

Architecture Des Ordinateurs

<<Mémoires et Unité Centrale>>

E.S.P de Dakar, 2020

Intervenant: Abdou DIOP

Organisation

Introduction

- Définition , Caractéristiques des mémoires ;
- Classification des mémoires;
- La mémoire centrale ;
 - Caractéristiques d'une mémoire centrale;
 - Structure d'une mémoire centrale;
 - Conception d'une mémoire centrale ;
 - Architecture d'une Mémoire centrale (architecture modulaire et architecture entrelacée);
- Unité de calcul;
- Unité de commande.

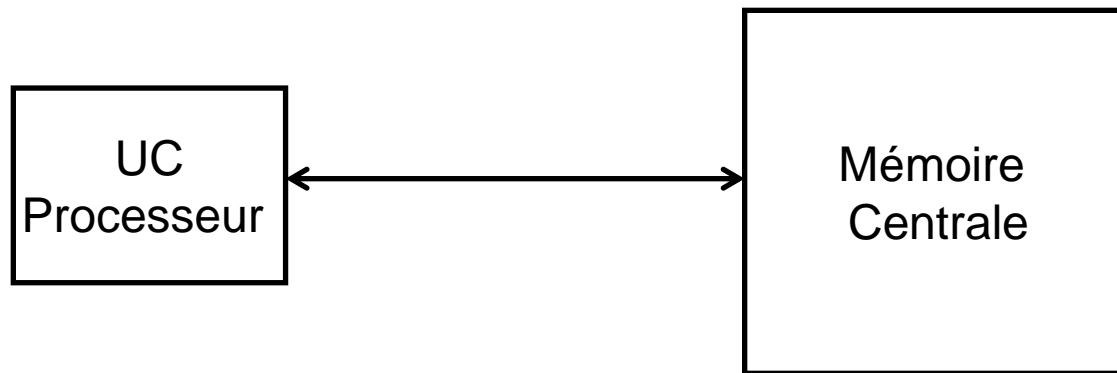
1. Introduction

- 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** .

Architecture matérielle d'une machine (architecture de Von Neumann)

L'architecture de Von Neumann est composée :

- D'une mémoire centrale,
- D'une unité centrale (UC) ou CPU (Central Processing Unit), processeur.
- Cette architecture est la base des architectures des ordinateurs.



L'Unité Centrale (UC)

- L'unité centrale (appelée aussi processeur) a pour rôle d'exécuter les programmes.
- L'UC est composée d'une unité arithmétique et logique (UAL) et d'une unité de contrôle.
 - L'unité arithmétique et logique réalise une opération élémentaire (addition, soustraction, multiplication, . . .) du processeur à chaque top d'horloge.
 - L'unité de commande contrôle les opérations sur la mémoire (lecture/écriture) et les opérations à réaliser par l'UAL selon l'instruction en cours d'exécution.
- Pour pouvoir effectuer les opérations sur des données et exécuter des programmes l'UC doit disposer d'un espace de travail . Cette espace de travail s'appelle la mémoire centrale.

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 dure
 - Une disquette
 - Un flash disque...
- La mémoire peut être **dans** le processeur (des registres) , **interne** (Mémoire centrale ou principale) ou **externe** (Mémoire secondaire).

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

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**).

3.Caractéristiques des mémoires

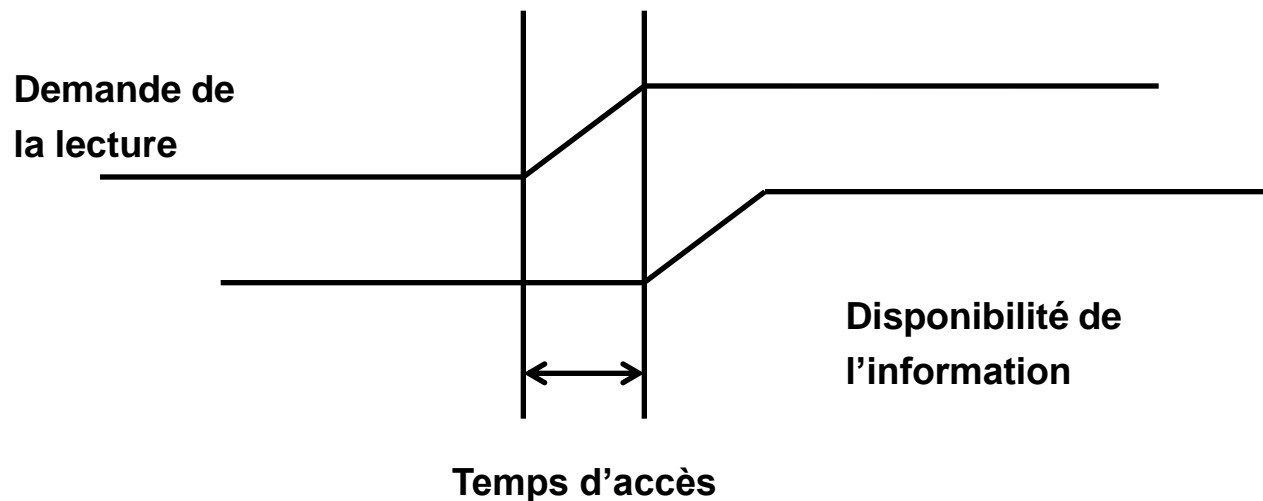
3. Mode d'accès à l'information (lecture /écriture)

- 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**.

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.

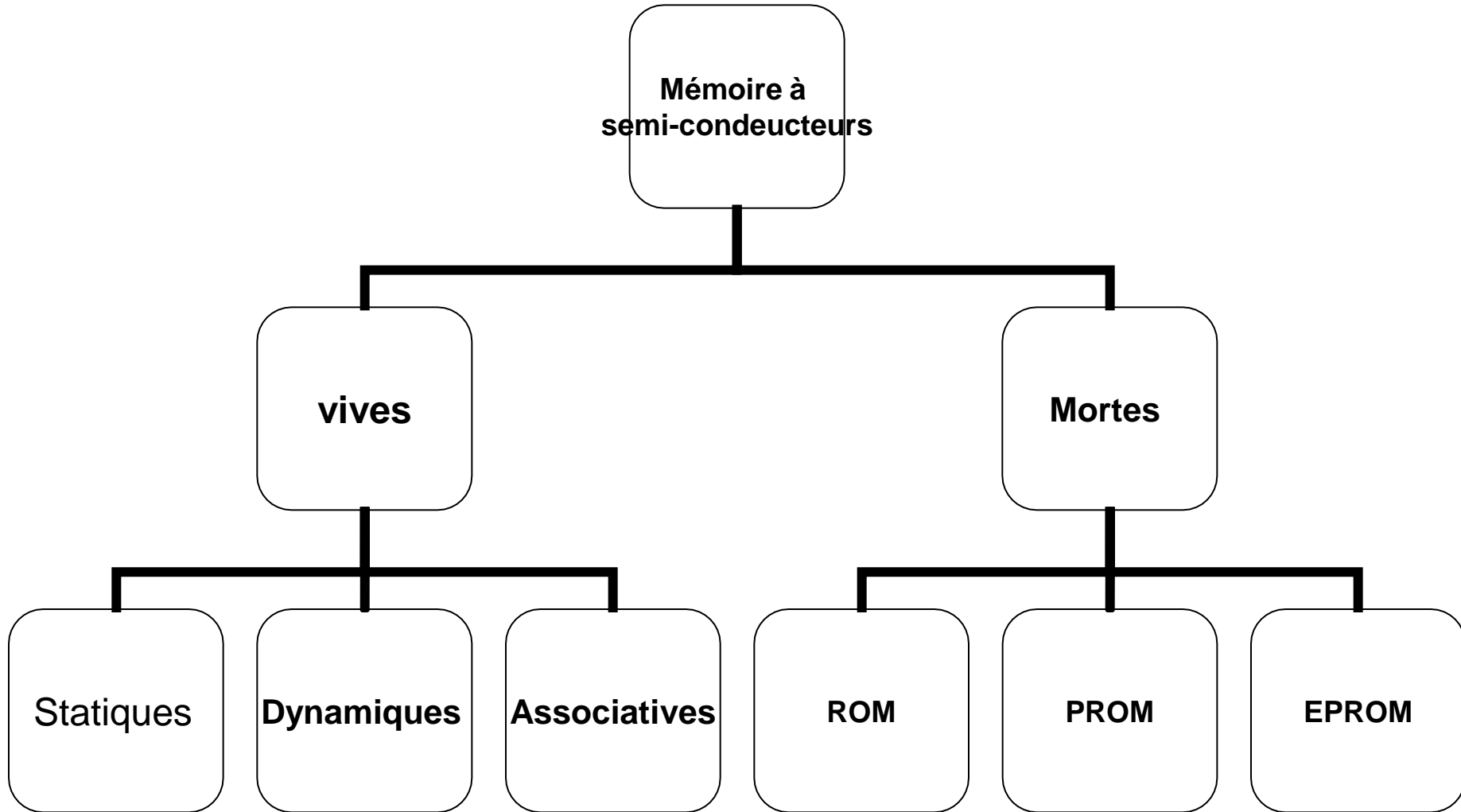


Le temps d'accès est un **critère important** pour déterminer **les performances** d'une mémoire ainsi que les performances d'une machine.

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,..)

5. Mémoire à semi-conducteur



La mémoire centrale

RAM : Random Acces memory
Mémoire à accès aléatoire

1. C'est quoi une 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).

2. Caractéristiques de la mémoire centrale

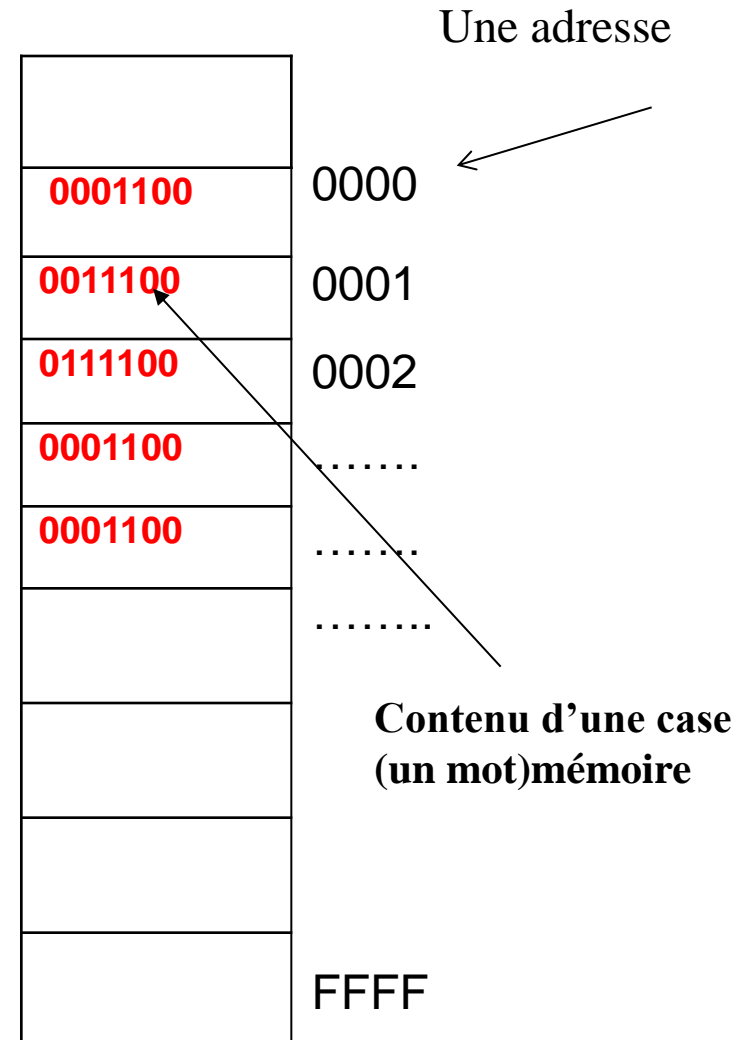
- La mémoire centrale est réalisée à base de semi-conducteurs.
- La mémoire centrale est une **mémoire vive** : accès en lecture et écriture.
- La mémoire centrale est dite à **accès aléatoire** (RAM : Random Acces Memory) c'est-à-dire que le temps d'accès à l'information est indépendant de sa place 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)

3. 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 .

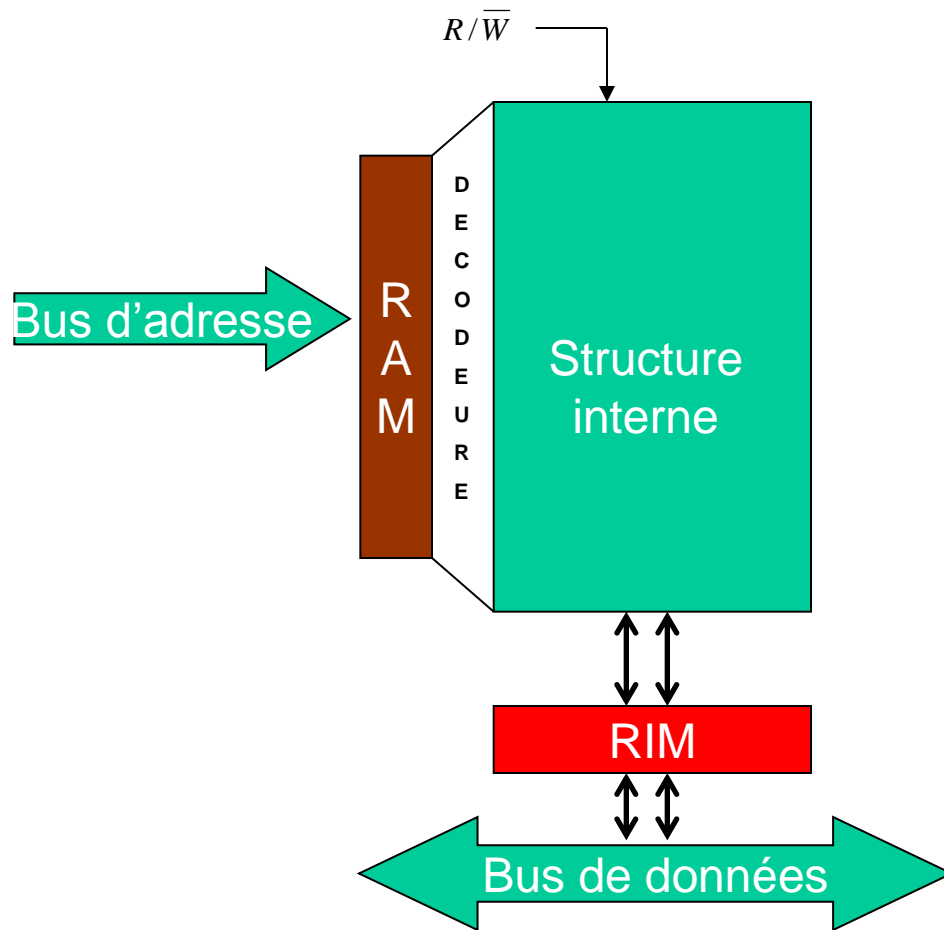
4. Vue logique de la mémoire centrale

- La mémoire centrale peut être vue 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.



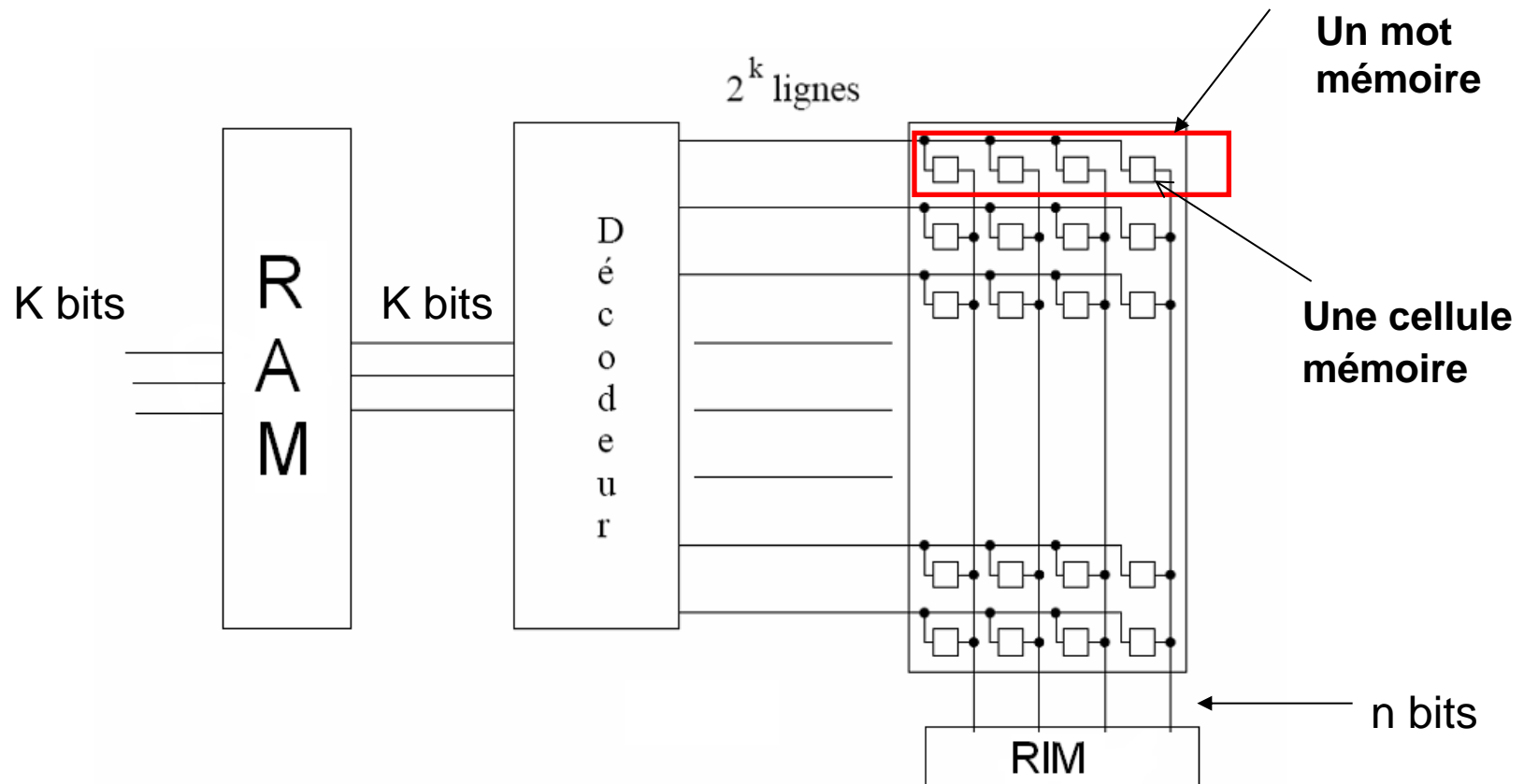
5. Structure physique d'une mémoire centrale

- **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/\overline{W}** : commande de lecture/écriture , cette commande permet de lire ou d'écrire dans la mémoire (si $R/\overline{W}=1$ alors lecture sinon écriture)
- Bus d'adresses de taille **k bits**
- Bus de données de taille **n bits**



5.1. Comment sélectionner 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**.



5.2 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 ?

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

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

5.3 Comment lire 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)

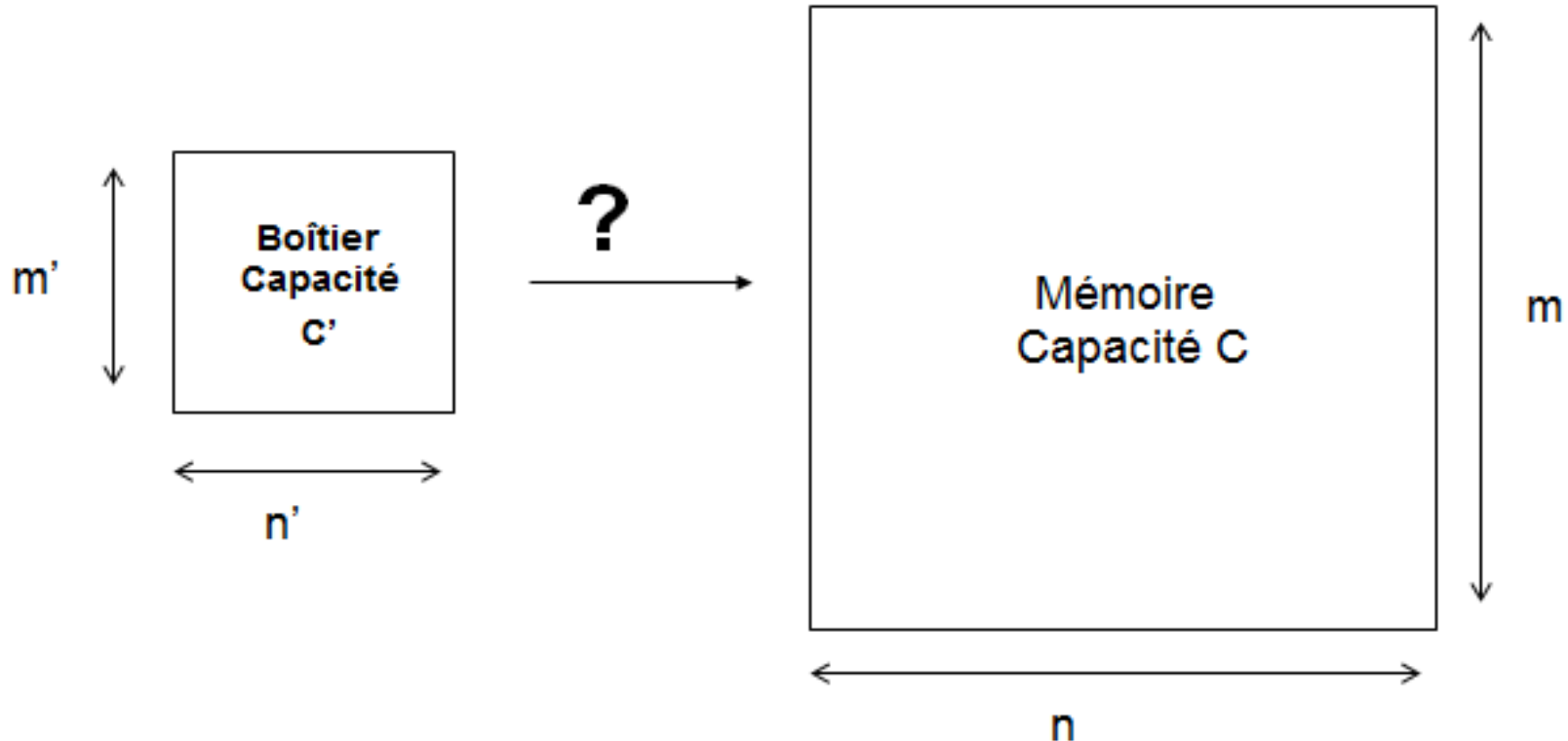
5.4 Comment écrire une information ?

- Pour écrire une information en MC il faut effectuer les opérations suivantes:
 - Charger dans le RAM l'adresse du mot où 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 .

6. Conception des MC

Problème ?

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



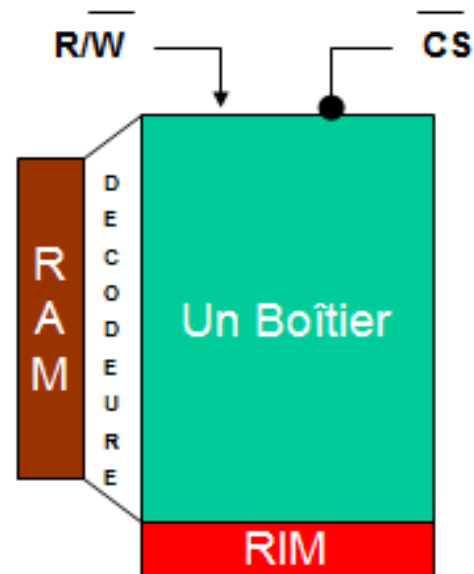
Structure d'un boîtier

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

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

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

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



Solution

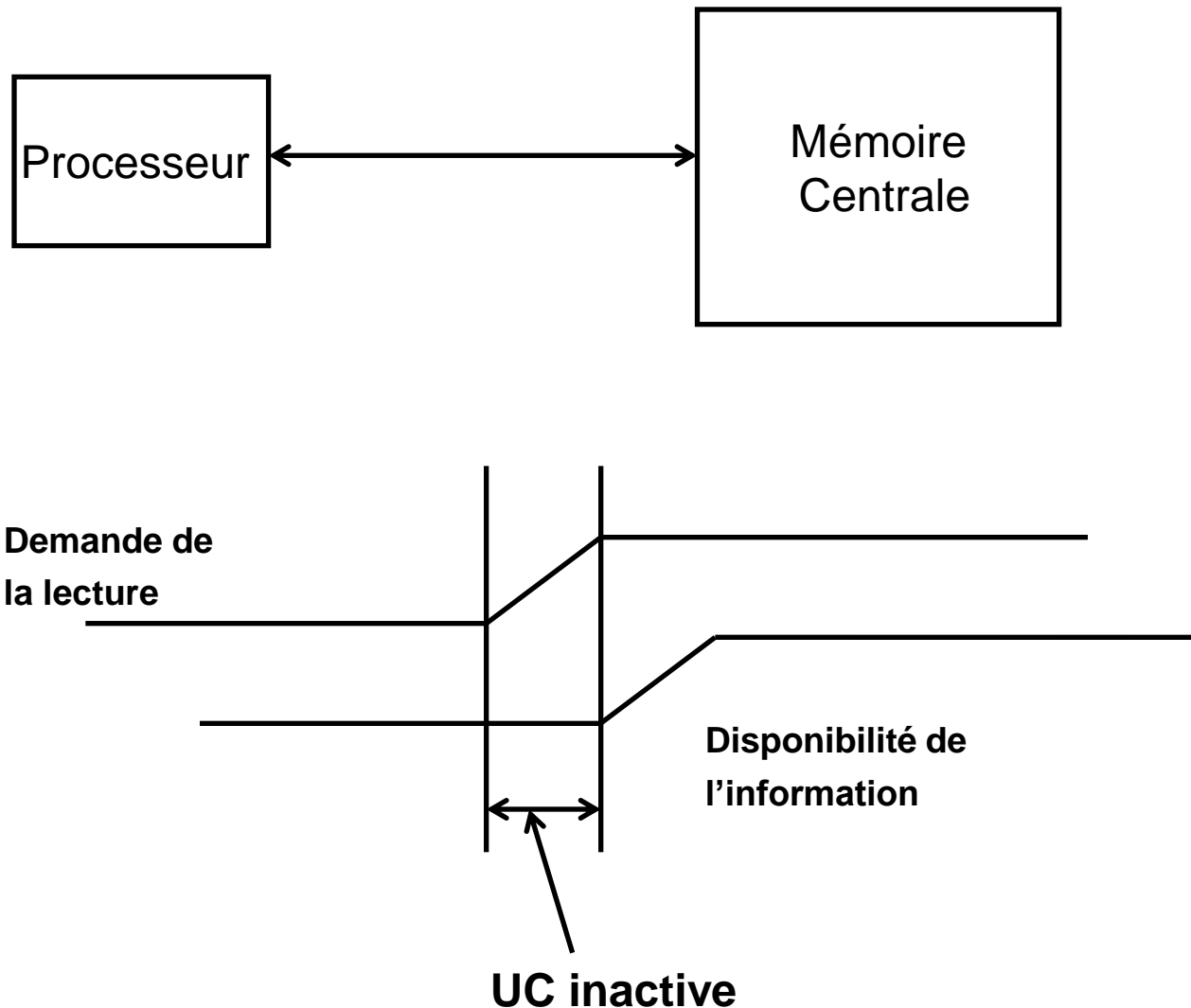
- Soit M une mémoire de capacité C , tel que **m** est le nombre de mot et **n** la taille **d'un mot**.
- Soit M' un boîtier de capacité C' , tel que m' le nombre de mot et n' la taille d'un mot.
- On suppose que $C > C'$ ($m \geq m'$, $n \geq n'$)
- Quel est le **nombre de boîtiers** M' nécessaire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

Solution (suite)

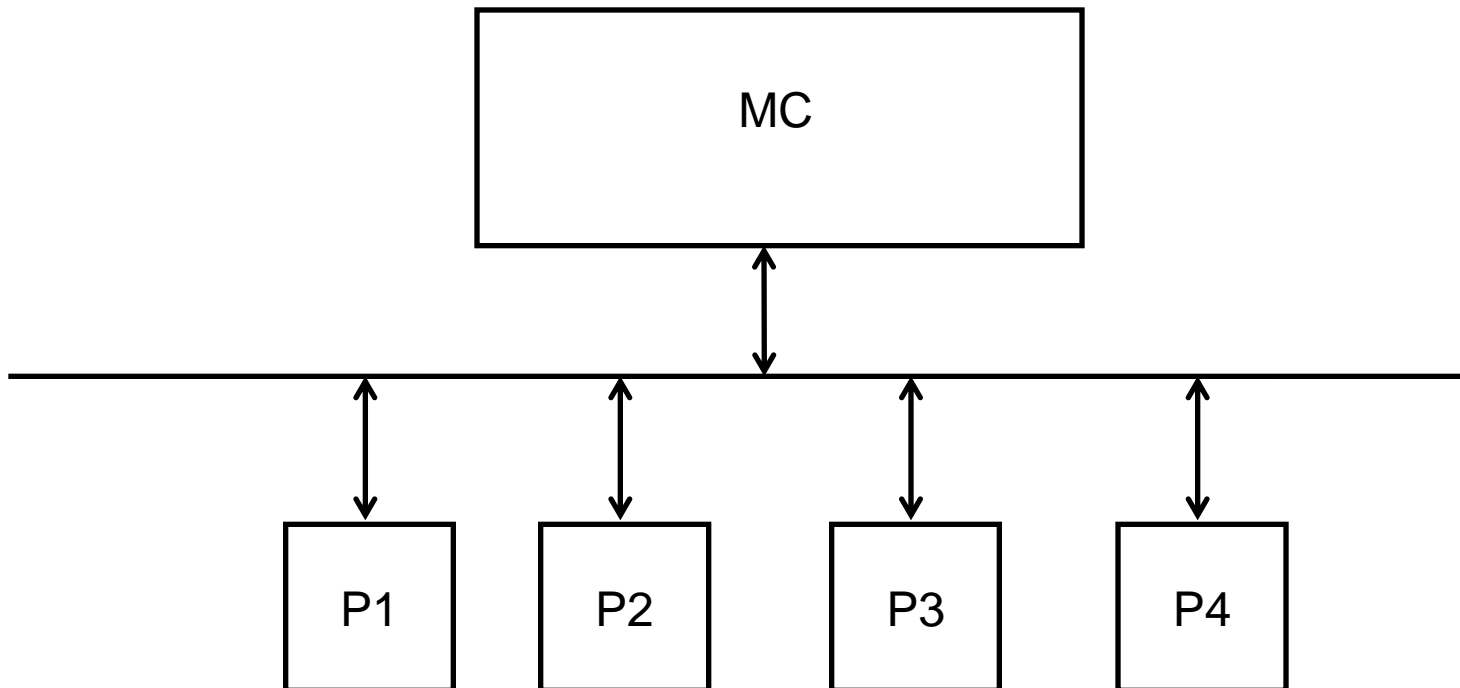
- P : permet de déterminer le nombre de boîtiers M' nécessaire pour obtenir le **nombre de mots** de la mémoire M (extension lignes).
- Q : permet de déterminer le nombre de boîtier M' nécessaire pour obtenir la **taille de mot** de la mémoire M (extension mots ou extension colonnes).
- **P.Q** donne 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.

7. 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.

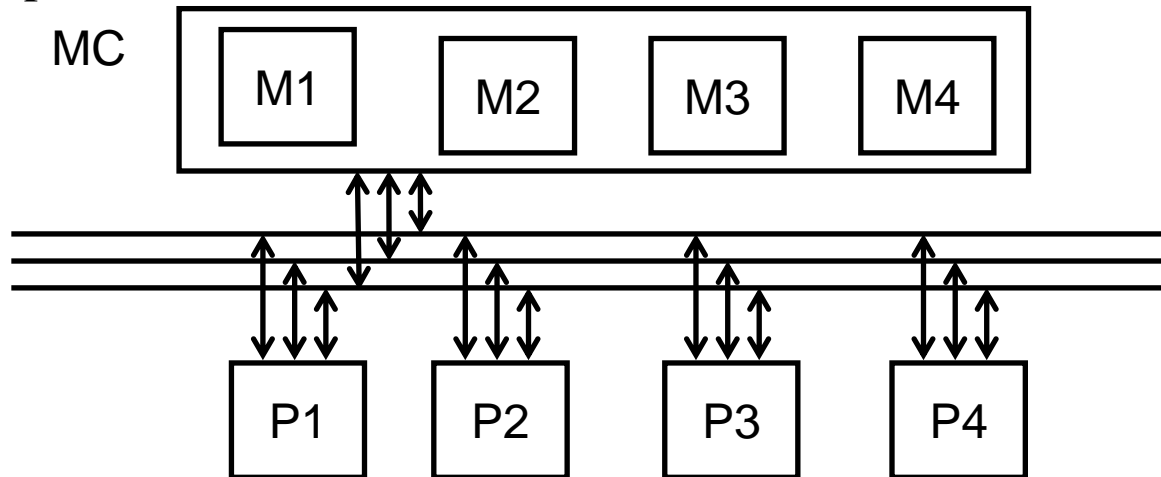


- Si le calculateur possède **plusieurs processeurs** qui fonctionnent en parallèle (en même temps), c'est possible que deux processeurs ou plus demandent d'accéder à la mémoire à la **même instant**.
- Si la mémoire est structurée en un **seul bloc** alors un processeur peut monopoliser la MC.
- Même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



7.1 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)

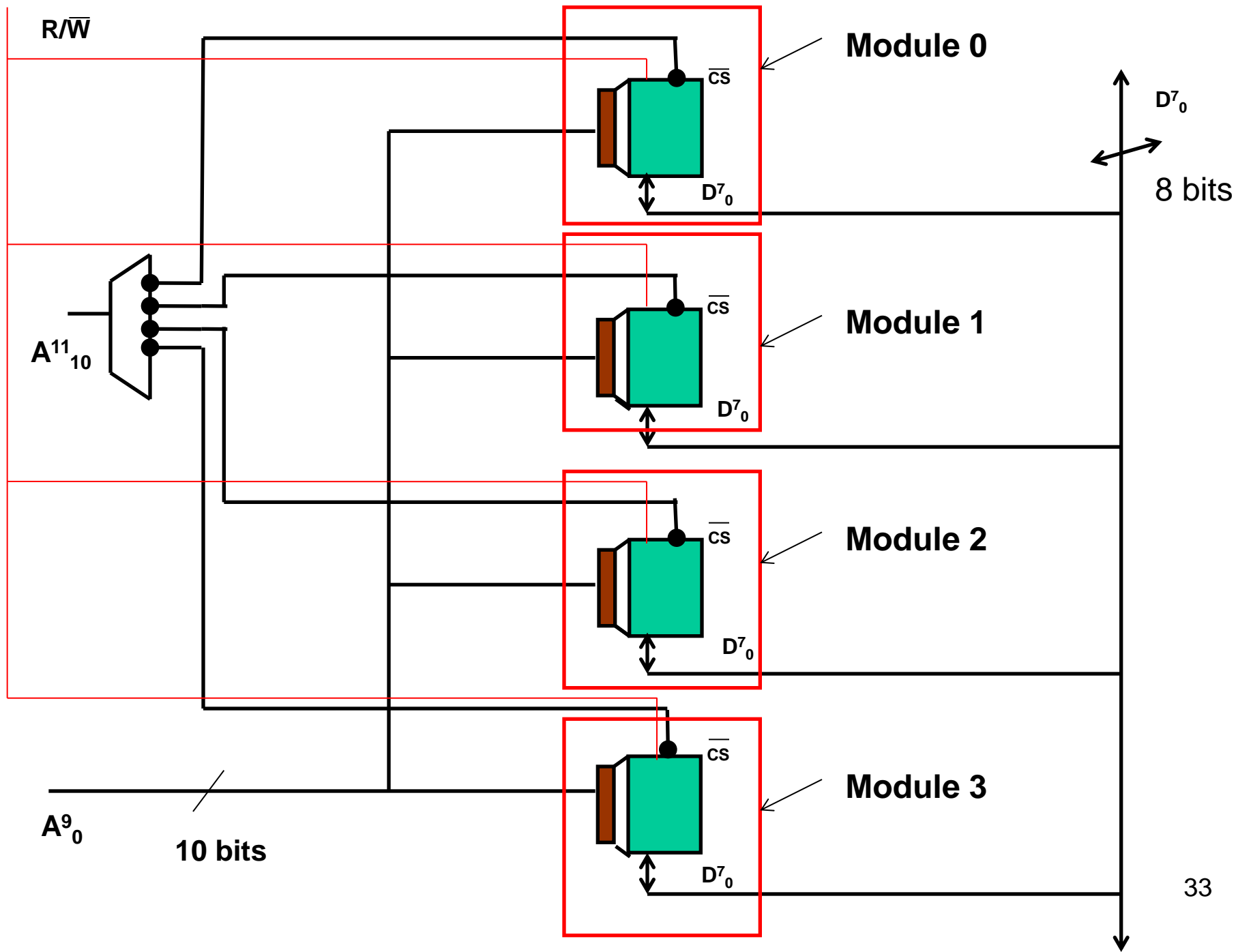
Comment sélectionner un mot dans une architecture modulaire ?

- 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.

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

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)



- **Exercice 1 :**

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 512 mots de 8 bits?

- **Exercice2 :**

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 512 mots de 4 bits?

7.2 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 consécutive sont placé dans des bloc différents .
 - Le nombre de blocs représente le degré d'entrelacement.

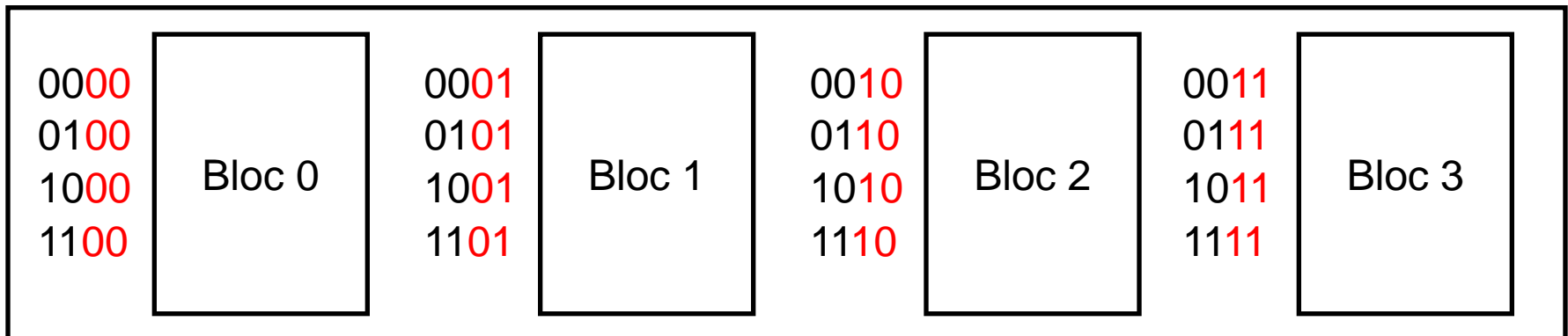
Sélectionner un mot dans une MC entrelacée

- 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 .



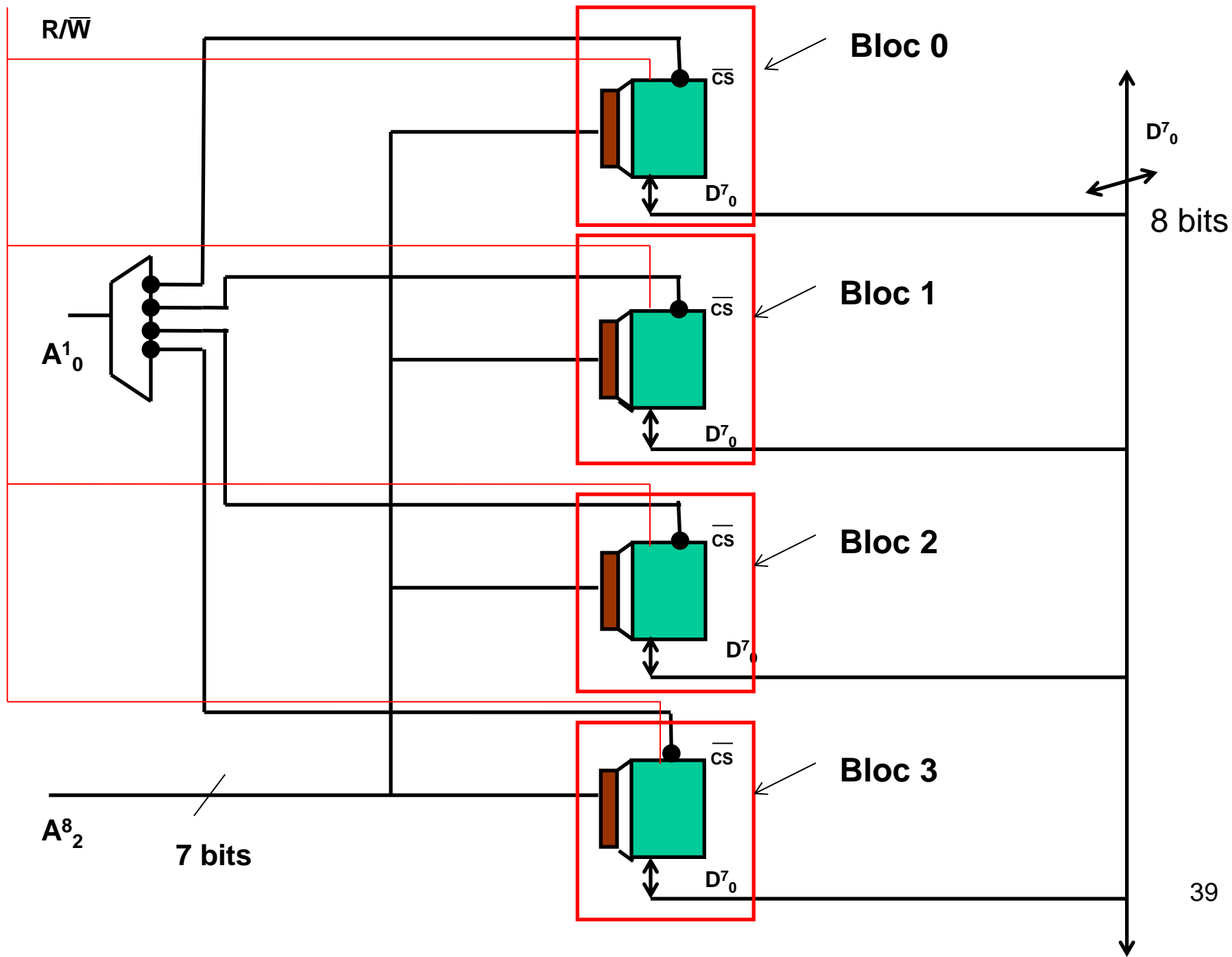
Exemple 1 : une mémoire entrelacée avec un degré d'entrelacement égale à 4 , un bloc est de taille de 4 mots

- 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
-
-



Exemple 2

- Réaliser une mémoire de capacité 512 mots de 8 bits avec des boîtiers de 128 mots de 8 bits avec un degré 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.



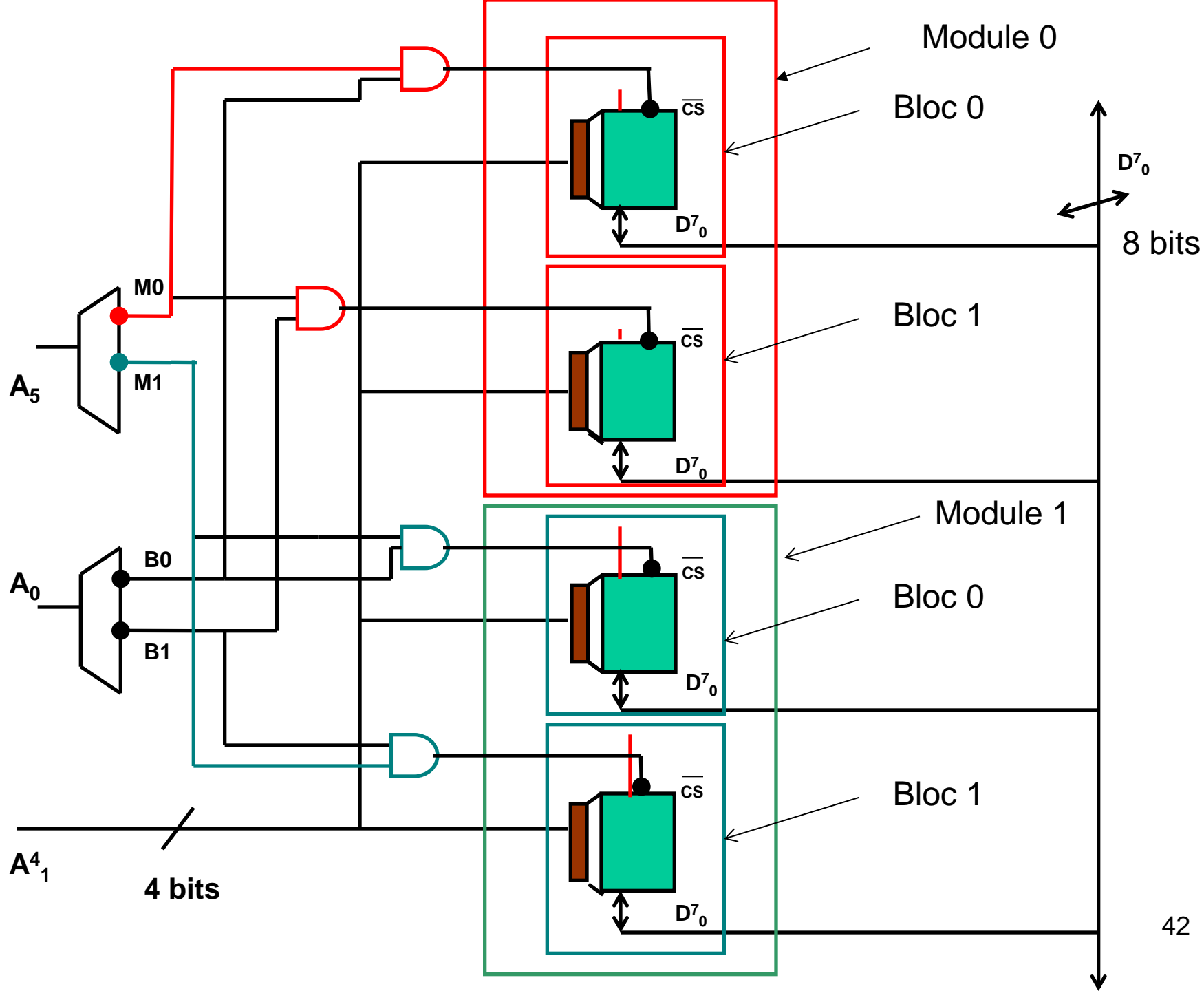
7.3 Les mémoires modulaires entrelacées

- 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
-----------	-------------	---------

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)

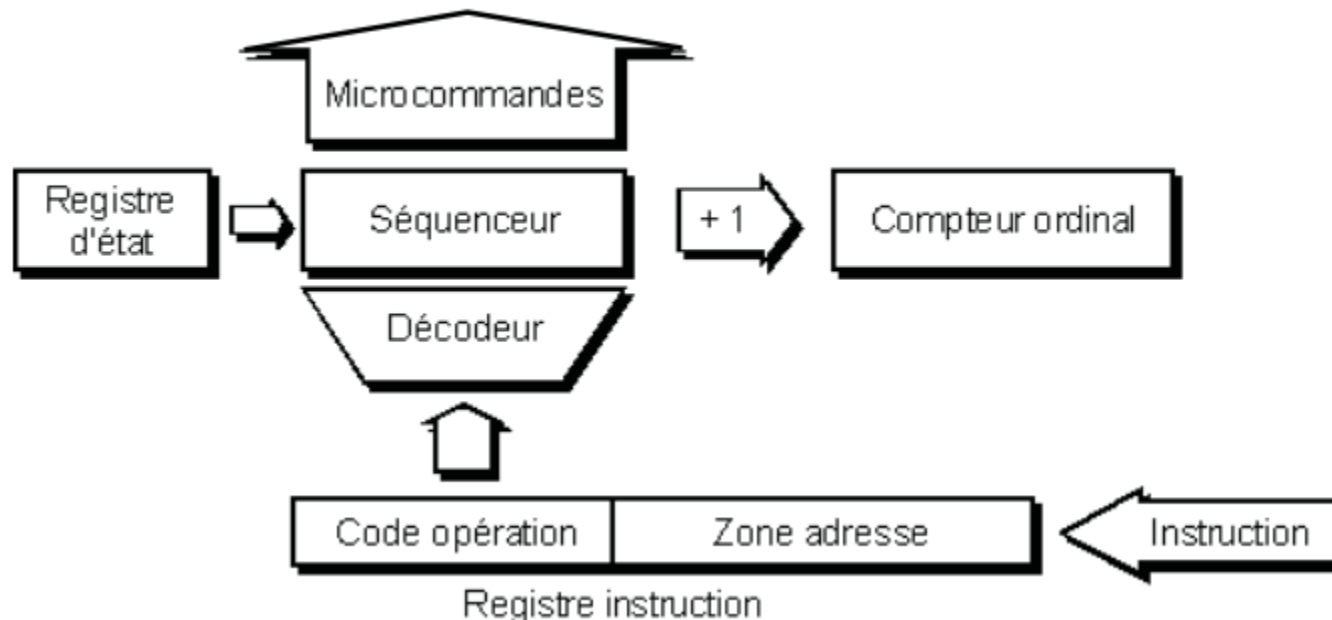


8. Unité Centrale

8.1 Unité de commande

Rôle

Nous avons déjà vu que l'unité de commande avait pour rôle de gérer le bon déroulement du programme. Elle doit donc prendre en compte, les unes après les autres, chacune des instructions ; décoder l'instruction en cours, lancer les ordres (**microcommandes**) aux composants du système qui participent à la réalisation de cette instruction ; puis aller chercher une nouvelle instruction et recommencer. Pour cela l'unité de commande est constituée d'un certain nombre de composants internes qui assurent chacun une fonction bien déterminée.



8. Unité Centrale

8.1 Unité de commande

Composants de l'unité centrale

➤ Le registre instruction

L'instruction que l'unité de commande va devoir traiter est chargée préalablement dans un registre particulier appelé **registre instruction**.

➤ Le séquenceur

En fonction de l'instruction à traiter – qui vient d'être chargée dans le **registre instruction** – l'unité de commande va devoir émettre un certain nombre de **microcommandes** vers les autres composants du système. Ces ordres ne seront bien évidemment pas émis n'importe quand, ni vers n'importe quel composant, mais respectent une chronologie bien précise selon le type d'instruction à exécuter. Cette chronologie (séquencement) est rythmée par une **horloge** interne au système. On comprend intuitivement que plus cette fréquence est élevée et plus l'unité centrale de traitement travaille « vite » – mais ce n'est pas le seul facteur de rapidité.

Le composant qui émet ces microcommandes – en fonction du code opération (préalablement décodé) de l'instruction située dans le registre instruction – est le **séquenceur**, qui, comme son nom le laisse entendre envoie une séquence de microcommandes vers les composants impliqués par l'instruction

8. Unité Centrale

8.1 Unité de commande

➤ Le registre d'état

Pour exécuter correctement son travail, le séquenceur doit en outre connaître l'état d'un certain nombre d'autres composants et disposer d'informations concernant la ou les opérations qui ont déjà été exécutées (par exemple, doit-on tenir compte dans l'addition en cours d'une éventuelle retenue préalable générée par une addition précédente). La connaissance de ces informations se fait à l'aide d'un autre composant appelé registre indicateur ou, plus couramment **registre d'état**, et qui, grâce à des indicateurs (**drapeaux** ou *flags*), qui ne sont en fait que des registres de bascules, va mémoriser certaines informations telles que retenue préalable, imparité, résultat nul, etc.

➤ Le compteur ordinal

Quand le séquenceur a fini de générer les microcommandes nécessaires, il faut qu'il déclenche le chargement, dans le registre instruction, d'une nouvelle instruction. Pour cela il dispose d'un registre « compteur d'instructions » (qui en fait serait plutôt un « pointeur » d'instructions).

Ce registre porte le nom de **compteur ordinal** (*Program Counter* ou *Instruction Pointer*). Il s'agit d'un registre spécialisé, qui est chargé automatiquement par le système lors du lancement d'un programme, avec **l'adresse** mémoire de la première instruction du programme à exécuter.

Par la suite, à chaque fois qu'une instruction a été chargée dans le registre instruction et qu'elle s'apprête à être exécutée, ce compteur ordinal est **incrémenté** (augmenté) de manière à pointer sur l'adresse de la **prochaine instruction** à exécuter.

8. Unité Centrale

8.2 unité arithmétique et logique

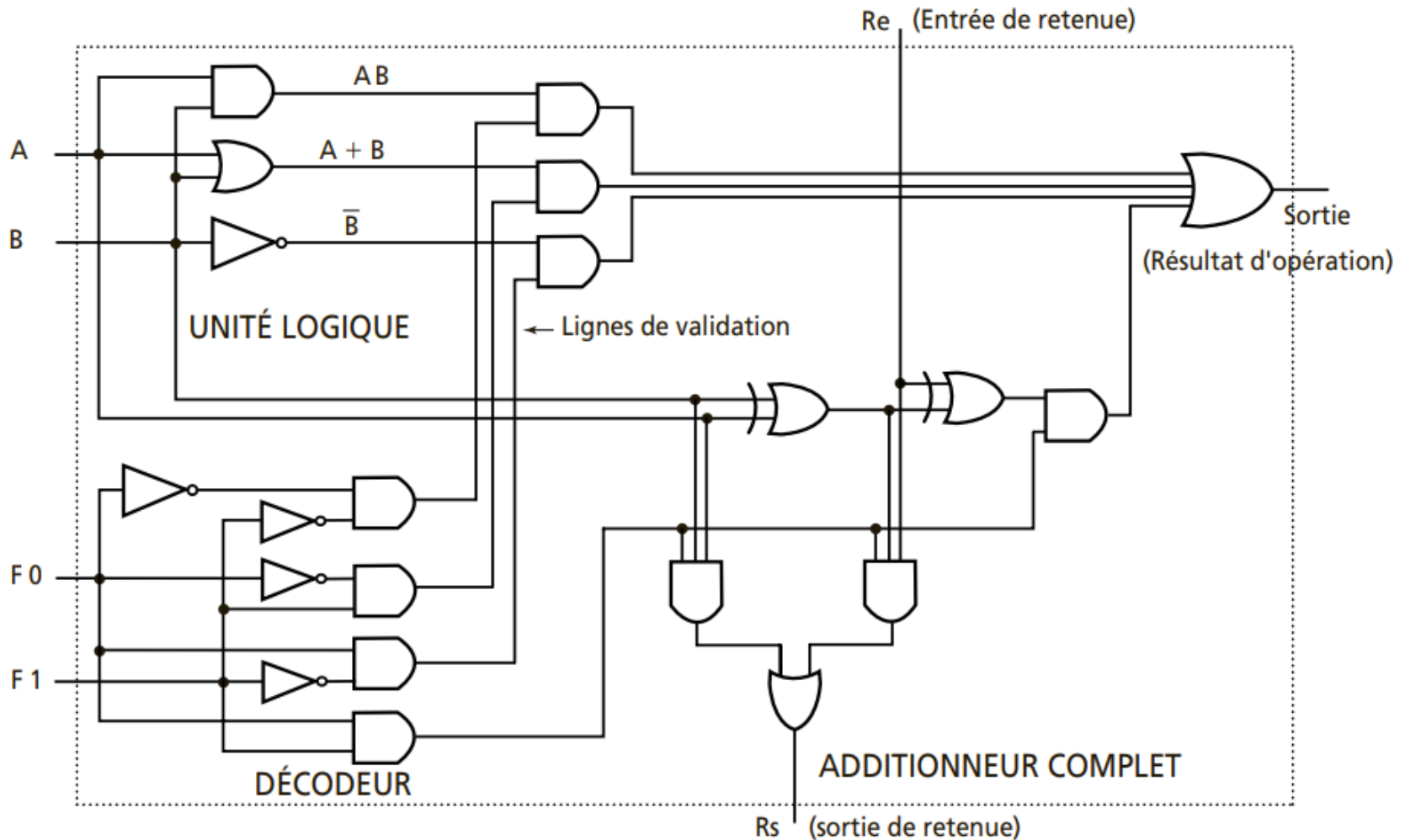
L'unité arithmétique et logique (UAL) est composée de circuits logiques tels que les additionneur, soustracteur, comparateurs logiques... selon le degré de technicité et d'intégration atteint par le fabricant du circuit intégré. Les données à traiter se présentent donc aux entrées de l'UAL, sont traitées, puis le résultat est fourni en sortie de cette UAL et généralement stocké dans un registre dit accumulateur. Ces deux composants, **unité de commandes** et **unité arithmétique et logique** constituent donc une **unité centrale**, aussi appelée **unité de traitement** ou **CPU** (*Central Processor Unit*).

Cette unité centrale (UC) permet ainsi :

- d'acquérir et décoder les instructions d'un programme, les unes après les autres en général – mais ce n'est pas toujours aussi simple ;
- de faire exécuter par l'UAL les opérations arithmétiques et logiques commandées par l'instruction ;
- de gérer les adresses des instructions du programme grâce au compteur ordinal ;
- de mémoriser l'état interne de la machine sous forme d'indicateurs grâce au registre d'état ;
- de fournir les signaux de commande et de contrôle aux divers éléments du système.

8. Unité Centrale

8.2 Exemple simple d'UAL 1 bit



D'après : Tanenbaum, Architecture de l'ordinateur, Inter-Editions.

8. Unité Centrale

8.3 Les bus

Les composants de l'unité centrale communiquent entre eux et avec les composants extérieurs à l'UC à l'aide de liaisons électriques (fils, circuits imprimés ou liaisons dans le circuit intégré), qui permettent le transfert des informations électriques binaires. Ces ensembles de « fils » constituent les **bus**.

Le bus de données

Le **bus de données** (*Data Bus*) permet, comme son nom l'indique, le transfert de données (instructions, ou données à traiter) entre les composants du système. Suivant le nombre de « fils » que compte le bus, on pourra véhiculer des mots de 8, 16, 32 ou 64 bits.

Ce nombre de bits pouvant circuler en même temps (en parallèle) détermine ce que l'on appelle la **largeur du bus**. Les informations pouvant circuler dans les deux sens sur un tel bus (de la mémoire vers l'unité centrale ou de l'unité centrale vers la mémoire par exemple), le bus de données est dit **bidirectionnel**.

Le bus d'adresses

Le **bus d'adresses** (*Address Bus*) comme son nom l'indique, est destiné à véhiculer des adresses, que ce soit l'adresse de l'instruction à charger dans le registre instruction, ou celle de la donnée à charger dans un registre particulier ou à envoyer sur une entrée de l'UAL. La largeur du bus d'adresses détermine la taille de la mémoire qui sera directement adressable (adressage physique) par le microprocesseur. Ainsi, avec un bus d'adresses d'une largeur de 16 bits on peut obtenir 2^{16} combinaisons soit autant de cellules mémoires où loger instructions ou données (avec un bus d'adresses de 32 bits on peut ainsi adresser 4 Go de mémoire physique). Dans ce type de bus les adresses ne circulent que dans le sens unité centrale vers mémoire, ce bus est dit **unidirectionnel**.

Le bus de commandes

Le **bus de commandes** (*Control Bus*) permet aux microcommandes générées par le séquenceur de circuler vers les divers composants du système.

8. Unité Centrale

8.4 FONCTIONNEMENT DE L'UNITÉ CENTRALE

En examinant ce que nous avons déjà énoncé quant au traitement des instructions dans l'unité centrale, on peut dire que pour chaque instruction d'un programme on peut distinguer :

- une phase de **recherche** de l'instruction,
- une phase de **traitement** de l'instruction.

Nous allons examiner de plus près ces deux phases à partir d'un exemple simplifié de fonctionnement. Pour faciliter la compréhension, nous utiliserons des mnémoniques représentant chacune des microcommandes que le séquenceur pourra générer (ces commandes mnémoniques sont **tout à fait arbitraires** et n'existent pas en tant que telles dans une véritable machine). Dans la réalité, les microcommandes correspondent en fait à de simples impulsions électriques émises par le séquenceur. Sur le schéma complet de l'unité centrale que nous présentons en page suivante, vous pouvez observer que les microcommandes issues du séquenceur sont représentées comme de petites flèches que l'on retrouve au niveau des relations entre les divers constituants de l'UC et qui pointent sur une zone grisée : cette zone peut être assimilée à une porte (c'est d'ailleurs souvent une porte logique) que la microcommande va « ouvrir », permettant ainsi le passage de l'information entre les composants.