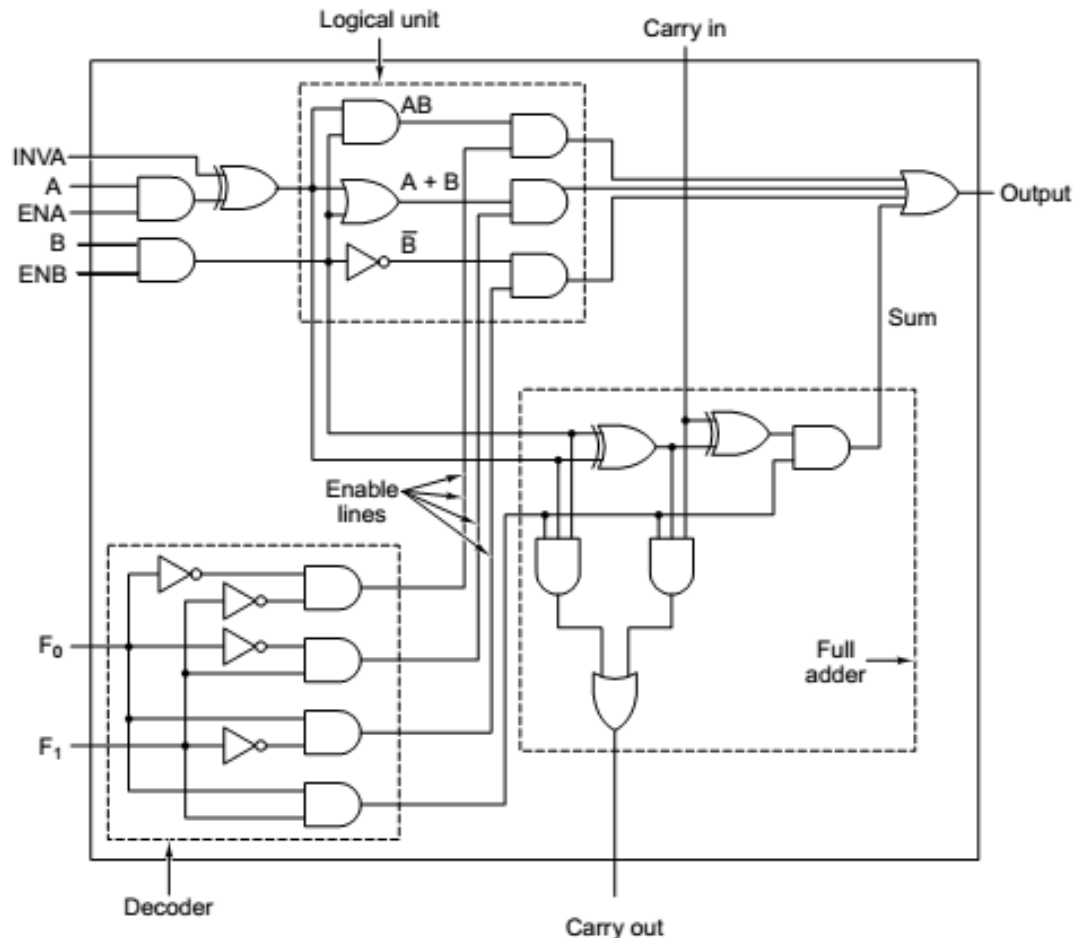


Une sortie est activée à la fois

Sélection Par un nombre:
Utilisable pour les signaux de contrôle

Décodeur: activation d'une sortie parmi plusieurs

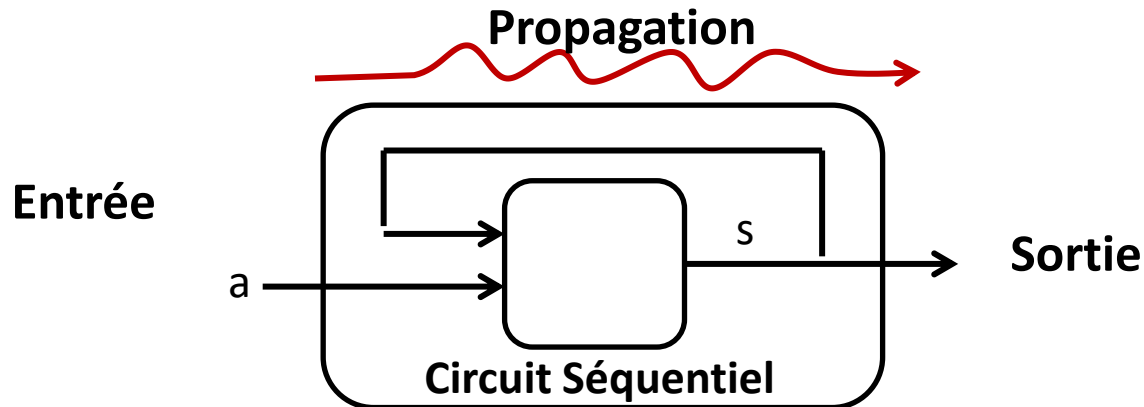
Exemples de circuit combinatoire



ALU: Unité Arithmétique et Logique

Logique séquentielle

Une logique dont les résultats ne dépendent pas seulement des données actuellement traitées mais aussi des données traitées précédemment.



→ Utilisation d'un état mémoire $s_t = F(s_{t-1}, a)$

Fonctionnements synchrone et asynchrone dans un circuit séq.

Les tâches peuvent être effectuées de deux manières.

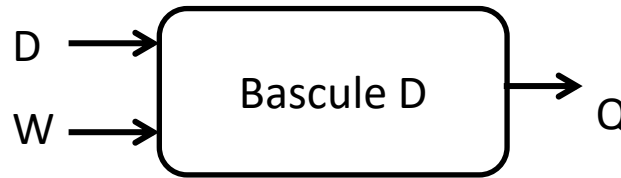
- **Fonctionnement asynchrone:**

La sortie logique peut changer d'état à tout moment quand une ou plusieurs entrées changent.

- **Fonctionnement synchrone:**

Le changement d'état est commandé par un signal d'horloge. Les informations évoluent en fonction du temps.

Exemples de circuit séquentiel

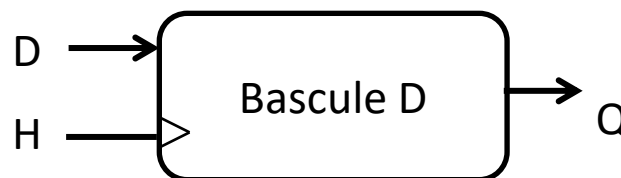


D	W	Q^{t+1}
0	0	Q^t
1	0	Q^t
0	1	0
1	1	1

D= DATA

W= WRITE

- Quand $W = 1$ alors $Q \leftarrow D$. (Cette valeur est mémorisée).
- Quand $W = 0$ alors la bascule reste dans le même état et la sortie vaut la dernière valeur mémorisée.



H	D	Q^{t+1}
0	-	Q^t
1	-	Q^t
\uparrow	0	0
\uparrow	1	1

D= DATA

H= Horloge

Mémoires

Mémoires

- **Définition**

Dispositif permettant d'enregistrer, de conserver et de restituer de l'information.

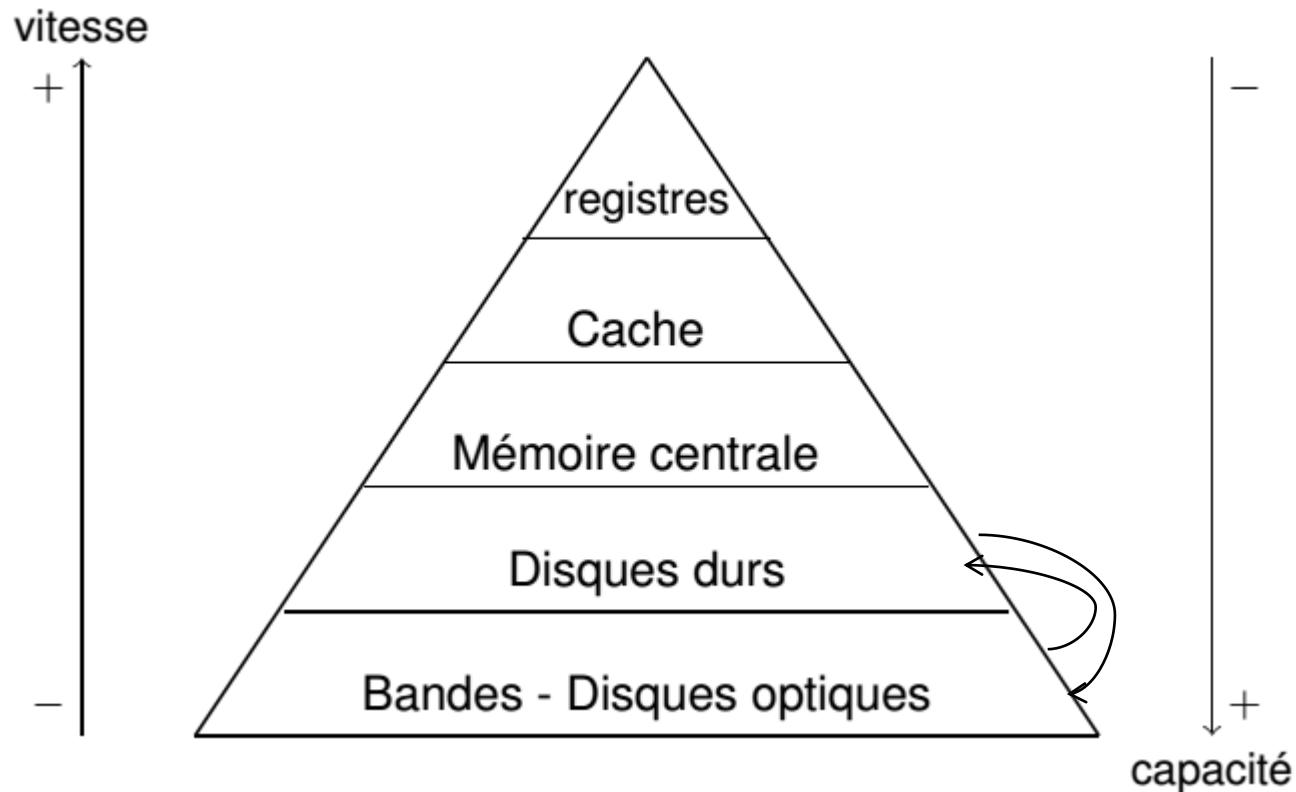
- **Différentes technologies**

Electronique- Magnétique – Optique.

- **Différentes caractéristiques**

- Capacité.
- Temps d'accès.
- Débit.
- Volatilité.

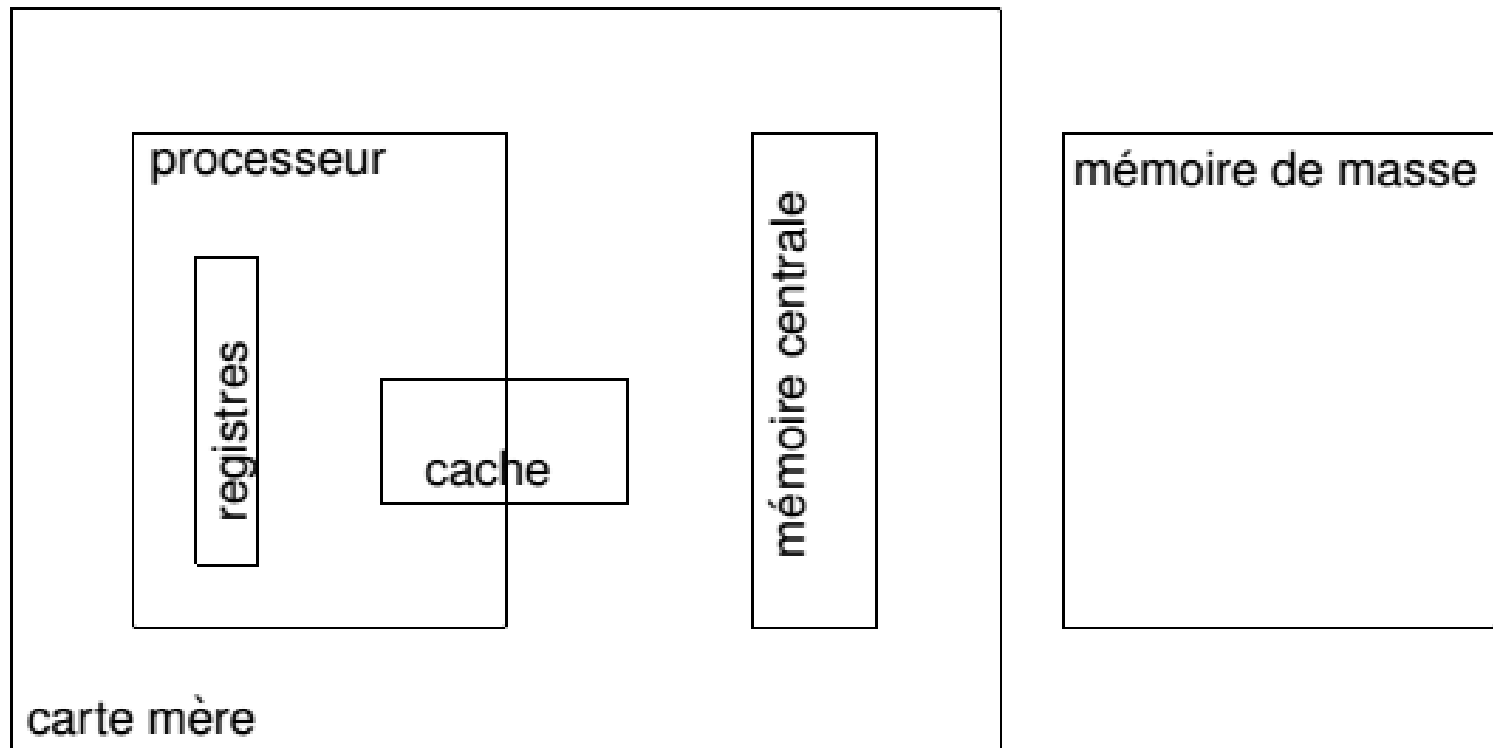
Types de mémoires



Comparaison des différents types de mémoire

	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

Localisation des différents types des mémoires

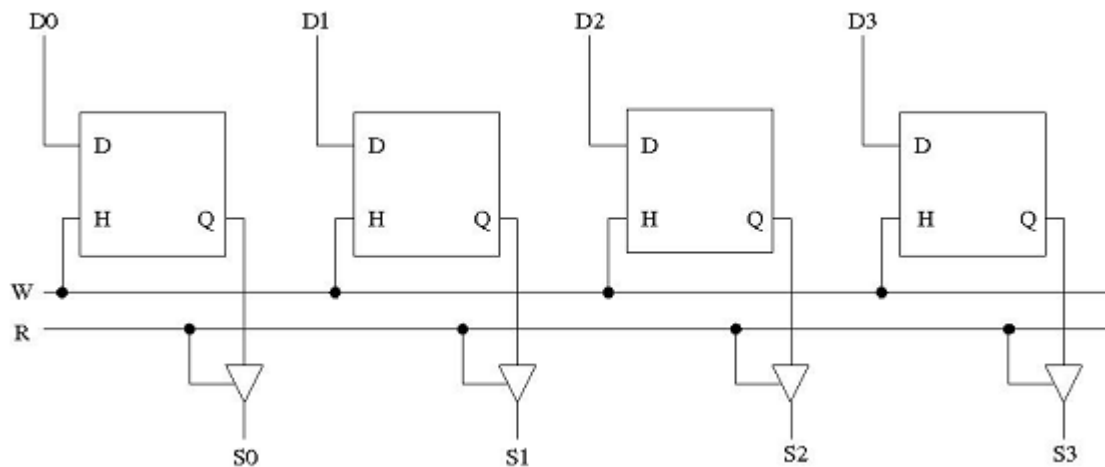


Méthode d'accès

- **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 dure.
 - 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.

Registres

- Mémoire de type SRAM.
 - L'information n'a pas besoin d'être rafraîchie.
 - Réalisation :
 - Bascule (RS ou D) qui stocke l'information. (4 transistors = 2 portes NOR) .
- Intégrés au cœur du processeur (Même dans d'autres composants).
- Un registre stocke une instruction ou les informations relatives à une instruction.
(Opérandes nécessaires à l'instruction, Résultats produits par l'instruction)
- Très peu nombreux (généralement < 40)
- Très rapides (cadencés à la vitesse du processeur).



Bancs de registres

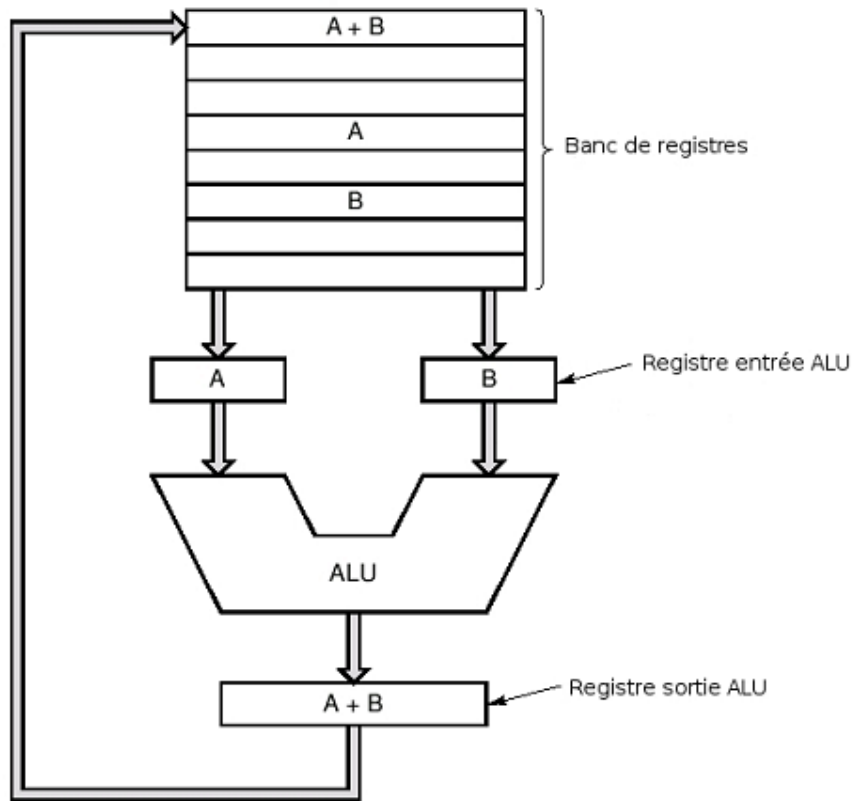
- Un **banc** de registres de hauteur **N** et de largeur **K** est un ensemble de **N** registres de **K** bits (une mémoire de N adresses).

Exemple:

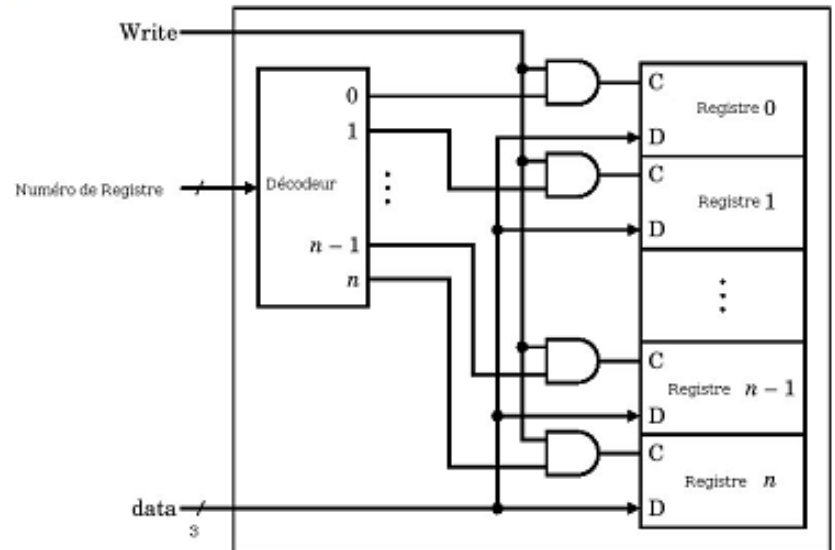
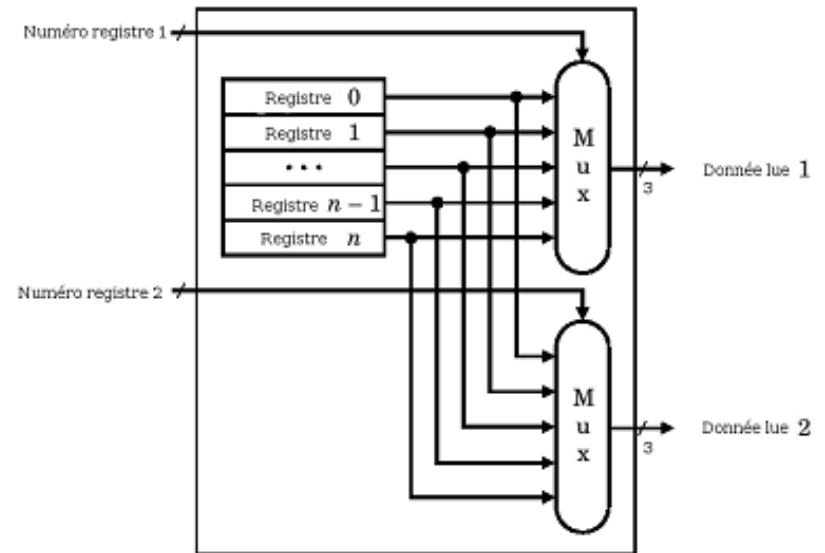
Une banc de N registres de 3 bits avec:

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

Bancs de registre (Exemple d'utilisation)



Exemple d'utilisation d'un banc de registre



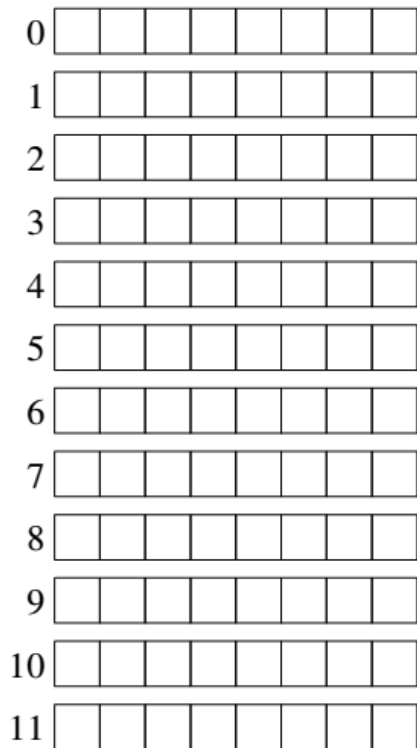
Fonctionnement en lecture Fonctionnement en écriture

Mémoire Centrale

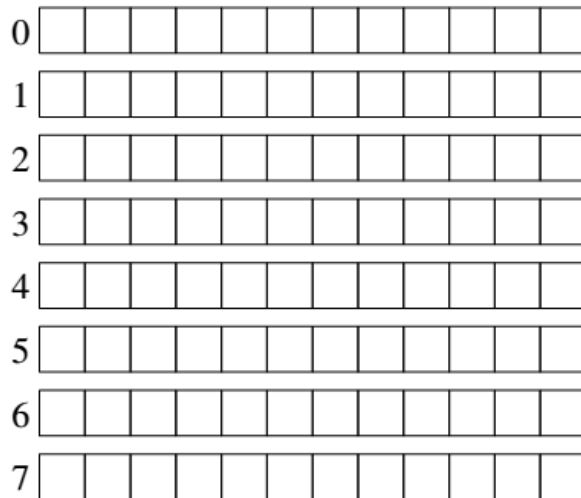
- Mémoire de type DRAM:
 - Dynamique : l'information doit être périodiquement rafraîchie.
 - 1 bit = 1 transistor + 1 condensateur.
- l'information y est stockée comme des mots (mémoire) d'un certain nombre de bits (8, 16, 32, 64 bits) : la longueur des mots est le format de la mémoire.
- vitesse relativement lente (comparée à celle du processeur).

Organisation de la mémoire

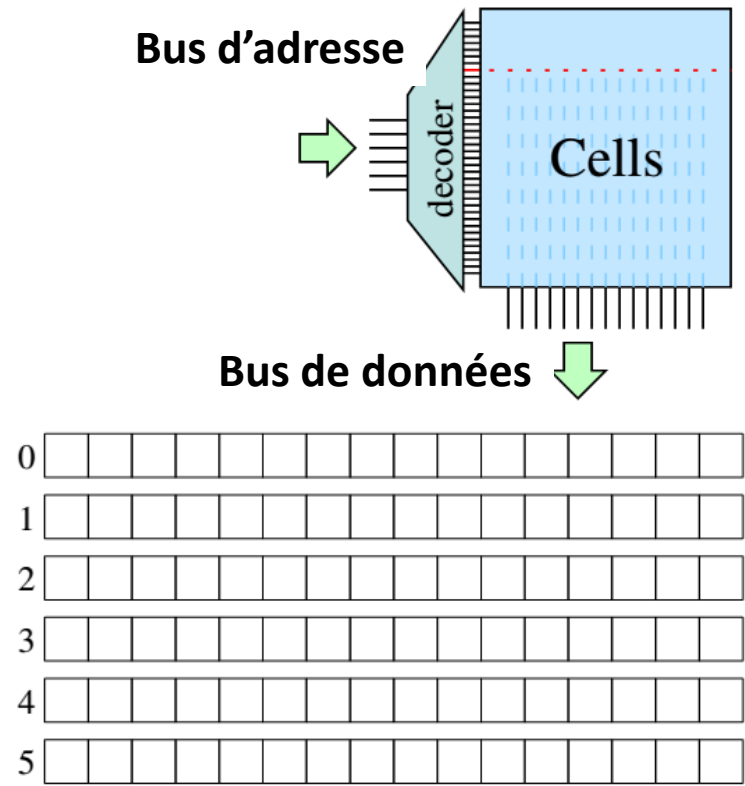
- Taille d'une cellule mémoire : K bits = nombre de fils de bus.
- Nombre de cellule mémoire : 2^N cellule \rightarrow N fils de bus d'adresse.
- Taille de mémoire: $2^N \times K$ bits.



12 cells de 8 bits

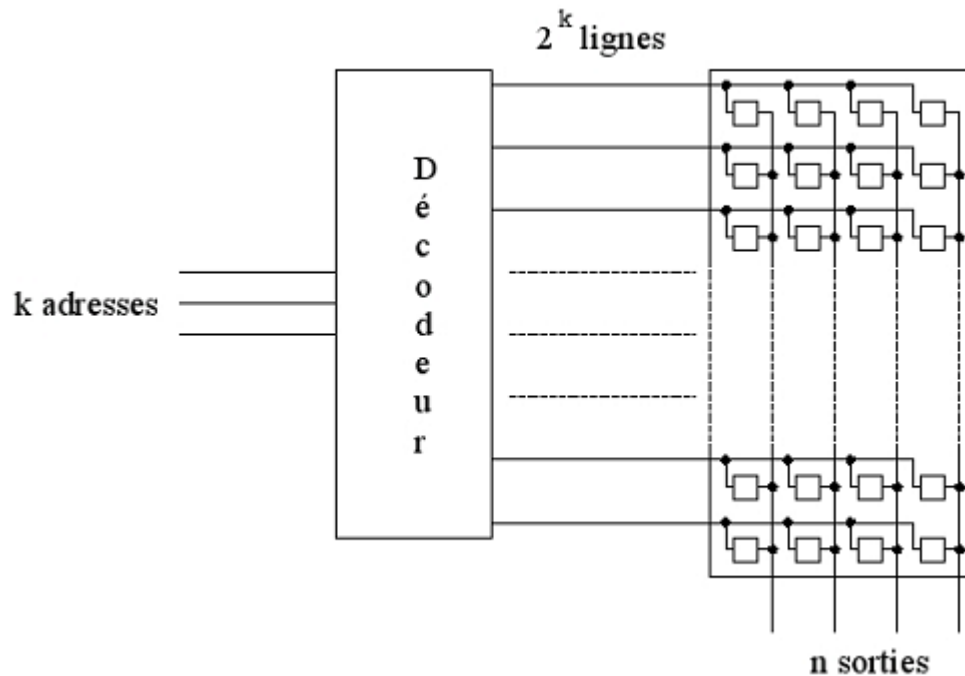


8 cells de 12 bits



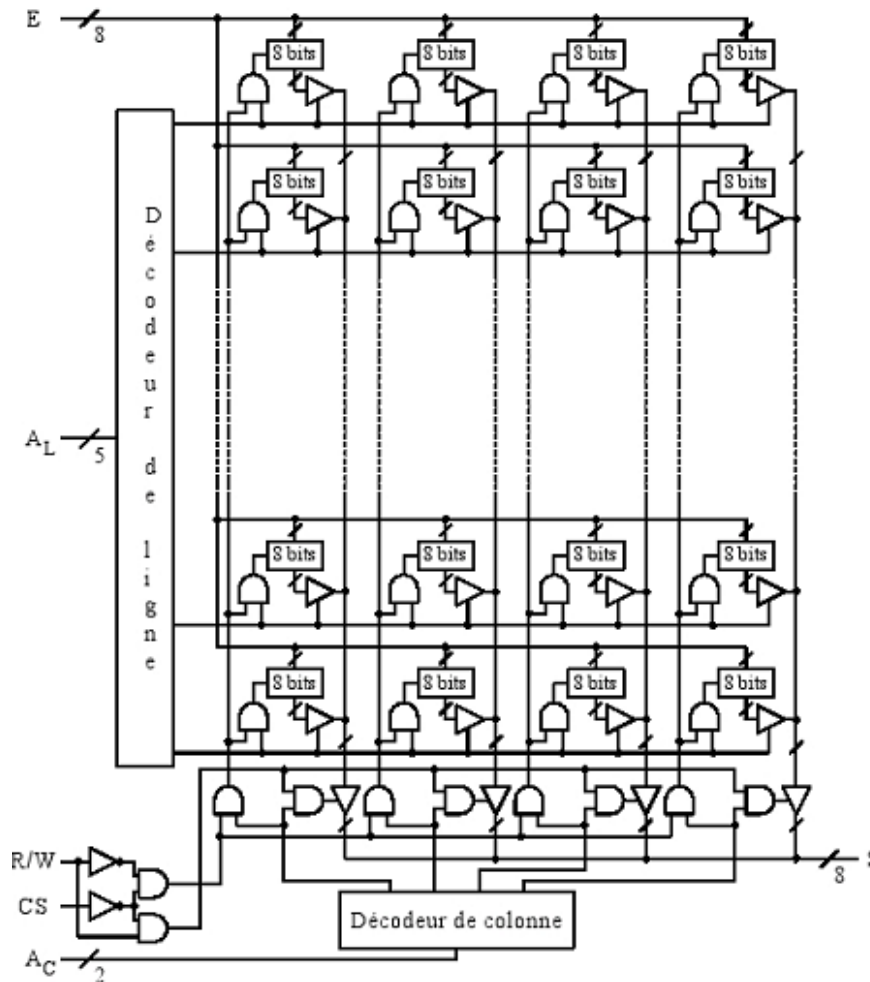
6 cells de 16 bits

Organisation de la mémoire (Mémoire Unidimensionnelle)



Nombre de portes dans le décodeur trop important

Organisation de la mémoire (Mémoire bidimensionnelle)



Avenage:

- Moins de port logique.
- Moins de broche de connexion.
- Possibilité d'accéder au colonne de la même ligne.

Inconvénient:

Environ deux fois plus de temps pour l'accès.

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

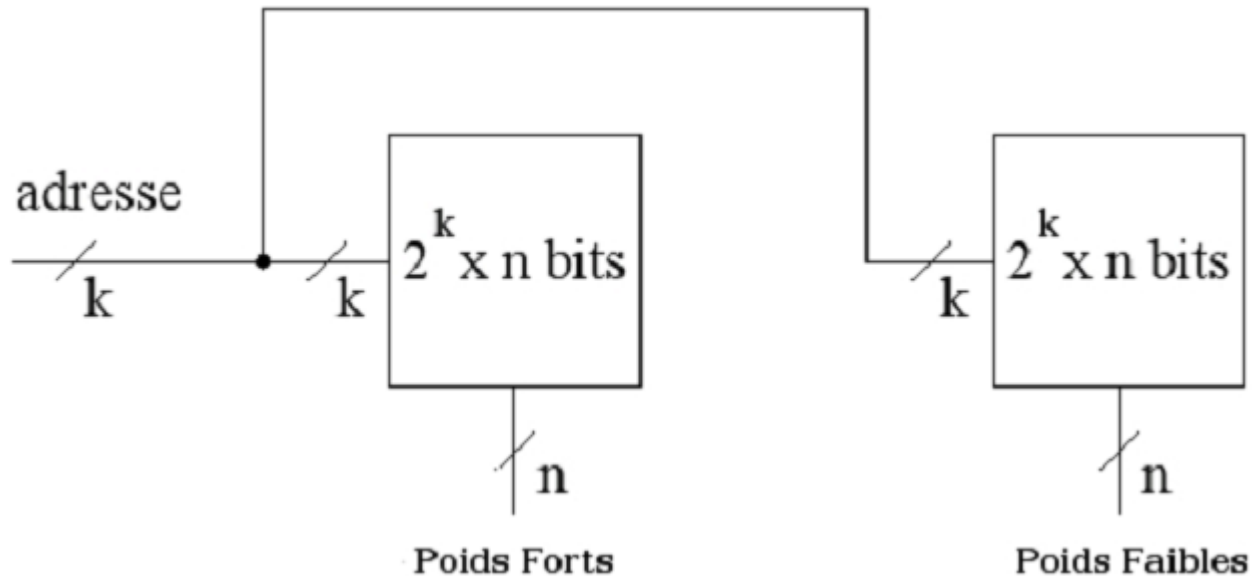
Représentation matricielle: nombre de ligne égale au nombre de colonne

Assemblage de boitiers mémoire

- Du aux limites technologiques d'intégration:
 - Les mémoires sont regroupés dans des boitiers mémoire.
 - pour obtenir des mémoires de grandes tailles, on associe plusieurs boitiers mémoires.
- Ces blocs sont assemblés :
 - pour augmenter la taille des mots de la mémoire.
 - pour augmenter le nombre de mots dans la mémoire.

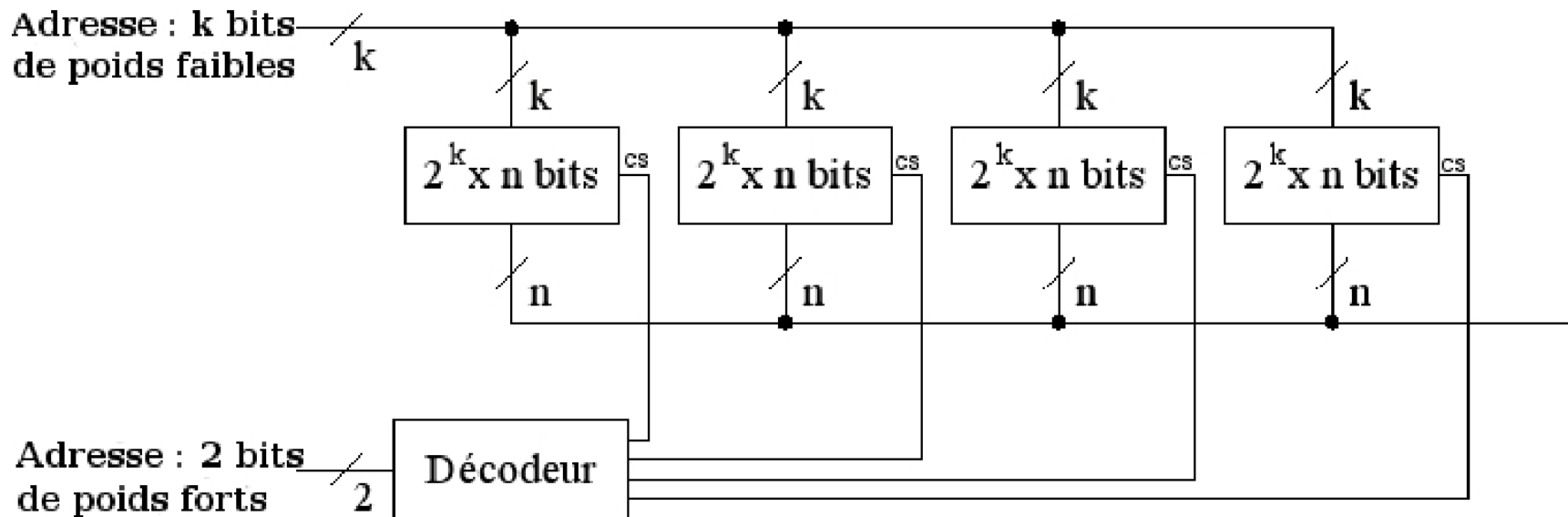
Assemblage de boitiers mémoire (Augmentation de taille d'un mot)

2 boitiers de 2^k mots de **N** bits \Rightarrow un bloc de 2^k mots de $2 \cdot N$ bits.



Assemblage de boitiers mémoire (Augmentation du nombre des mots)

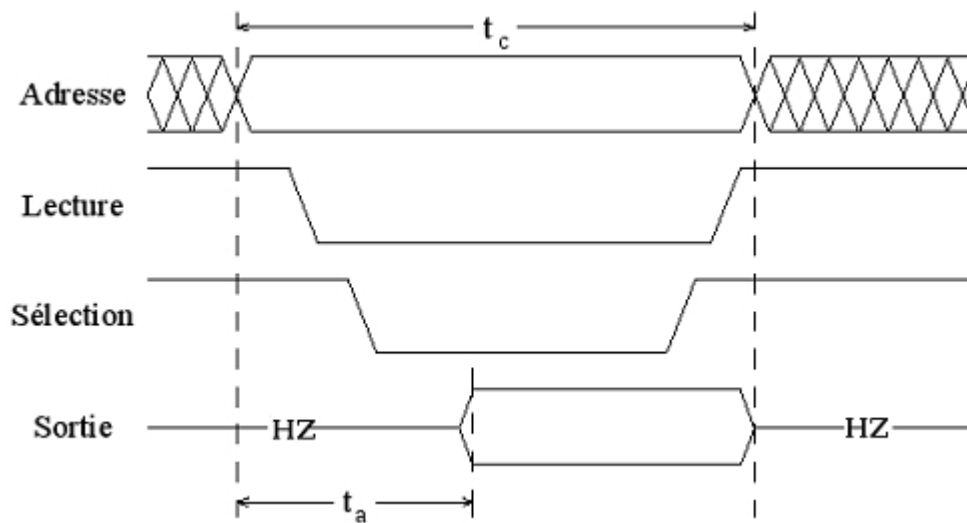
4 boitiers mémoires de 2^k mots de N bits \Rightarrow un bloc de $4 * 2^k$ mots de n bits
Adresse pour les nouveaux blocs sur $K + 2$ bits



Fonctionnement de la mémoire (Cas de lecture)

Cycle de lecture:

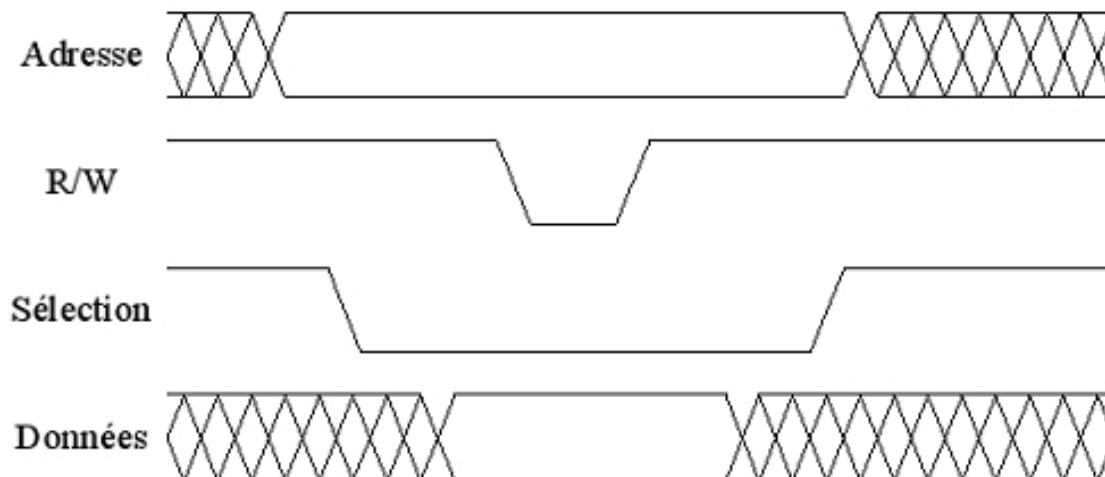
- Etablissement de l'adresse.
- Signal de lecture (R/W=0 par exemple)
- Sélection du boîtier (CS=0)
- Après un certain temps, l'information apparaît sur la sortie et reste présente jusqu'à la fin du cycle.



Fonctionnement de la mémoire (Cas d'écriture)

Cycle d'écriture:

- Etablissement de l'adresse.
- Sélection du boîtier (CS=0).
- Etablissement de la donnée sur l'entrée.
- Signal d'écriture (R/W=0 ci-dessous).



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.

$B = n / t_c$ n est le nombre de bits transférés par cycle.

Optimisation des accès

- 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**.

Mémoire et erreurs

- Les informations en mémoire peuvent comporter une ou des erreurs (Du fait de sa nature physique).
- Pour détecter et corriger, on ajoute des bits de contrôle:
 - Bit de parité: un bit supplémentaire pour calculer le nombre de 1 dans la donnée (Pair, Impair).
 - Mémoire ECC: (Error correction coding)
possède plusieurs bits supplémentaires pour détecter et corriger les erreurs.

Mémoire Logique

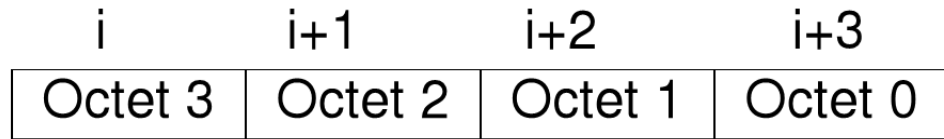
- La mémoire logique est la façon dont le processeur (ou le programmeur) voit la mémoire (physique).
- La mémoire est définie comme un ensemble de N octets consécutifs dont la première adresse est 0 la dernière adresse est $N-1$.
- Adressage de la mémoire par des mots de : 8 (octet), 16, 32, 64, .. bits.
- Un mot de 32 bits est constitué de 4 octets consécutifs.

Mémoire Logique

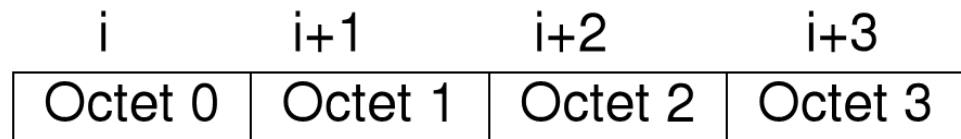
(Positionnement des données)

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**



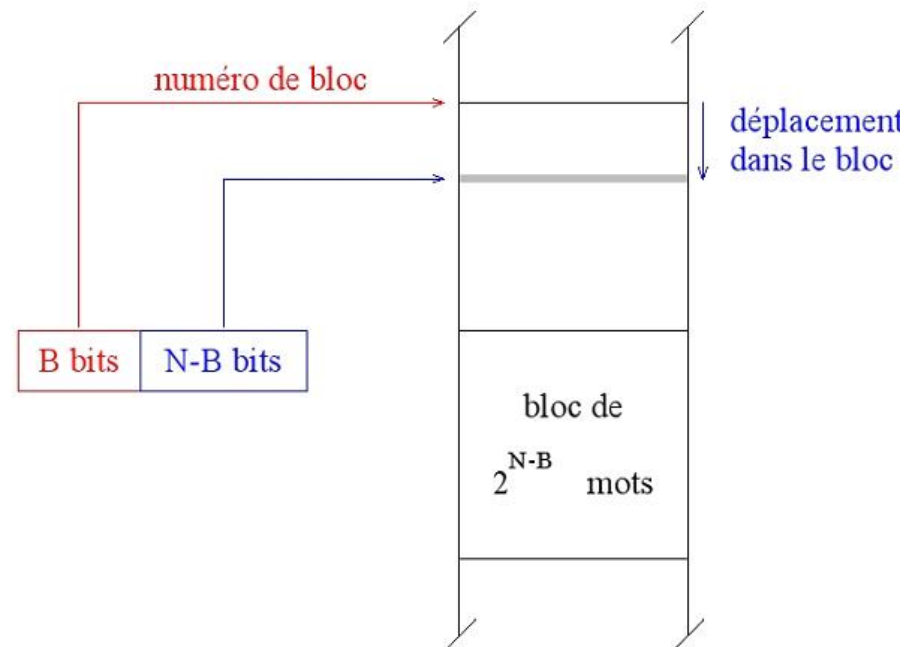
Un mot mémoire ne peut commencer n'importe où

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

Mémoire Logique (Segmentation)

- Segmentation de la mémoire : découpage logique de la mémoire en un certain nombre de blocs (ou segments).
- Une adresse est codée comme un numéro de blocs
 - un déplacement dans le bloc (**offset**).
 - **N** bits d'adresses $\Rightarrow 2^N$ cases mémoire.
 - B bits (de poids fort) pour le numéro de bloc.
 - N – B bits (de poids faible) pour le déplacement dans le bloc.

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



Ce qui reste à savoir

- Fonctionnement d'une mémoire cache.
- Fonctionnement d'une MMU.

- Pagination de la mémoire.
- Segmentation de la mémoire.