

# ARCHITECTURE DES ORDINATEURS

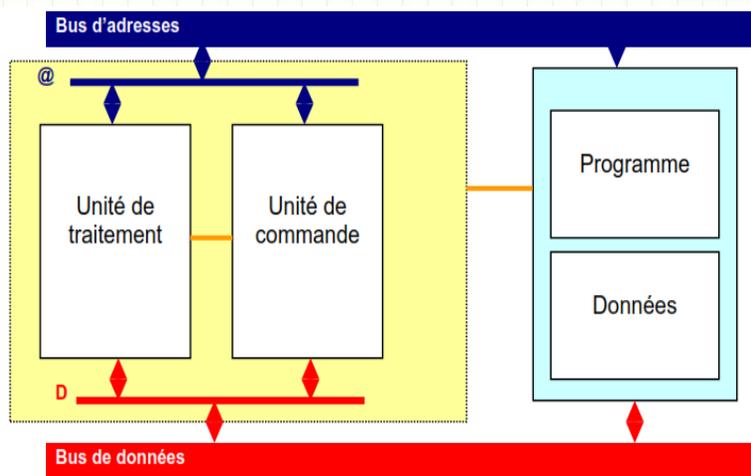
2<sup>ème</sup> Année Informatique

Chapitre2:  
Principaux composants d'un ordinateur  
Partie 1

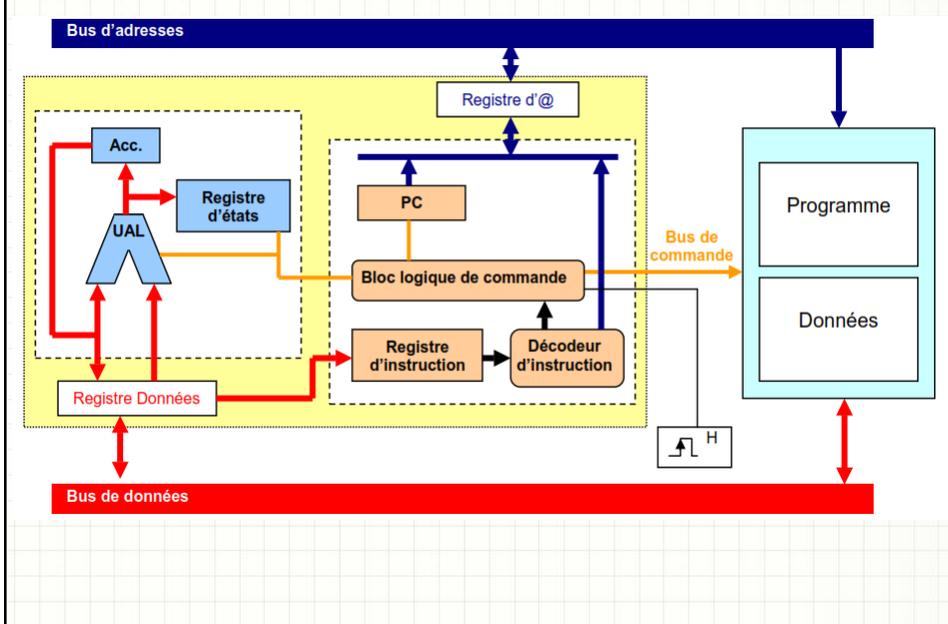
Centre universitaire Mila  
2023-2024

1

## Architecture de base d'un microprocesseur



## Schéma fonctionnel



## L'unité de commande

Elle permet de séquencer le déroulement des instructions. Elle effectue la recherche en mémoire de l'instruction. Comme chaque instruction est codée sous forme binaire, elle en assure le décodage pour enfin réaliser son exécution puis effectue la préparation de l'instruction suivante. Pour cela, elle est composée par :

- le compteur de programme
- le registre d'instruction et le décodeur d'instruction
- Bloc logique de commande (ou séquenceur)

## L'unité de commande

- **le compteur de programme (pc)** constitué par un registre dont le contenu est initialisé avec l'adresse de la première instruction du programme. Il contient toujours l'adresse de l'instruction à exécuter.
- **le registre d'instruction et le décodeur d'instruction** : chacune des instructions à exécuter est rangée dans le registre instruction puis est décodée par le décodeur d'instruction.
- **Bloc logique de commande (ou séquenceur)** : Il organise l'exécution des instructions au rythme d'une horloge. Il élabore tous les signaux de synchronisation internes ou externes (bus de commande) du microprocesseur en fonction des divers signaux de commande provenant du décodeur d'instruction ou du registre d'état par exemple.

5

## L'unité de traitement

C'est le cœur du microprocesseur. Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions :

- **L'Unité Arithmétique et Logique (UAL)**
- **Le registre d'état**
- **Les accumulateurs**

6

## L'unité de traitement

- **Le registre d'état** est généralement composé des bits à considérer individuellement. Chacun de ces bits est un indicateur dont l'état dépend du résultat de la dernière opération effectuée par l'UAL. On les appelle *indicateur d'état* ou *flag* ou *drapeaux*. Dans un programme le résultat du test de leur état conditionne souvent le déroulement de la suite du programme. On peut citer par exemple les indicateurs de :
  - retenue (**carry : C**)
  - retenue intermédiaire (**Auxiliary-Carry : AC**)
  - signe (**Sign : S**)
  - débordement (**overflow : OV ou V**)
  - parité (**Parity : P**)
- **Les accumulateurs** sont des registres de travail qui servent à stocker un opérande au début d'une opération arithmétique et le résultat à la fin de l'opération.

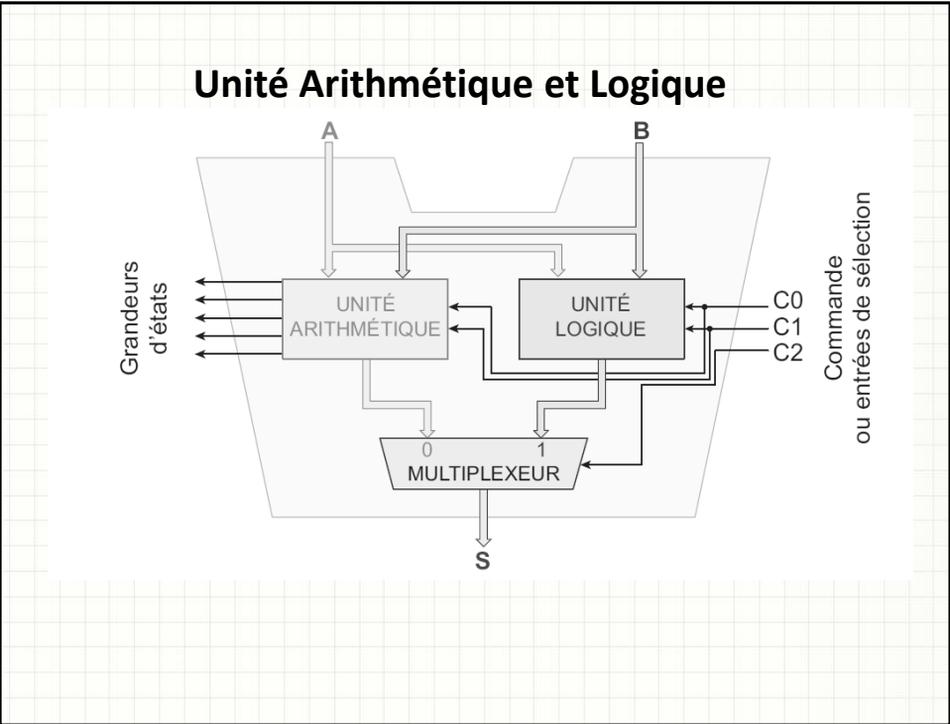
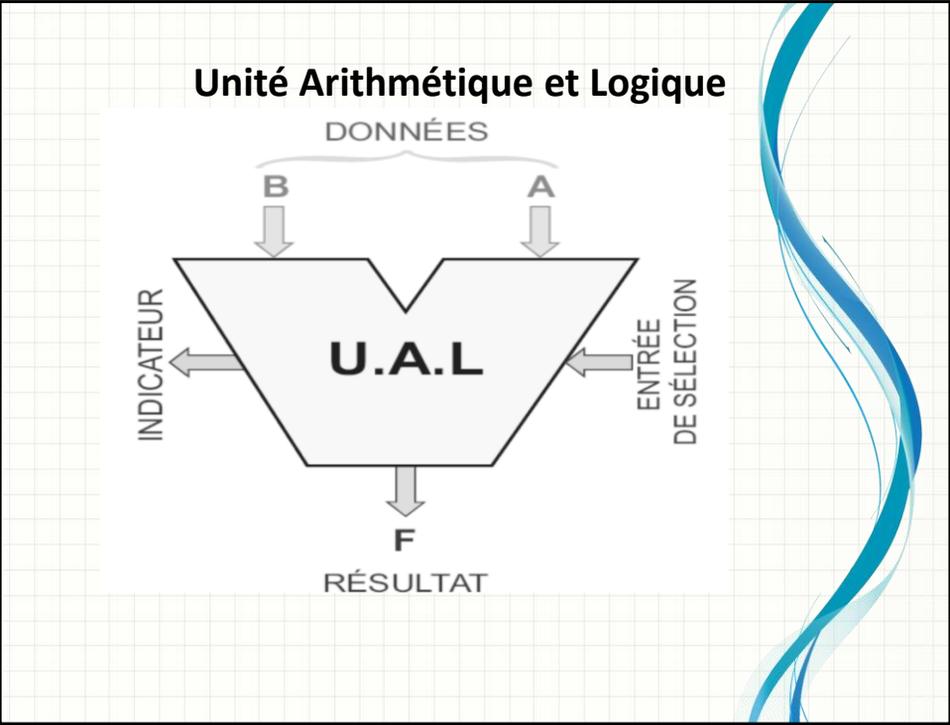
7

## Unité Arithmétique et Logique

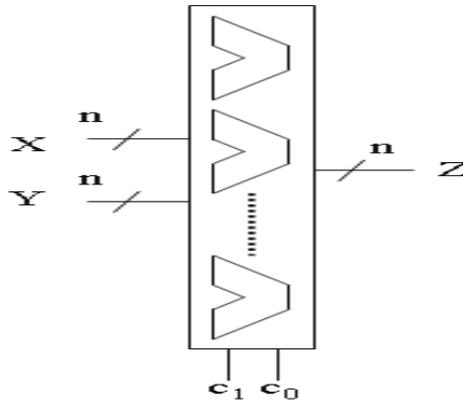
L'**UAL** (Unité d'Arithmétique et Logique) est l'élément de base d'un microprocesseur, son rôle est la réalisation d'opérations :

- arithmétiques (additions, soustractions...)
- et logiques (OR, AND, NOR...),
- de décalage,
- de comparaisons
- et de transfert.

Le nombre d'opérations élémentaires arithmétiques et logiques implémentées dans l'UAL d'un processeur et la façon dont elles y sont implémentées varient considérablement d'un processeur à l'autre,



## UAL $n$ -bits



**UAL  $n$ -bits.** Tout comme l'additionneur  $n$ -bits, l'UAL  $n$ -bits est obtenu en chaînant  $n$  **UAL 1-bit**s.

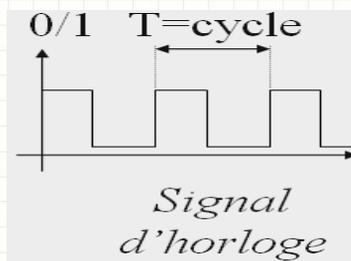
## Notions d' horloge

Pour pouvoir chaîner entre eux plusieurs composants, il est donc nécessaire d'estimer le temps de passage à travers chaque composant, et de bloquer la propagation du résultat vers les composants suivants tant que le calcul n'est pas terminé.

Pour cela, on utilise des « barrières », des composants tampons placés en entrée et en sortie des composants de calcul, destinées à contrôler la propagation des résultats d'un composant à l'autre. Afin de simplifier la commande de ces barrières, dans la plupart des processeurs, elles s'ouvrent et se ferment à intervalle fixe. Elles sont donc commandées par un signal périodique comportant une phase haute (ouverture) et une phase basse (fermeture), appelé **une horloge**.

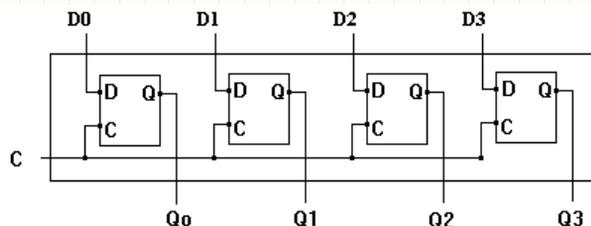
## Notions d' horloge

Afin de simplifier la synchronisation des différents composants d'un processeur, il n'existe en général qu'une seule horloge à l'intérieur d'un processeur. La durée de la période, également appelée **un cycle**, doit être égale au plus long temps de calcul du plus lent des composants du processeur ; ce composant est en général appelé le facteur limitatif puisqu'il détermine la cadence d'exécution du processeur.



## Les registres

- Un registre est un ensemble de mémoires élémentaires (bascules), synchronisées par le même signal d'horloge.
- Les registres sont largement utilisés dans les systèmes de traitement numérique pour réaliser des opérations : de mémorisation provisoire (mémoire tampon), de décalage, de rotation, ...
- Une bascule est l'élément de base de la logique séquentielle.
- Une bascule permet de mémoriser un seul bit.
- Exemple :

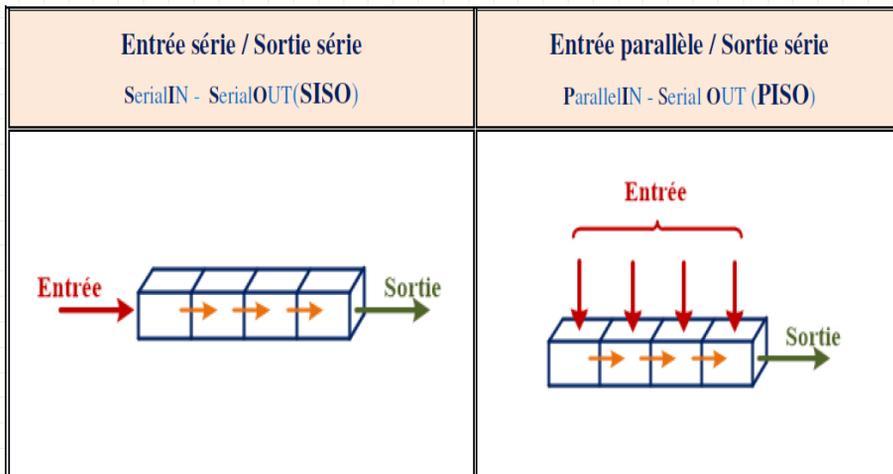


## Caractéristiques d'un registre

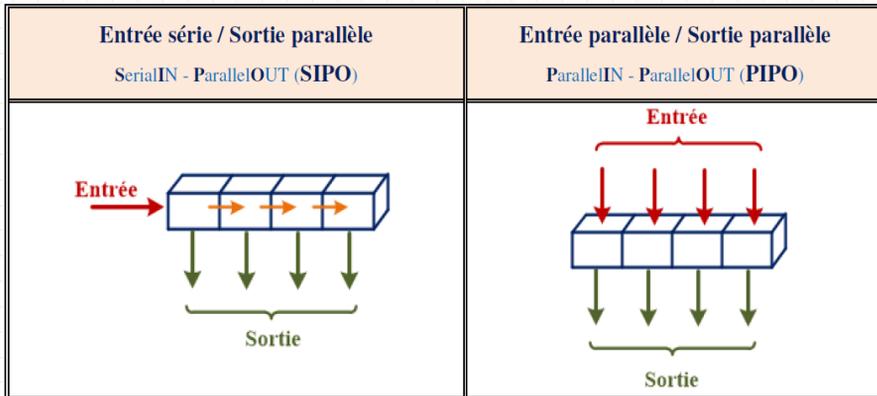
- **La capacité:** nombre de bits du mot binaire qu'il peut mémoriser.
- **Le mode d'écriture ou de chargement:** dépend du nombre d'entrées :
  - ❖ écriture série : génération bit par bit, avec transmission par un seul fil conducteur.
  - ❖ écriture parallèle : génération globale du mot de  $n$  bits, avec transmission par un bus de  $n$  bits.
- **Le mode de lecture:**
  - ❖ lecture série : exploitation bit par bit du mot (une seule sortie).
  - ❖ lecture parallèle : exploitation globale du mot ( $n$  sorties).

15

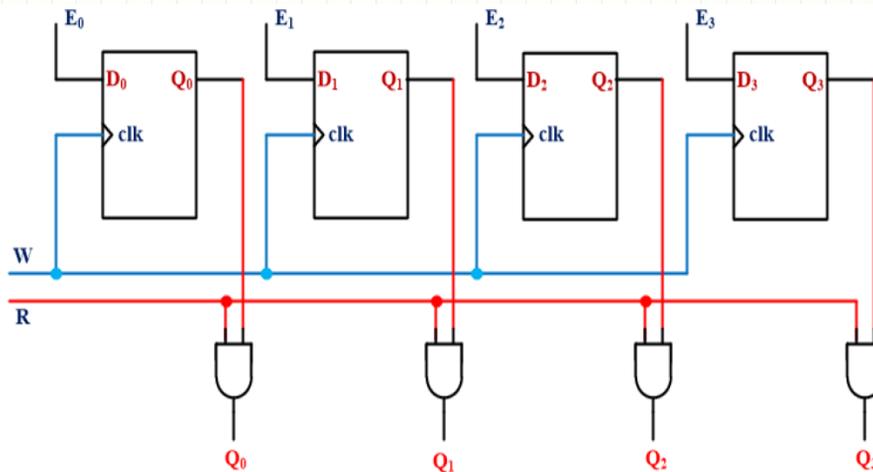
## Types des registres



## Types des registres

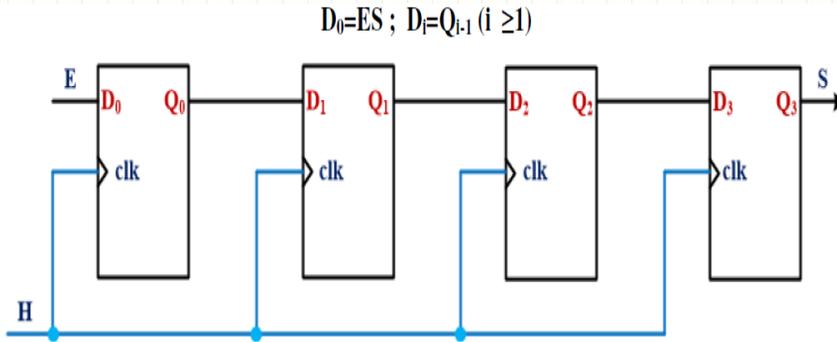


## Registres de mémorisation



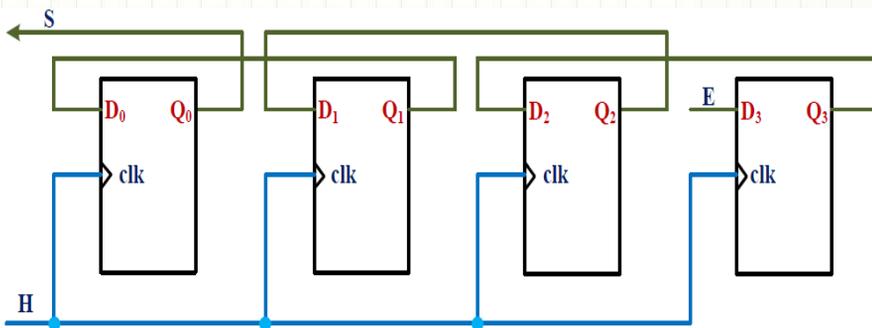
## Registres à décalage

Registres à décalage *entrée série - sortie série* (Décalage à droite)



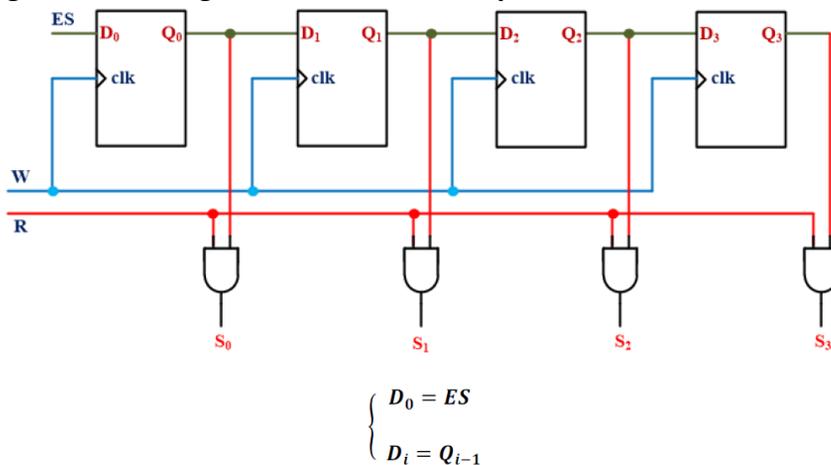
## Registres à décalage

Registres à décalage *entrée série - sortie série* (Décalage à gauche)



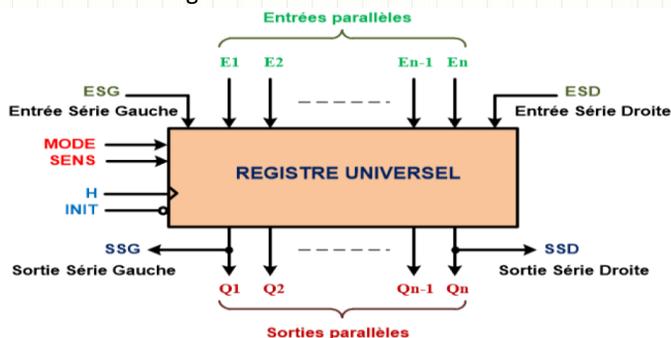
## Registres à décalage

### Registre à décalage entrée série - sortie parallèle



## Registre universel

C'est un registre qui effectue un chargement des données série ou parallèle et un décalage à gauche ou droite avec une lecture série ou parallèle. Il dispose d'entrées de mode de fonctionnement qui définissent le type de chargement et le sens de décalage. La figure ci-dessous représente la configuration d'un tel registre.



- L'entrée MODE permet de choisir le mode de chargement série ou parallèle.
- L'entrée SENS permet de choisir le sens de décalage à gauche ou à droite.
- L'entrée INIT permet d'initialiser le registre.

## Applications des registres

- Mémorisation temporaire d'une information
- Conversion parallèle-série de mots binaires
- Conversion série-parallèle d'un train binaire
- Ligne à retard numérique
- Division et multiplication par  $2^n$

23

## Les mémoires

## 1. Introduction

- Avec une bascule c'est possible de mémoriser une information sur **1 seul bit**.
- Avec un registre c'est possible de mémoriser une information sur **n bits**.
- Si on veut mémoriser une information de **taille important** → il faut utiliser **une mémoire** .

25

## 2. C'est quoi une mémoire ?

- Une mémoire est un **dispositif** capable :
  - D'enregistrer une information,
  - De la conserver ( mémoriser )
  - et de la restituer ( possible de la lire ou la récupérer par la suite).
- Exemple de mémoire :
  - La mémoire centrale
  - Un disque dur
  - Une disquette
  - Un flash disque
  - .....
- La mémoire peut être **dans** le processeur ( des registres ) , **interne** (Mémoire centrale ou principale) ou **externe** (Mémoire secondaire) .

26

### 3. Caractéristiques des mémoires

#### 1. La capacité d'une mémoire

- La **capacité** ( taille ) d'une mémoire est **le nombre (quantité)** d'informations qu'on peut enregistrer ( mémoriser ) dans cette mémoire.
- La capacité peut s'exprimer en :
  - **Bit** : un bit est l'élément de base pour la représentation de l'information .
  - **Octet** : 1 Octet = 8 bits
  - **kilo-octet (KO)** : 1 kilo-octet (KO) = 1024 octets =  $2^{10}$  octets
  - **Méga-octet (MO)** : 1 Méga-octet (MO) = 1024 KO =  $2^{20}$  octets
  - **Géga-octet (GO)** : Géga-octet (GO) = 1024 MO =  $2^{30}$  octets
  - **Téra-octet (To)** : 1 téra-octet (To) = 1024 Go =  $2^{40}$  octets

27

### 3. Caractéristiques des mémoires

#### 2. Volatilité

- Si une mémoire **perd son contenu** ( les informations ) lorsque la source d'alimentation **est coupée** alors la mémoire est dite **volatile**.
- Si une mémoire **ne perd pas ( conserve )** son contenu lorsque la source d'alimentation **est coupée** alors la mémoire est dite **non volatile ( mémoire permanente ou stable)**.

28

### 3. Caractéristiques des mémoires

#### 3. Mode d'accès à l'information ( lecture /écriture )

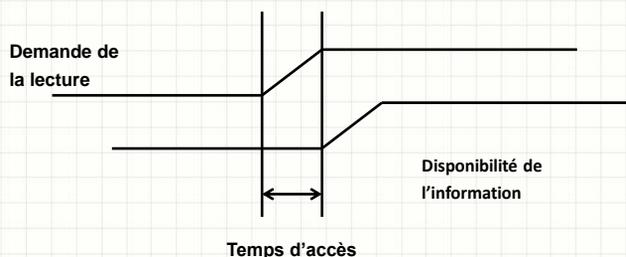
- Sur une mémoire on peut effectuer l'opération de :
  - **lecture** : récupérer / restituer une information à partir de la mémoire.
  - **écriture** : enregistrer une nouvelle information ou modifier une information déjà existante dans la mémoire .
- Il existe des mémoires qui offrent **les deux** modes lecteur/écriture , ces mémoire s'appelles **mémoires vives**.
- Il existent des mémoires qui offrent **uniquement** la possibilité de la **lecture** ( c'est pas possible de modifier le contenu ). Ces mémoires s'appelles **mémoires mortes**.

29

### 3. Caractéristiques des mémoires

#### 4. Temps d'accès

- C'est le temps **nécessaire pour effectuer** une opération de lecture ou d'écriture.
- Par exemple pour l'opération de lecture , le temps d'accès est le **temps** qui sépare **la demande** de la **lecture** de la **disponibilité** de l'information.



Le temps d'accès est un **critère important** pour déterminer **les performances** d'une mémoire ainsi que les performances d'une machine.

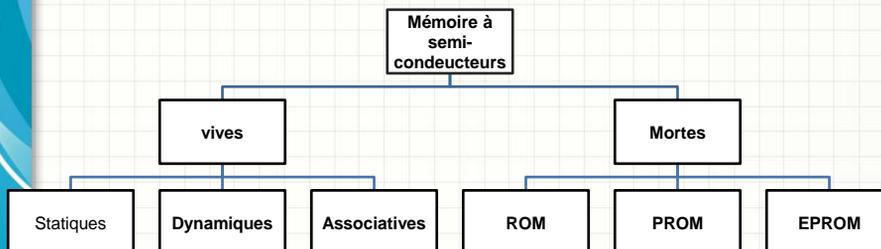
30

## 4. Classification des mémoires

- Les mémoires peuvent être classées en trois catégories selon la technologie utilisée :
  - Mémoire à semi-conducteur ( mémoire centrale, ROM, PROM,.....) : très rapide mais de taille réduite.
  - Mémoire magnétique ( disque dur, disquette,...) : moins rapide mais stocke un volume d'informations très grand.
  - Mémoire optique ( DVD, CDROM,..)

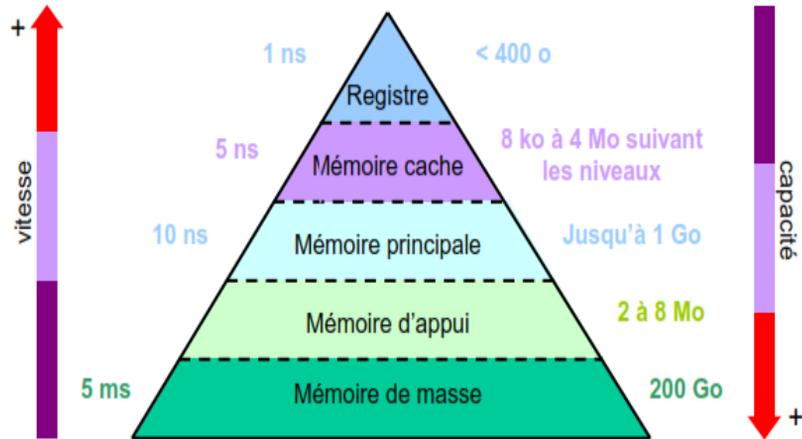
31

## 5. Mémoire à semi-conducteur



32

## 6. - Hiérarchie des mémoires



33

## LA MÉMOIRE CENTRALE

- **RAM : Random Acces memory**  
**Mémoire à accès aléatoire**

34

## Mémoire centrale

- La mémoire centrale (MC) représente l'espace de travail de l'ordinateur (calculateur).
- C'est l'organe principal de rangement des informations utilisées par le processeur.
- Dans une machine (ordinateur / calculateur) pour exécuter un programme il faut le charger (copier) dans la mémoire centrale.
- Le temps d'accès à la mémoire centrale et sa capacité sont deux éléments qui influent sur le temps d'exécution d'un programme (performance d'une machine).

35

## Caractéristiques de la mémoire centrale

- La mémoire centrale est réalisée à base de semi-conducteurs.
- La mémoire centrale est une mémoire vive : accès en lecture et écriture.
- La mémoire centrale est dite à accès aléatoire (RAM : Random Acces Memory) c'est-à-dire que le temps d'accès à l'information est indépendant de sa place en mémoire.
- La mémoire centrale est volatile : la conservation de son contenu nécessite la permanence de son alimentation électrique.
- Un temps d'accès à une mémoire centrale est moyen mais plus rapide que les mémoires magnétiques.
- La capacité d'une mémoire centrale est limitée mais il y a toujours une possibilité d'une extension.
- Pour la communication avec les autres organes de l'ordinateur, la mémoire centrale utilise les bus (bus d'adresses et bus de données)

36

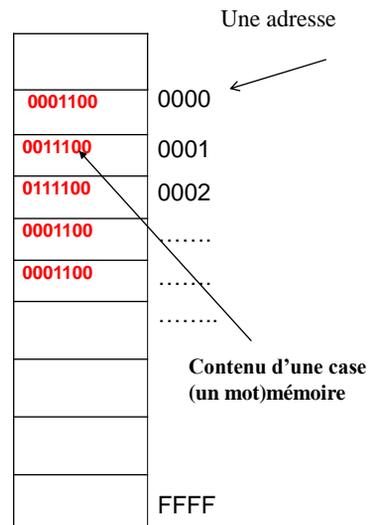
## Types des mémoires centrales

- Il existent deux grandes familles des mémoires centrales : les mémoires statiques (SRAM) et les mémoires dynamiques (DRAM).
  - Les **mémoires statiques** sont à base de bascules de type D , elles possèdent un faible taux d'intégration mais un temps d'accès rapide ( Utilisation pour les mémoires cache ).
  - Les **mémoires dynamiques** à base de condensateurs , ces mémoires possèdent un très grand taux d'intégration, elle sont plus simples que les mémoires statiques mais avec un temps d'accès plus long .

37

## Vue logique de la mémoire centrale

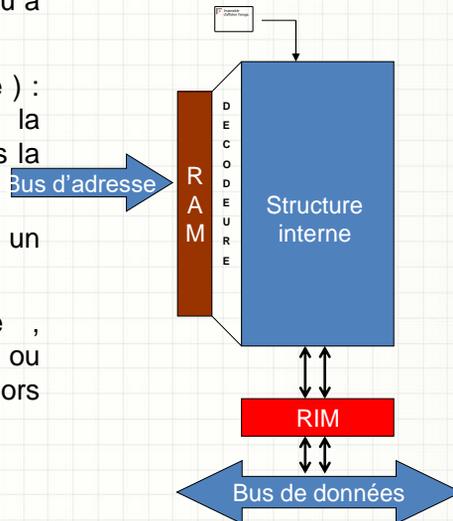
- La mémoire centrale peut être vu comme un large **vecteur ( tableau )** de **mots** ou **octets**.
- Un mot mémoire stocke une information sur **n** bits.
- un mot mémoire contient plusieurs **cellules** mémoire.
- Une cellule mémoire stock **1 seul** bit .
- Chaque mot possède sa propre **adresse**.
- Une adresse est un numéro unique qui permet d'accéder à un mot mémoire.
- Les adresses sont séquentielles ( consécutives )
- La taille de l'adresse ( le nombre de bits ) dépend de la capacité de la mémoire.



38

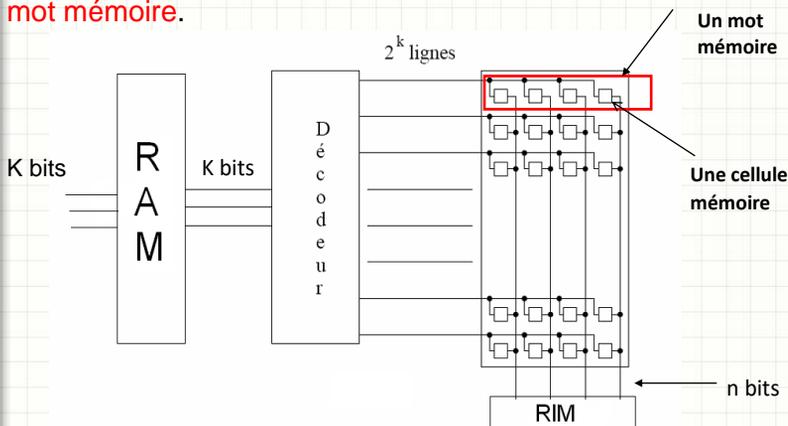
## Structure physique d'une mémoire centrale

- **RAM** (Registre d'adresse Mémoire) : ce registre stock l'adresse du mot à lire ou à écrire .
- **RIM** ( Registre d'information mémoire ) : stock l'information lu à partir de la mémoire ou l'information à écrire dans la mémoire.
- **Décodeur** : permet de sélectionner un mot mémoire.
- **R/W** : commande de lecture/écriture , cette commande permet de lire ou d'écrire dans la mémoire ( si  $R/\bar{W}=1$  alors lecture sinon écriture )
- Bus d'adresses de taille **k bits**
- Bus de données de taille **n bits**



## Sélection d'un mot mémoire

- Lorsque une adresse est chargée dans le registre **RAM** , le décodeur va recevoir la même information que celle du RAM.
- A la sortie du décodeur nous allons avoir **une seule sortie** qui est active → Cette sortie va nous permettre de sélectionner **un seul mot mémoire**.



## Comment calculer la capacité d'une MC ?

- Soit  $k$  la taille du bus d'adresses ( taille du registre RAM)
- Soit  $n$  la taille du bus de données ( taille du registre RIM ou la taille d'un mot mémoire )
- On peut exprimer la capacité de la mémoire centrale soit en nombre de **mots mémoire** ou en **bits** ( octets, kilo-octets,...)
  - La capacité =  $2^k$  Mots mémoire
  - La capacité =  $2^k * n$  Bits

### Exemple :

Dans une mémoire la taille du bus d'adresses  $K=14$  et la taille du bus de données  $n=4$ . Calculer la capacité de cette mémoire ?

$$C=2^{14} = 16384 \text{ Mots de 4 bits}$$

$$C= 2^{14} * 4 = 65536 \text{ Bits} = 8192 \text{ Octets} = 8 \text{ Ko}$$

41

## La lecture d'une information

- Pour lire une information en mémoire centrale il faut effectuer les opérations suivantes:
  - Charger dans le registre RAM l'adresse du mot à lire.
  - Lancer la commande de lecture ( R/W=1)
  - L'information est disponible dans le registre RIM au bout d'un certain temps ( temps d'accès)

42

## L'écriture d'une information

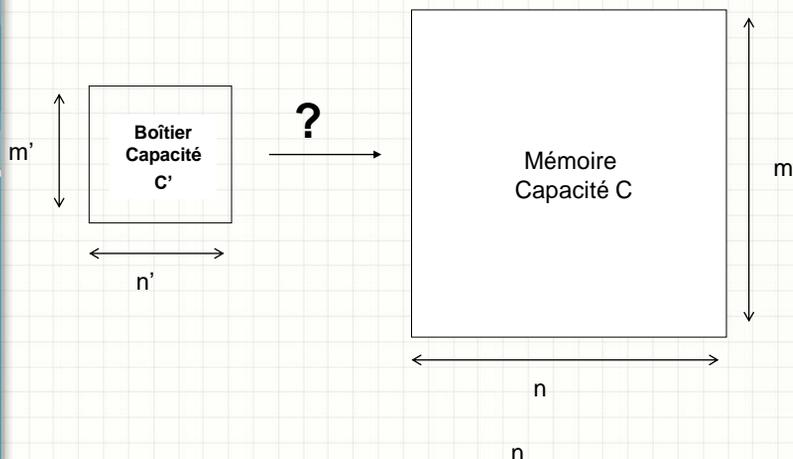
- Pour écrire une information en MC il faut effectuer les opérations suivantes:
  - Charger dans le RAM l'adresse du mot ou se fera l'écriture.
  - Placer dans le RIM l'information à écrire.
  - Lancer la commande d'écriture pour transférer le contenu du RIM dans la mémoire .

43

## Conception des MC

### Problème ?

On veut réaliser une mémoire de **capacité C** , mais nous disposons uniquement de boîtiers ( des circuits ) de **taille inférieur** ?



44

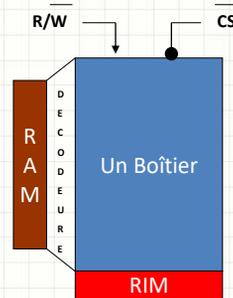
## Structure d'un boîtier

Un boîtier possède la même structure qu'une mémoire ( RAM,RIM,...) en plus de la commande CS.

$\overline{\text{CS}}$  (Chip Select) : c'est une commande en logique négative qui permet de sélectionner ( activer ) un boîtier .

$\overline{\text{CS}}=0$  le boîtier est sélectionné

$\overline{\text{CS}}=1$  le boîtier n'est pas sélectionné



45

## Solution

- Soit M une mémoire de capacité C , tel que **m** est le nombre de mot et **n** la taille **d'un mot**.
- Soit M' un boîtier de capacité C' , tel que m' le nombre de mot et n' la taille d'un mot.
- On suppose que  $C > C'$  (  $m \geq m'$  ,  $n \geq n'$  )
- Quel est le **nombre de boîtiers** M' nécessaire pour réaliser la mémoire M ?
- Pour connaître le nombre de boîtiers nécessaire , il faut calculer les deux facteurs suivants :
  - $P = m/m'$
  - $Q = n/n'$

46

## Solution (suite )

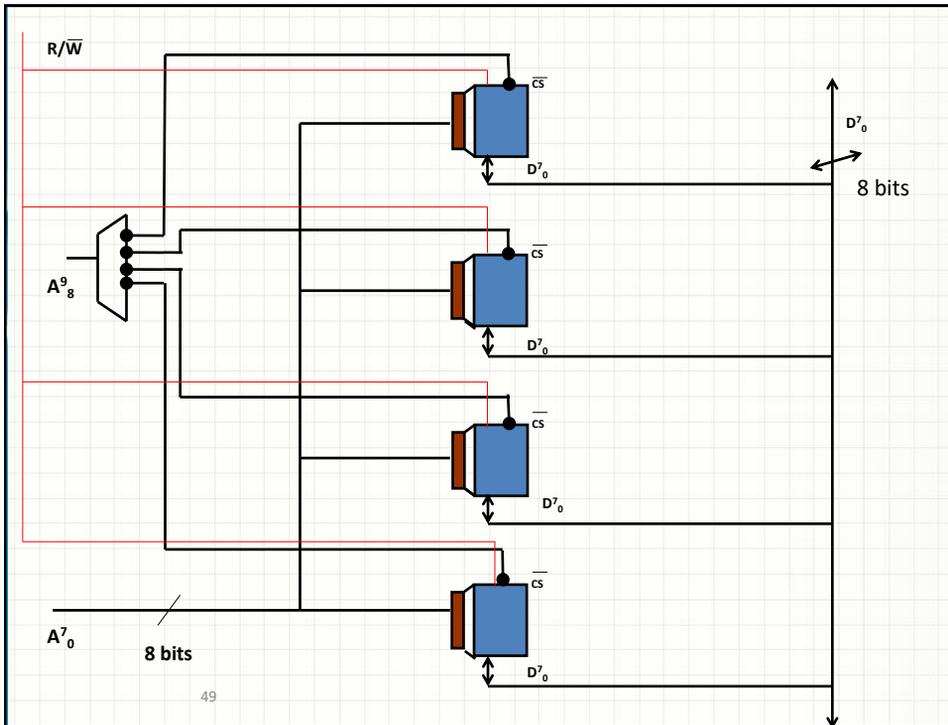
- P : permet de déterminer le nombre de boîtiers M' nécessaire pour obtenir le **nombre de mots** de la mémoire M ( extension lignes ).
- Q : permet de déterminer le nombre de boîtier M' nécessaire pour obtenir la **taille de mot** de la mémoire M ( extension mots ou extension colonnes).
- **P.Q** donne le nombre totale de boîtiers nécessaire pour réaliser la mémoire M.
- Pour sélectionner les boîtiers on utilise **les bits de poids forts** d'adresses. Si P est le facteur d'extension lignes alors on prend k bits tel que  $P=2^k$ .
- Les autres bits d'adresses restants sont utilisés pour sélectionner un mot dans un boîtier.

47

## Exemple

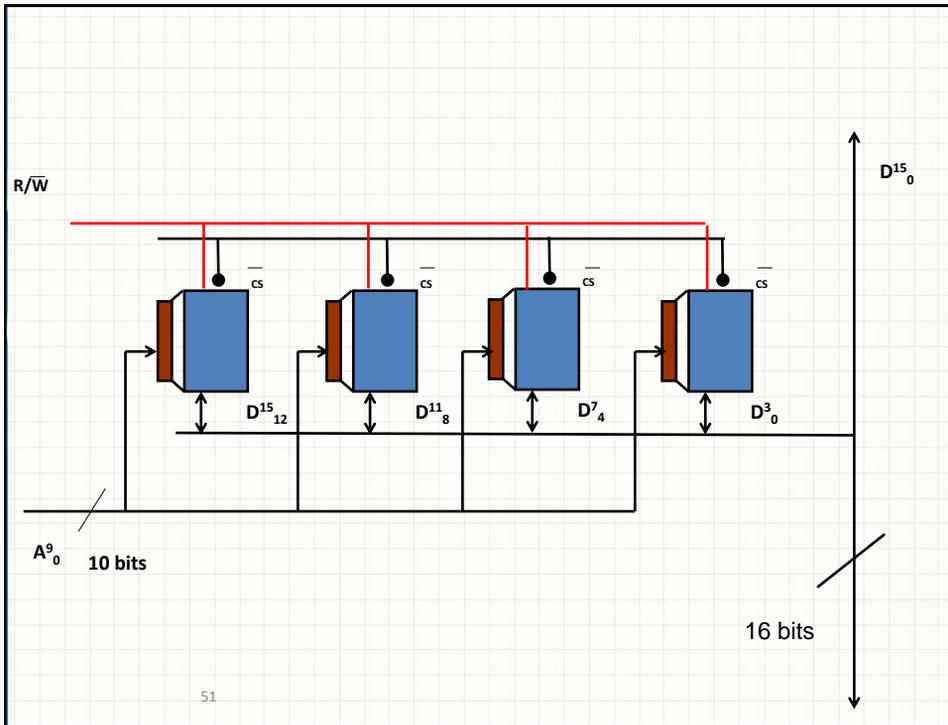
- Réaliser une mémoire de 1Ko ( la taille d'un mot est de 8 bits) en utilisant des boîtiers de taille 256 mots de 8 bits ?
- **Solution :**  
(m,n)=(1024,8) → taille du bus d'adresses est de 10 bits  $A^9_0(A9...A0)$ , taille du bus de données est de 8 bits  $D^7_0(D7....D0)$   
(m',n')=(256,8) → taille du bus d'adresses est de 8 bits ( $A7'...A0'$ ), taille du bus de données est de 8 bits ( $D7'....D0'$ )
- Calculer les deux facteurs d'extension lignes et colonnes :  
 $P= m/m' =1024/256=4$  ( extension lignes )  
 $Q= n/n' =8/8=1$  (extension colonnes)
- Le nombre totale de boîtiers  $P.Q=4$

48



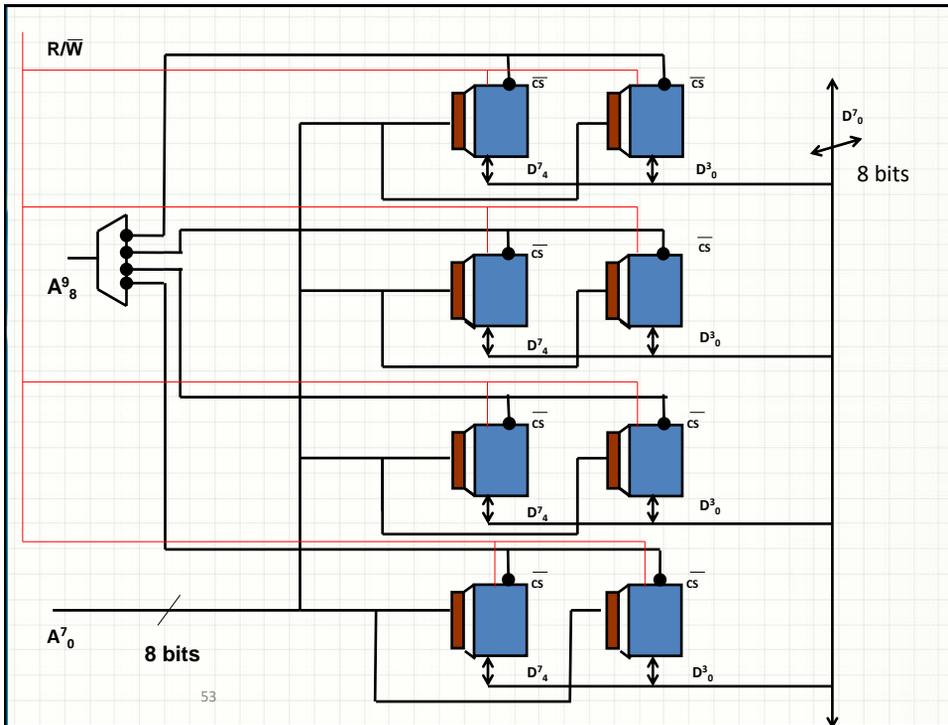
## Exemple 2

- On veut réaliser une mémoire de 1024 mot( la taille d'un mot est de 16 bits) en utilisant des boîtiers de taille 1024 mots de 4 bits ) ?
- Solution :  
 $(m,n)=(1024,16) \rightarrow$  taille du bus d'adresses est de 10 bits ( $A_9 \dots A_0$ ),  
 taille de bus de données est du 16 bits ( $D_{15} \dots D_0$ )  
 $(m',n')=(1024,4) \rightarrow$  taille du bus d'adresses est de 10 bits ( $A_9' \dots A_0'$ ),  
 taille de bus de données est du 4 bits ( $D_3' \dots D_0'$ )
- $P=1024/1024=1$  ( extension lignes )
- $Q=16/4=4$  (extension colonnes)
- Le nombre totale de boîtiers  $P.Q=4$



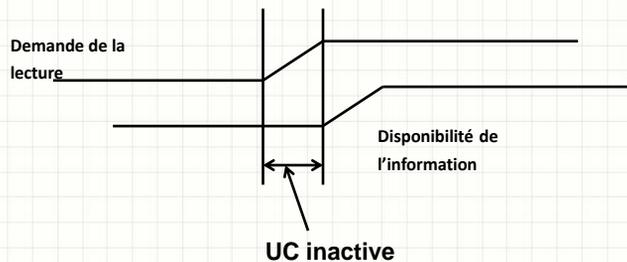
### Exemple 3

- On veut réaliser une mémoire de 1KO ( la taille d'un mot est de 8 bits) en utilisant des boîtiers de taille 256 mots de 4 bits ) ?
- Solution :  
 $(m,n)=(1024,8) \rightarrow$  taille du bus d'adresses est de 10 bits (A<sub>9</sub>...A<sub>0</sub>), taille du bus de données est de 8 bits (D<sub>7</sub>...D<sub>0</sub>)  
 $(m',n')=(256,4) \rightarrow$  taille du bus d'adresses est de 8 bits (A<sub>7</sub>...A<sub>0</sub>), taille du bus de données est de 4 bits (D<sub>3</sub>...D<sub>0</sub>)
- $P=1024/256=4$  ( extension lignes )
- $Q=8/4=2$  ( extension colonnes)
- Le nombre totale de boîtiers  $P.Q=8$



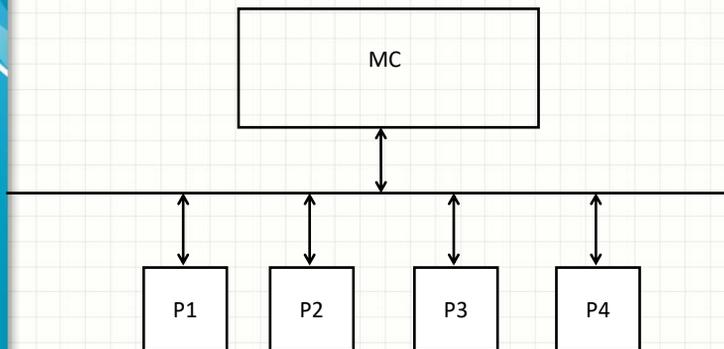
## Architectures des mémoires centrales

- Dans une architecture à **un seul processeur** : le processeur à l'exclusivité d'accéder à la mémoire. Le rendement de l'UC n'est conditionnée que par le temps d'accès à la MC.



## Architectures des mémoires centrales

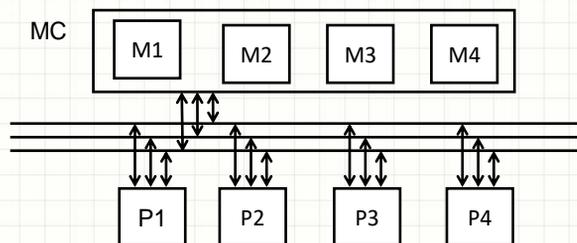
- Si le calculateur possède **plusieurs processeurs** qui fonctionnent en parallèle ( en même temps ), c'est possible que deux processeurs ou plus demandent d'accéder à la mémoire à la **même instant**.
- Si la mémoire est structurée en un **seul bloc** alors un processeur peut monopoliser la MC.
- Même si le temps d'accès est très petit, des processeurs vont être pénalisés → donc la structure de la MC est aussi importante.



55

## 1. Mémoire modulaire

- La solution est de découper la mémoire en **plusieurs modules**.
- Plusieurs bus permettent d'accéder simultanément (en même temps) à la MC.
- Possible d'avoir autant d'accès que de modules.
- On ne peut pas accéder simultanément **à un module**.



Remarques :

- Les adresses à l'intérieur d'un module sont séquentiels ( successives )
- C'est possible qu'un module soit réalisé avec des boîtiers de taille inférieure ( il faut calculer les facteurs d'extension lignes et colonnes )

56

## sélectionner un mot dans une architecture modulaire

- L'adresse est divisée en deux parties :
  - Les **bits de poids forts** pour sélectionner un module. Si le nombre de module est égale à  $n$ , alors il faut prendre  $k$  bits tel que  $2^k \geq n$
  - Les bits de **poids faibles** pour sélectionner un mot dans un module.

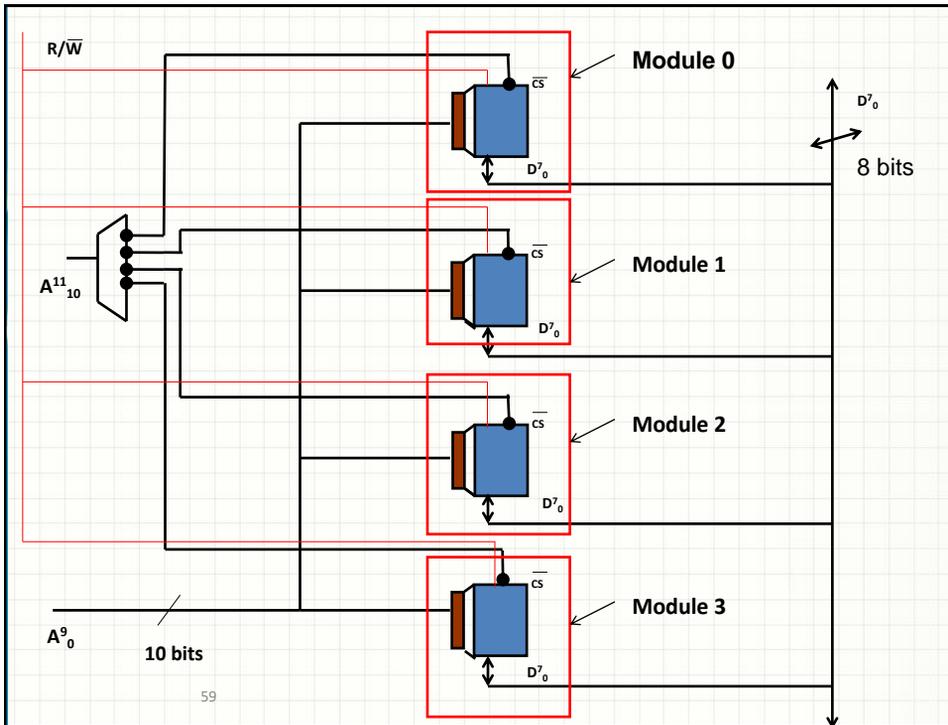


57

## Exemple

- Soit une mémoire de taille de 4 Ko. Cette mémoire est découpée en 4 modules. Donner le schéma de cette mémoire en utilisant des boîtiers de 1 Ko?
- **Solution :**
- Capacité = 4 Ko =  $4 * 2^{10} = 2^{12}$  → la taille du bus d'adresses est de 12 bits ( $A^{11}_0$ ).
- 4 modules → 2 bits du poids forts pour la sélection des modules ( $A^{11}_{10}$ )
- Les autres bits pour la sélection d'un mot dans un module ( $A^9_0$ )

58

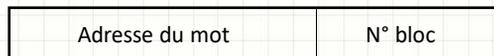


## 2. Mémoire entrelacée

- Avec une MC modulaire , c'est possible qu'un processeur **monopolise** un module ( par exemple il accède a des adresse consécutive ), Pour éviter ce problème :
  - Un module est divisé en plusieurs Blocs .
  - les adresses consécutives sont placées dans des blocs différents .
  - Le nombre de blocs représente le degré d'entrelacement.

## Sélectionner un mot dans une MC entrelacée

- L'adresse est divisée en deux parties :
  - Les **bits de poids faibles** pour sélectionner le bloc. Si on dispose de  $n$  bloc, il faut prendre  $k$  bits tel que  $2^k \geq n$ .
  - Les bits de poids forts pour sélectionner le mot dans le bloc.

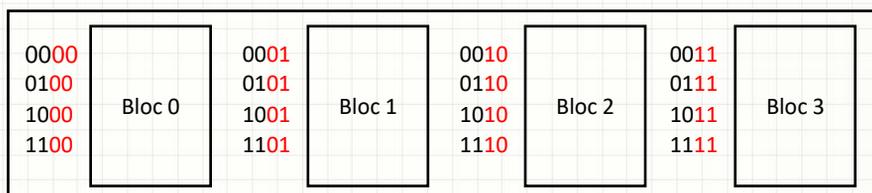


61

**Exemple 1 :** une mémoire entrelacée avec un degré d'entrelacement égale à 4, un bloc est de taille de 4 mots

- 4 blocs et la taille d'un bloc est égale à 4 mots de 4 bits → taille de la mémoire est égale à 16 mots de 4 bits.
- Il existe 4 blocs → 2 bits de poids faibles pour la sélection  $A^1_0$
- Les bits de poids forts ( $A^3_2$ ) pour sélectionner un mot dans un bloc.

- L'adresse 0000 → bloc 0 ( bits poids faible 00)
- L'adresse 0001 → bloc 1 ( bits poids faible 01)
- L'adresse 0010 → Bloc 2
- L'adresse 0011 → **Bloc 3**
- L'adresse 0100 → Bloc 0
- .....
- .....

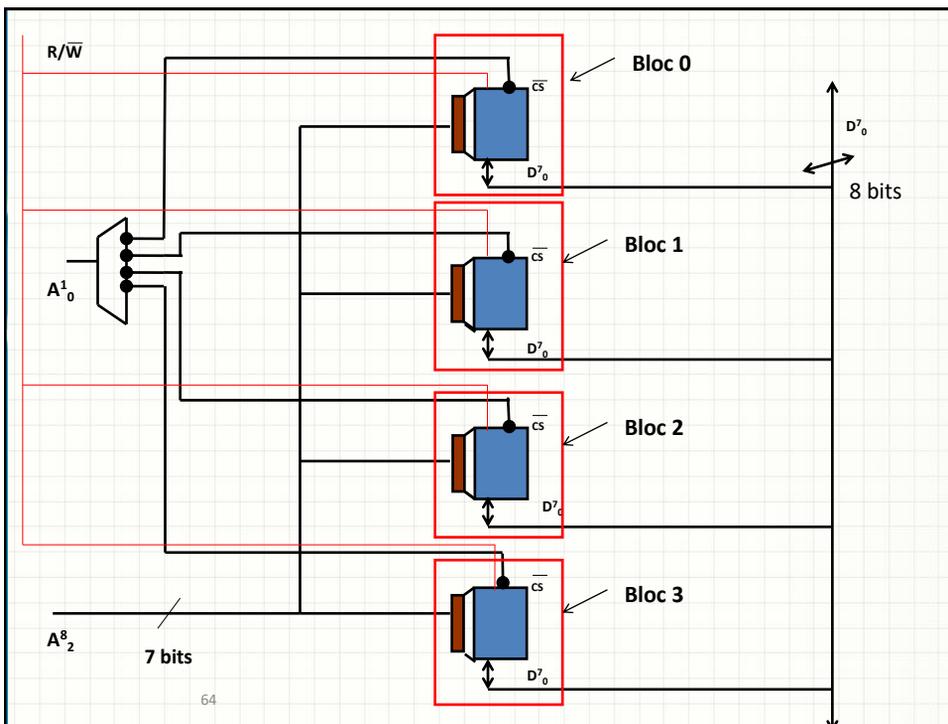


62

## Exemple 2

- Réaliser une mémoire de capacité 512 mots de 8 bits avec des boîtiers de 128 mots de 8 bits avec un degré d'entrelacement de 4.
- Capacité  $512 = 2^9$  (taille de bus d'adresses = 9)
- Taille d'un mot = 8  $\rightarrow$  taille du bus de données = 8
- 4 blocs  $\rightarrow$  taille d'un bloc =  $512/4 = 128$
- Taille d'un boîtier =  $128 * 8 \rightarrow$  un boîtier par bloc est suffisant
- 2 bits de poids faibles pour la sélection d'un bloc  $A^1_0$
- Les bits de poids fort ( $A^8_2$ ) pour sélectionner un mot dans un bloc.

63



### 3. Les mémoires modulaires entrelacées

- La MC est divisée en plusieurs modules
- Chaque module est divisé en n Blocs ( n le degré d'entrelacement)
- Pour sélectionner un mot :
  - Il faut sélectionner le module ( bits de poids forts )
  - Sélectionner le bloc dans le module ( bits de poids faibles )
  - Sélectionner le mot dans le bloc ( les bits restant )

N° module	adresse mot	N° Bloc
-----------	-------------	---------

65

### Exemple

- Réaliser une mémoire de 64 mots de 8 bits organisé en deux modules entrelacé , l'entrelacement se fait à l'intérieur ( D=2). En utilisant des circuits ( boîtiers ) de 16 mots de 8 bits.
- La taille du bus d'adresses  $k=6$  (  $64=2^6$  )  $\rightarrow A^5_0$
- Le nombre de module  $m=2$  , la taille d'un module est égale à 32 mots.
- Le nombre de bits pour sélectionner un module est égale à 1 (  $A^5$  ).
- Le nombre de blocs dans un module  $D=2 \rightarrow$  le nombre de bits nécessaire pour sélectionner un bloc est égale à 1 (  $A_0$  )
- la taille d'un bloc est égale 16 mots  $\rightarrow$  un circuit suffit pour réaliser un bloc
- Le nombre de bits nécessaire pour sélectionner une mot dans le bloc est égale à 4 (  $A^4_1$  )

66

