



UNIVERSITÉ INTERNATIONALE DE CASABLANCA

Informatique Industrielle: système à μ processeurs

1



Sommaire:

- Objectifs
- Informatique Industrielle
- Du système câblé au microprogrammé
- Systèmes à microprocesseur
 - ✓ Système minimal
 - ✓ Microprocesseur
 - ✓ Mémoire
 - ✓ Interfaces d'entrée/sortie
- Exploitation d'un système à μp
- Du microprocesseur au microcontrôleur



Objectifs du cours :

Le but de ce cours est de vous familiariser avec le fonctionnement et l'utilisation des systèmes à microprocesseurs et de vous rendre capable de **choisir**, de **programmer**, de **mettre en œuvre** un microcontrôleur dans un système industriel automatisé.

Le cours est composé de cours théoriques, de séances de travaux dirigés et de travaux pratiques et réparti en deux parties:

➤ **Non dédié à un microcontrôleur:**

- ✓ Logique câblée vs logique programmée
- ✓ Architecture des systèmes microprogrammés
- ✓ Constituants d'un système microprogrammé
- ✓ Fonctionnement d'un système microprogrammé

➤ **Dédié à un microcontrôleur:**

- ✓ Architecture
- ✓ Jeu d'instructions et modes d'adressage
- ✓ Notion d'E/S d'interruption
- ✓ Programmation

Définition:

- ❑ L'informatique industrielle développe des interfaces entre l'informatique et les appareils industriels.
- ❑ Les outils informatiques, qui vont de l'analyse à la programmation, sont alors mis en œuvre pour réaliser l'interface entre l'informatique, l'électronique, la mécanique, la robotique, l'électrotechnique, etc. à vocation industrielle et qui ne sont pas uniquement à base d'ordinateurs.

Domaines d'application:

Alarme, automobile, aviation, ferroviaire, instrumentation, médicale, téléphonie mobile, terminaux de paiement automatique ...



Applications:

- ✓ Automates, Robotique
- ✓ Acquisition et exploitation de données industrielles
- ✓ Systèmes temps réel
- ✓ Systèmes embarqués
- ✓





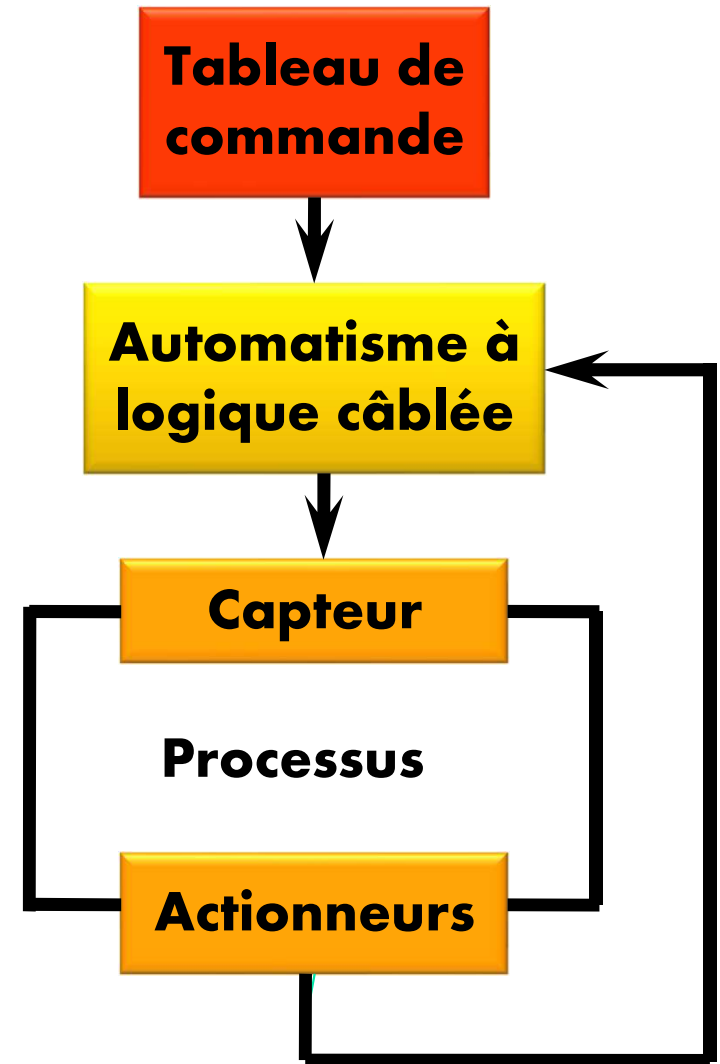
Sommaire:

- Objectifs
- Informatique Industrielle
- **Du système câblé au microprogrammé**
- Systèmes à microprocesseur
 - ✓ Système minimal
 - ✓ Microprocesseur
 - ✓ Mémoire
 - ✓ Interfaces d'entrée/sortie
- Exploitation d'un système à μp
- Du microprocesseur au microcontrôleur

Logique câblée :

Avant l'avènement des microprocesseurs, les systèmes logiques étaient conçus à l'aide des circuits intégrés logiques accomplissant les opérations logiques et arithmétiques fondamentales.

Il s'agit de la logique câblée et qui relève du matériel (hardware).



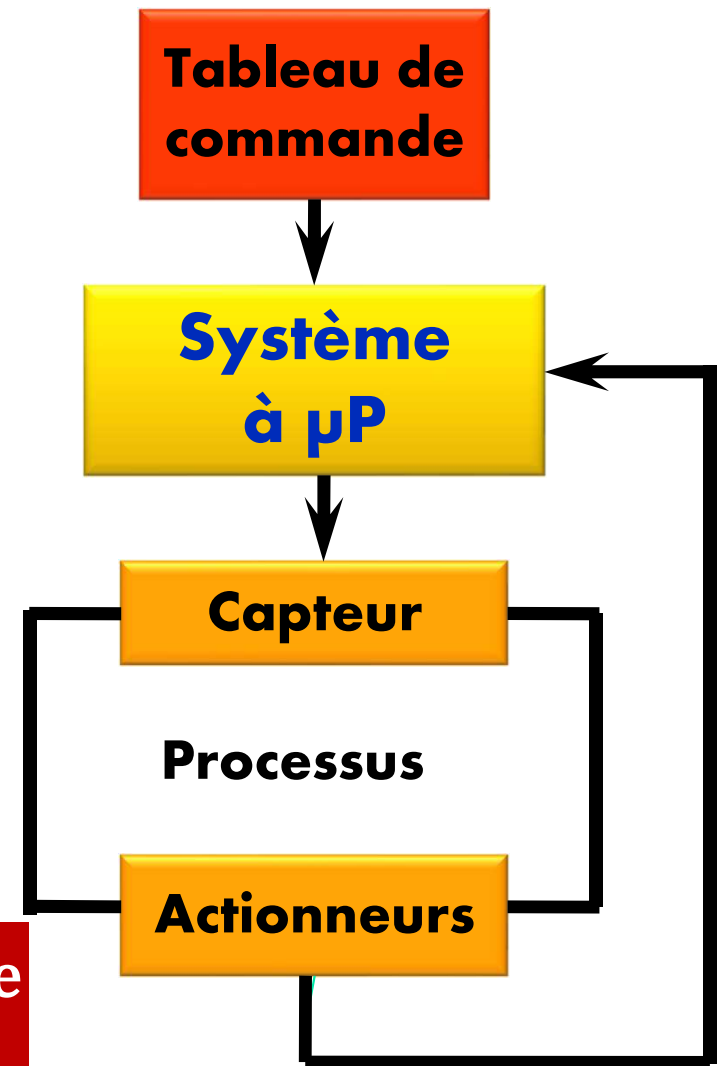
Logique programmée :

L'apparition des microprocesseurs sur le marché permet de concevoir tout autrement un système logique.

A l'aide des instructions (mot binaire), on configure le microprocesseur pour réaliser une fonction choisie.

Ainsi pour réaliser un ensemble d'opérations, il faut donner une suite d'instructions appelée programme.

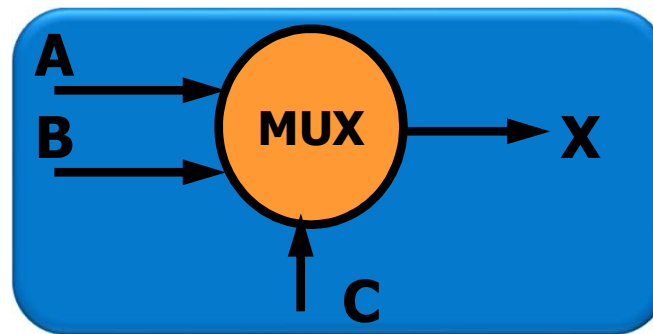
Une telle logique est dite logique programmée et qui correspond au logiciel (software).



Systeme câblé ou programmé ?

- *Pourquoi faire appel à une unité programmable?*
- *Pourquoi ne pas utiliser une solution totalement matérielle ?*

Exemple:



vs

```
IF (C==0)
  X=A;
ELSE
  X=B;
```

Hardware (HW):

- ✓ *L'architecture est conçue pour réaliser toujours la même fonction. Il n'y a pas de programme.*
- ✓ *Solution Chère*
- ✓ *Temps d'exécution Rapide*

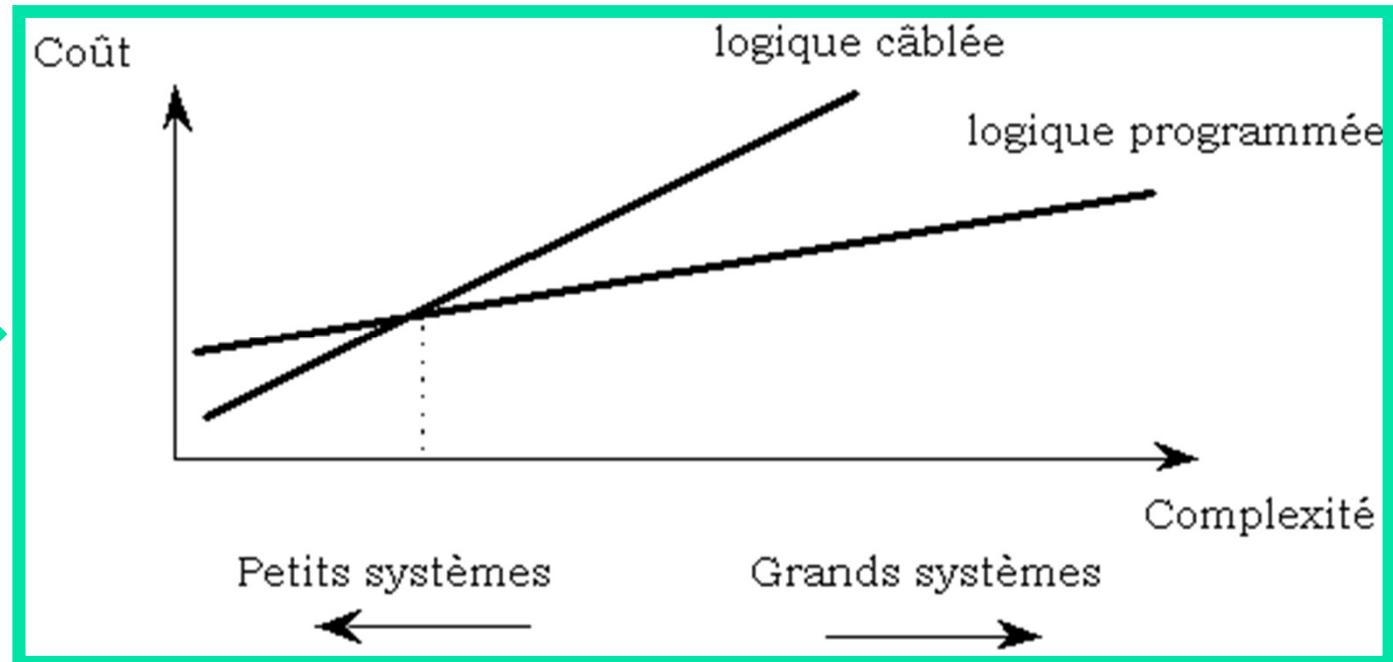
Software (SW):

- ✓ *Permet de spécialiser plus rapidement et plus facilement un STN.*
- ✓ *Mise au point du système facile et rapide*
- ✓ *Solution de faible coût.*
- ✓ *Solution moins Rapide*

- *Le SW domine le coût dans la plupart des systèmes de traitement numérique (STN).*
- *La part du logiciel dans les STN ne cessent d'augmenter.*

Systeme câblé ou programmé ?

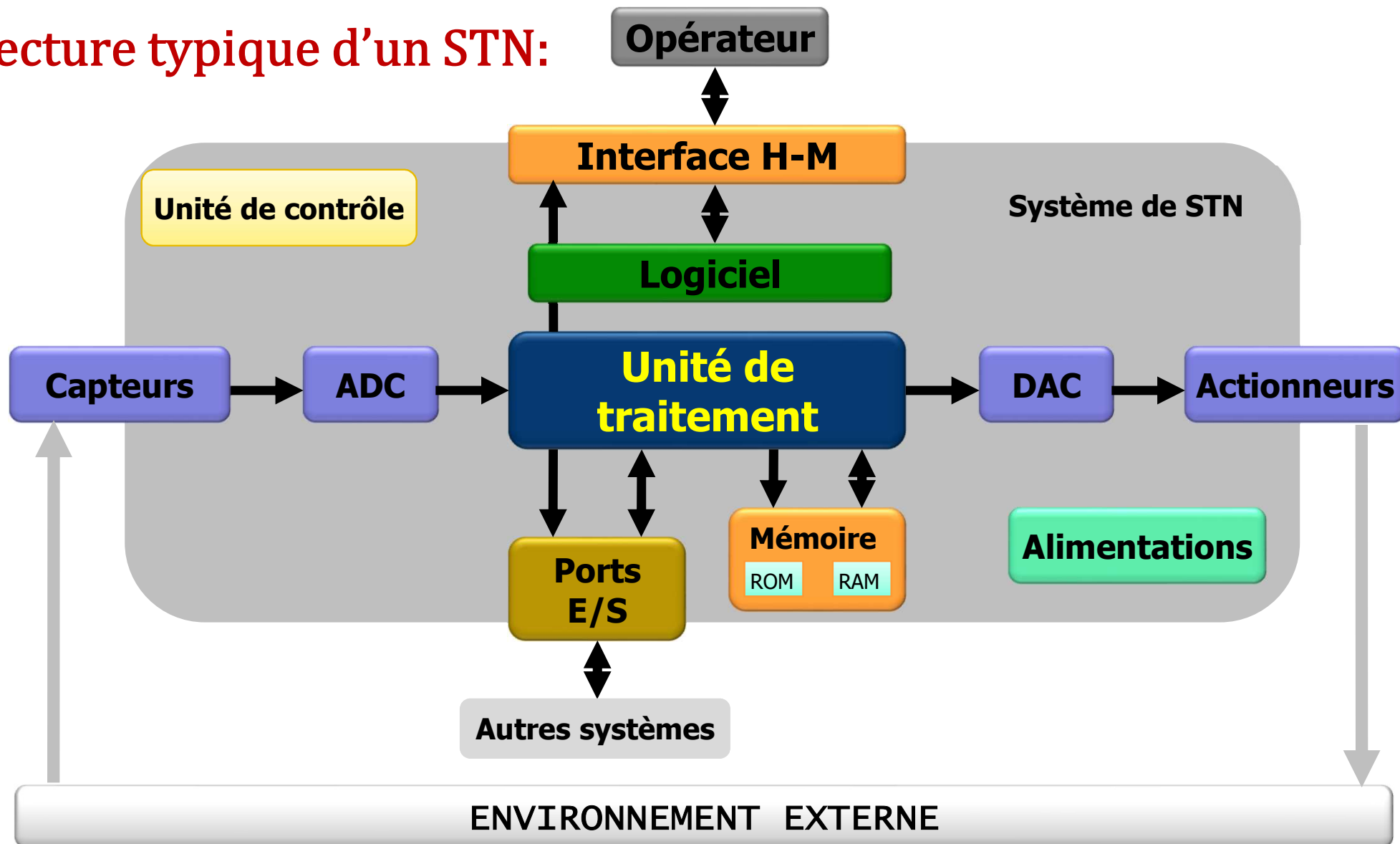
*Courbe d'évolution
du coût par rapport à
la complexité:*



A retenir:

- Si la fonction à réaliser est simple \Rightarrow une logique câblée
- Si le nombre d'unités à réaliser est très important \Rightarrow circuits intégrés dédiés en logique câblée pour les fonctions simples, sinon circuit programmables

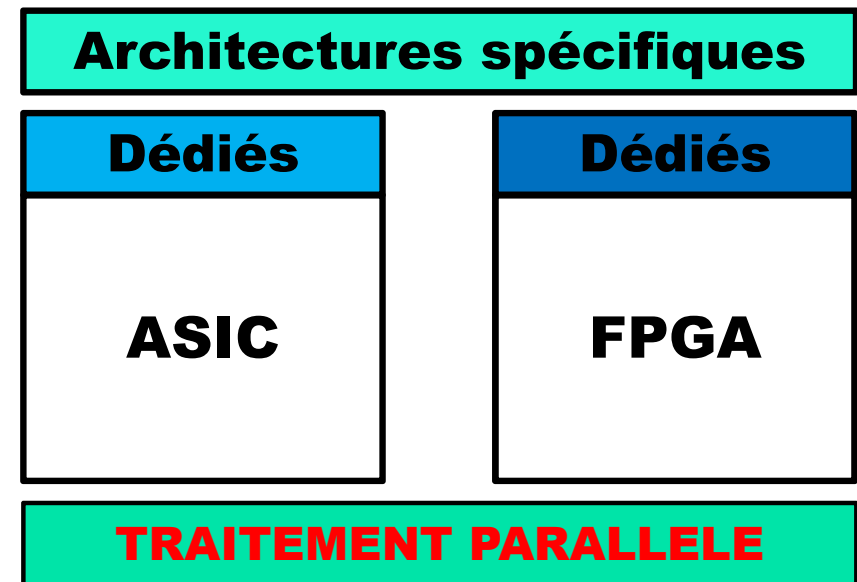
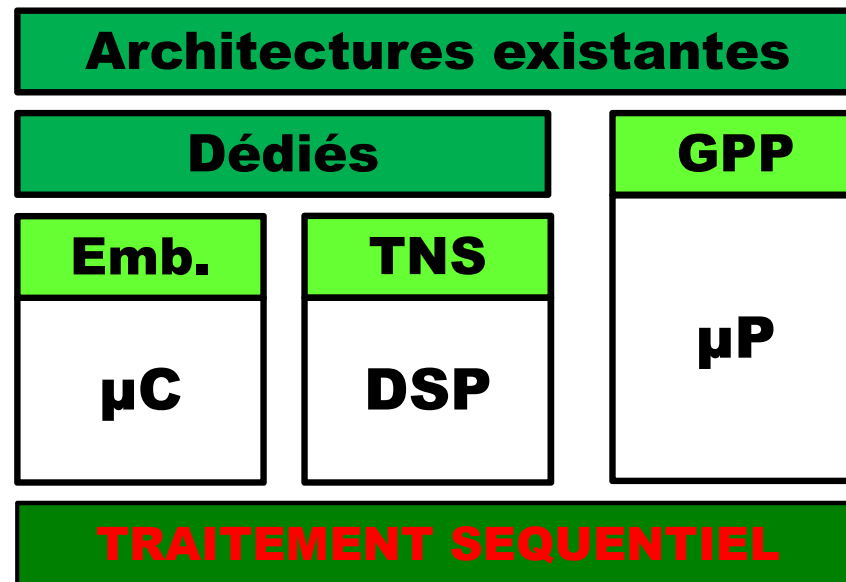
Architecture typique d'un STN:



Architecture typique d'un STN:

Pour concevoir un STN "intelligent", deux solutions principales :

- des architectures existantes (microcontrôleurs)
- des architectures dédiées (FPGA, ASIC)

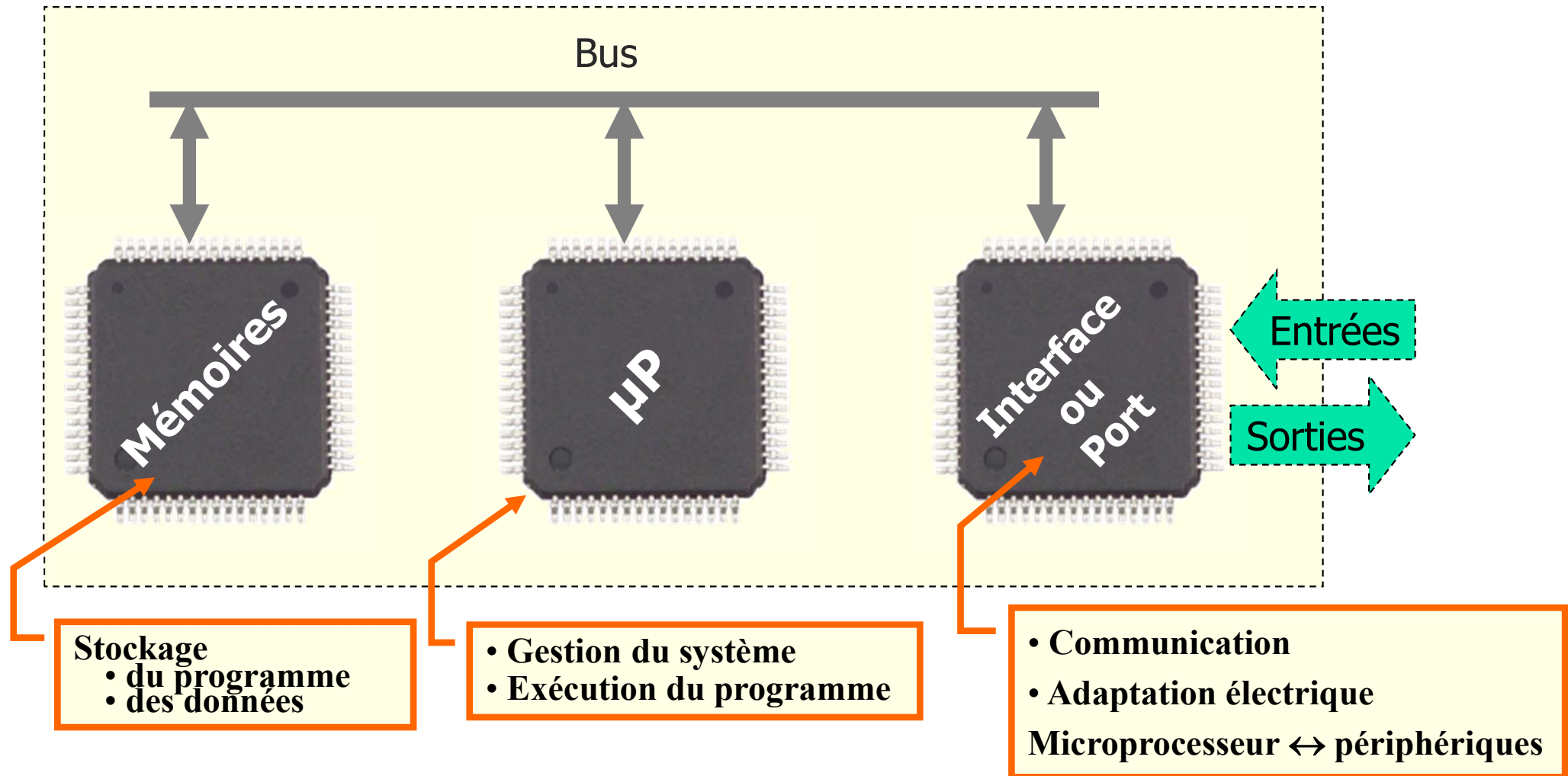




Sommaire:

- Objectifs
- Informatique Industrielle
- Du système câblé au microprogrammé
- **Systèmes à microprocesseur**
 - ✓ Système minimal
 - ✓ Microprocesseur
 - ✓ Mémoire
 - ✓ Interfaces d'entrée/sortie
- Exploitation d'un système à μp
- Du microprocesseur au microcontrôleur

Un système à μp est une combinaison entre un μp et divers circuits annexes.



Le microprocesseur:

- Unité centrale de traitement de données : CPU
- Fournit aux autres éléments du système la synchronisation fonctionnelle nécessaire au traitement des données.

Il assure deux fonctions essentielles:

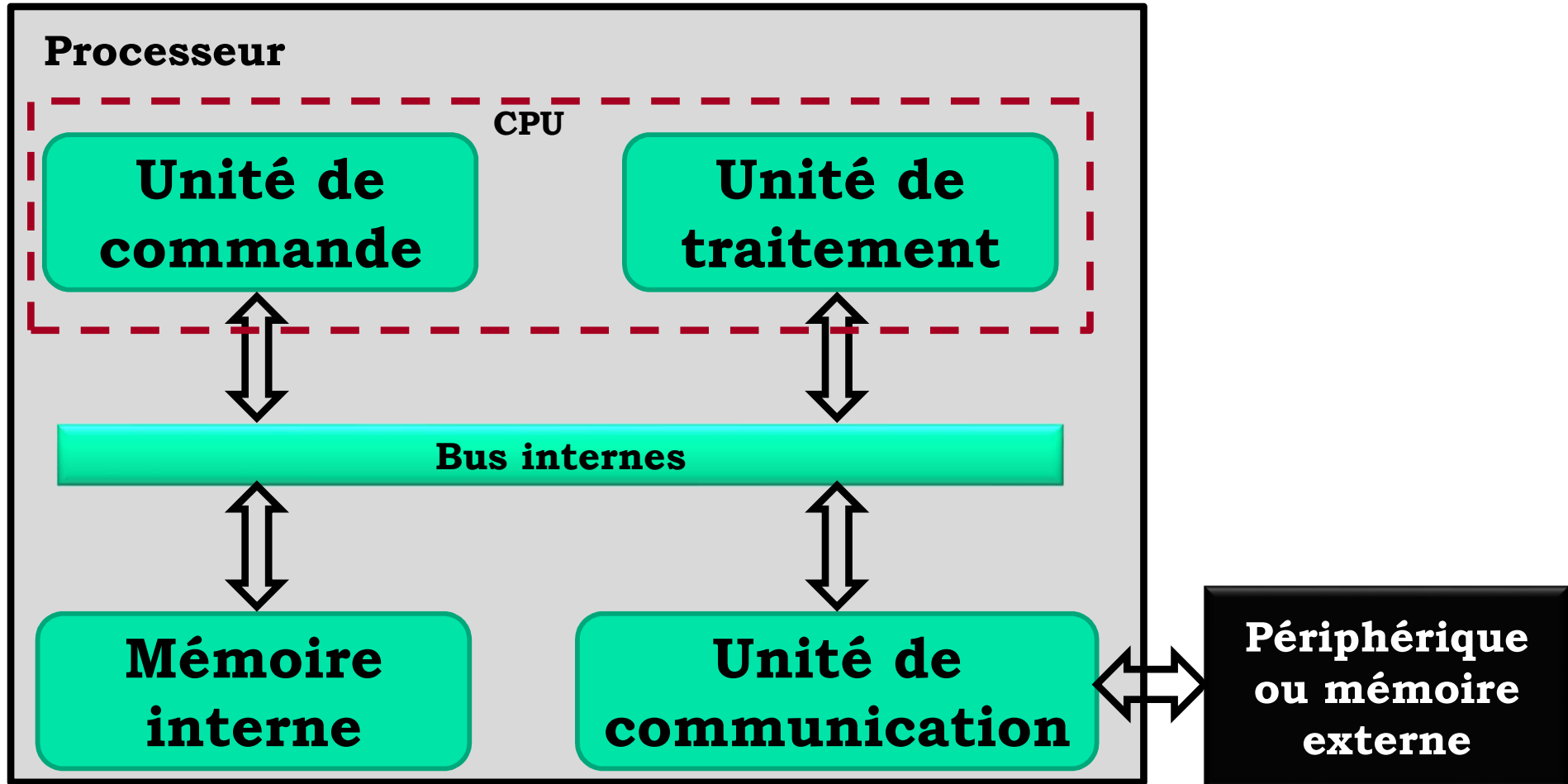
Le traitement des données :

On parle d'unité de traitement. Cette fonction est dédiée à l'U.A.L.
Elle concerne la manipulation des données:
(transfert, opérations arithmétiques, opérations logiques....)

Le contrôle du système:

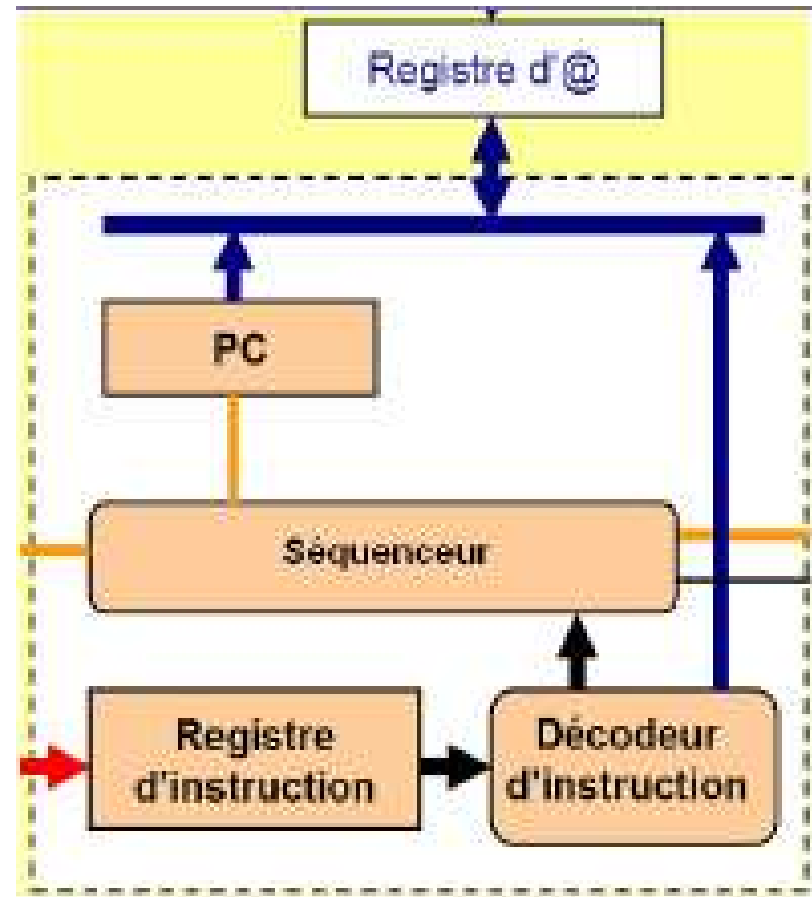
Cette fonction se traduit par des opérations de décodage et d'exécution des ordres exprimés sous forme d'instruction.

Le microprocesseur: Structure interne



Le microprocesseur: unité de commande (1/4)

- ✓ Elle permet de séquencer le déroulement des instructions.
- ✓ Elle effectue la recherche en mémoire de l'instruction, le décodage de l'instruction codée sous forme binaire.
- ✓ Enfin elle pilote l'exécution de l'instruction.

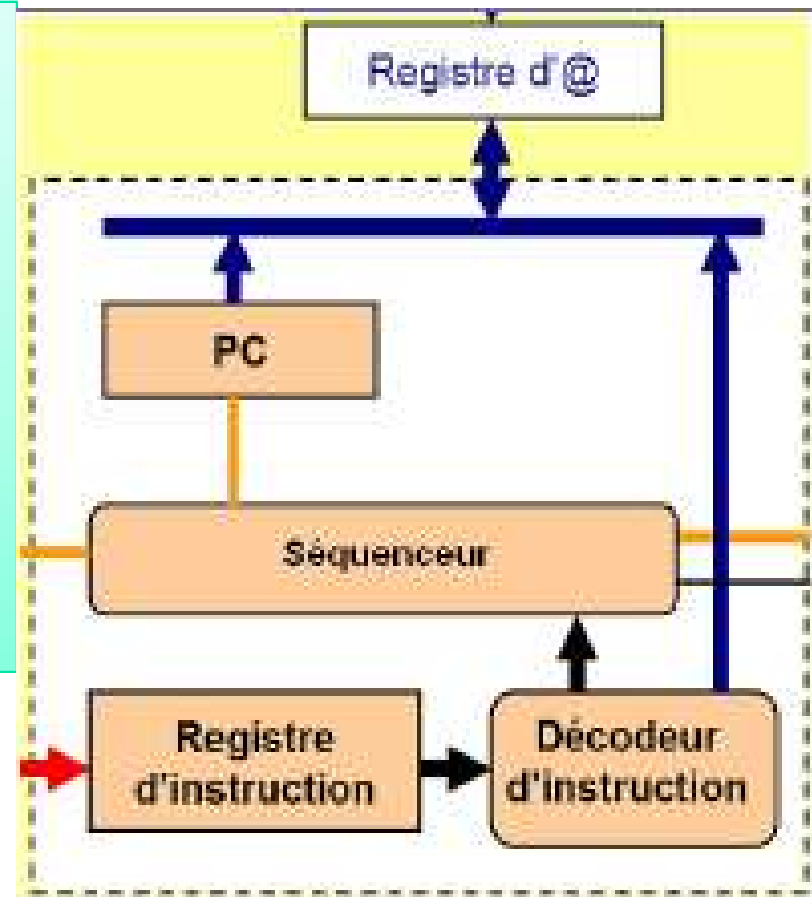


Le microprocesseur: unité de commande (2/4)

➤ Le compteur de programme :

(PC: Program Counter)

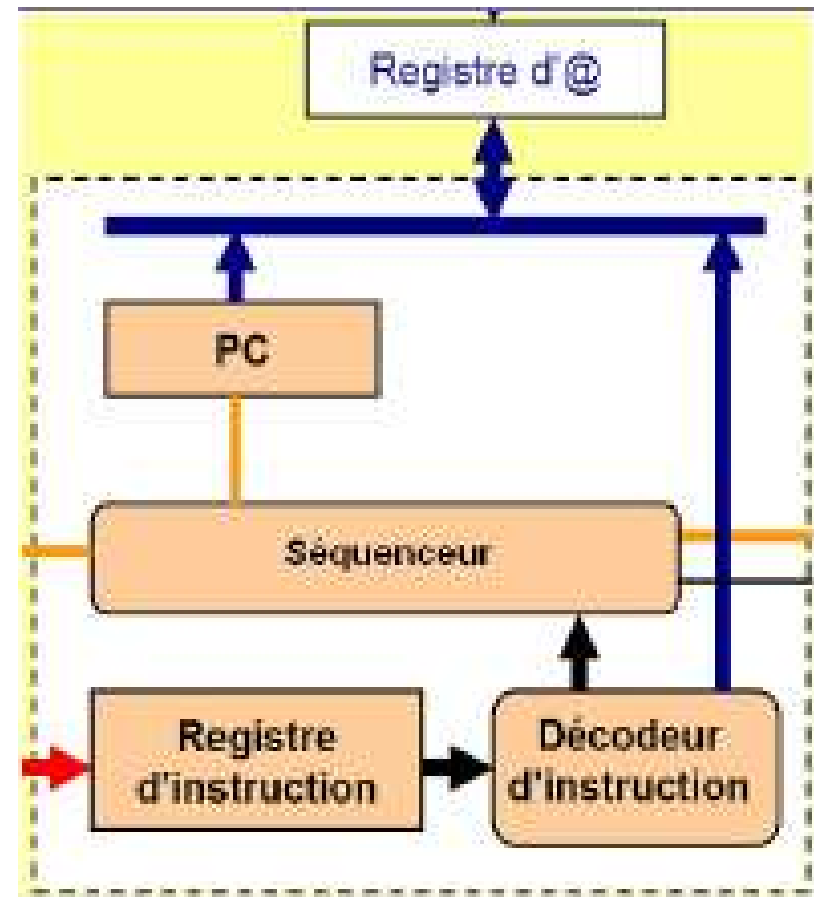
- ✓ appelé aussi Compteur Ordinal (CO)
- ✓ C'est un registre dont le contenu est initialisé avec l'adresse de la première instruction du programme.
- ✓ Il contient toujours l'adresse de la prochaine instruction à exécuter.



Le microprocesseur: unité de commande (3/4)

➤ Le registre d'instruction et le décodeur d'instruction :

Chacune des instructions à exécuter est transféré depuis la mémoire dans le registre instruction puis est décodée par le décodeur d'instruction.

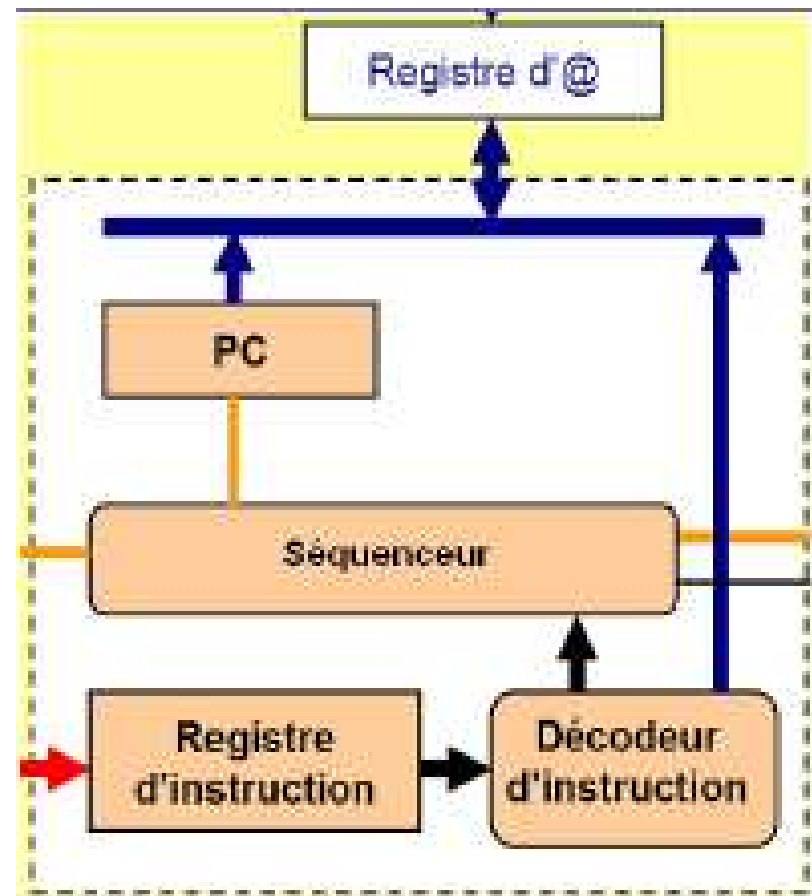


Le microprocesseur: unité de commande (4/4)

➤ Bloc logique de commande :

(ou séquenceur) :

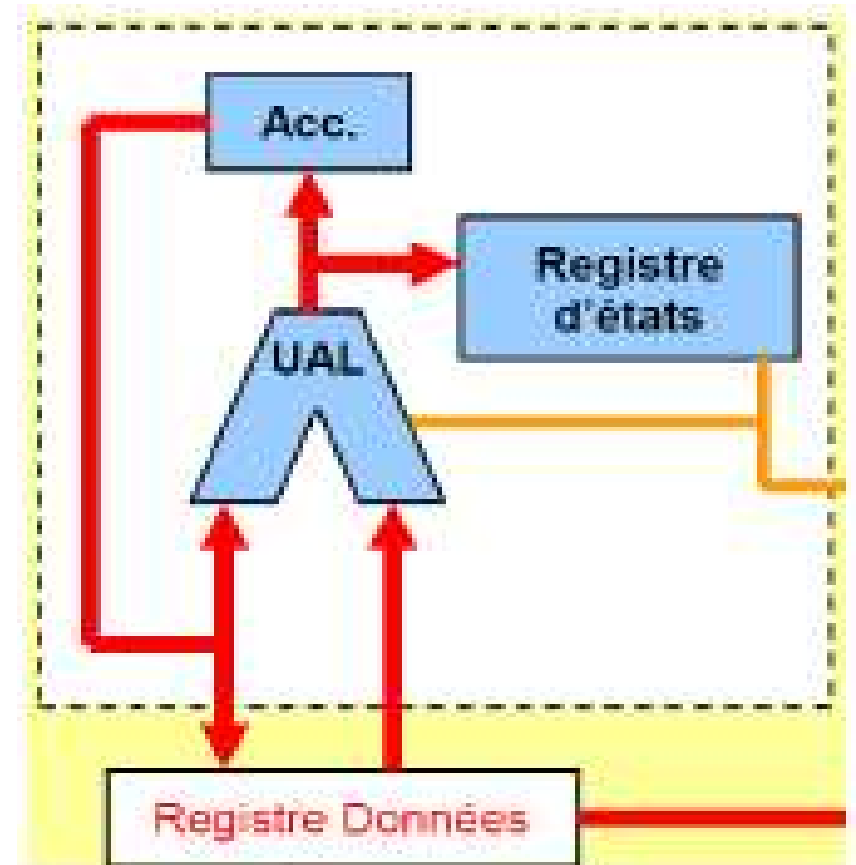
- ✓ Il organise l'exécution des instructions au rythme d'une horloge.
- ✓ Il élabore tous les signaux de synchronisation internes ou externes (bus de commande) du microprocesseur en fonction de l'instruction qu'il a à exécuter.
- ✓ Il s'agit d'un automate réalisé de façon microprogrammée.



Le microprocesseur: unité de traitement

➤ Le registre d'état :

- ✓ 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), Débordement (overflow), Zéro (zero)



Le microprocesseur: unité d'E/S et mémoire

➤ Unité d'entrée-sortie :

Unité qui prend en charge la communication avec la mémoire de l'ordinateur ou la transmission des ordres destinés à piloter ses processeurs spécialisés, permettant au processeur d'accéder aux périphériques de l'ordinateur.

➤ Mémoire cache :

qui permet d'accélérer les traitements, en diminuant les temps d'accès à la mémoire. Ces mémoires tampons sont en effet beaucoup plus rapides que la RAM et ralentissent moins le CPU. Le cache instructions reçoit les prochaines instructions à exécuter, le cache données manipule les données. Parfois, un seul cache unifié est utilisé pour le code et les données.

Plusieurs niveaux de caches peuvent coexister, on les désigne souvent sous les noms de L1, L2 ou L3. Dans les processeurs évolués, des unités spéciales du processeur sont dévolues à la recherche, par des moyens statistiques et/ou prédictifs, des prochains accès en mémoire centrale.

Le microprocesseur: unité d'E/S et mémoire

➤ Unité d'entrée-sortie :

Unité qui prend en charge la communication avec la mémoire de l'ordinateur ou la transmission des ordres destinés à piloter ses processeurs spécialisés, permettant au processeur d'accéder aux périphériques de l'ordinateur.

➤ Mémoire cache :

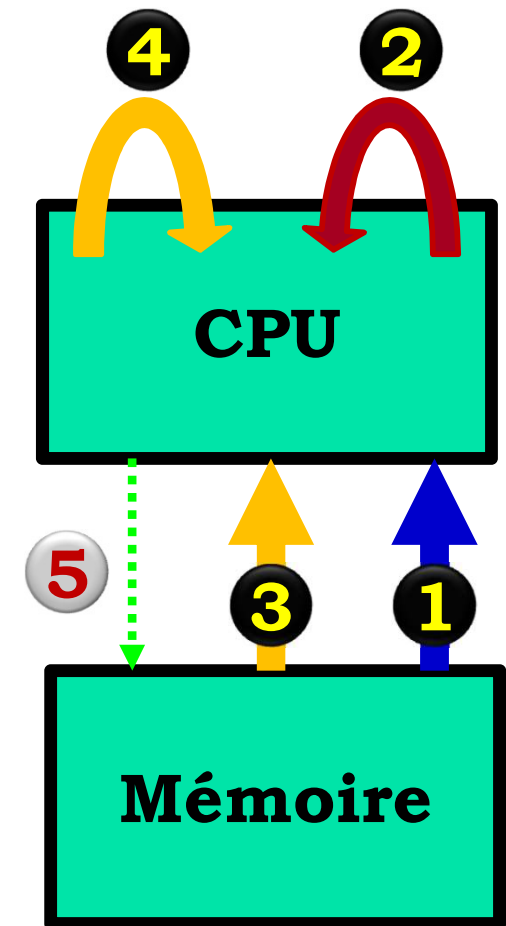
qui permet d'accélérer les traitements, en diminuant les temps d'accès à la mémoire. Ces mémoires tampons sont en effet beaucoup plus rapides que la RAM et ralentissent moins le CPU. Le cache instructions reçoit les prochaines instructions à exécuter, le cache données manipule les données. Parfois, un seul cache unifié est utilisé pour le code et les données.

Plusieurs niveaux de caches peuvent coexister, on les désigne souvent sous les noms de L1, L2 ou L3. Dans les processeurs évolués, des unités spéciales du processeur sont dévolues à la recherche, par des moyens statistiques et/ou prédictifs, des prochains accès en mémoire centrale.

Le microprocesseur: Execution d'une operation

Une operation est realisee en un cycle de base:

1. Aller chercher l'instruction en memoire programme (**Fetch**) ;
2. Realiser le decodage de l'instruction, et des adresses des operandes (**Decode**) ;
3. Lire les operandes en memoire de donnees (**Read**) ;
4. Executer l'operation (**Execute**) ;
5. Ecrire les resultats en memoire (**Write back**).



Le microprocesseur: Jeu d'instruction

(1/2)

Ensemble des commandes primitives que le programmeur peut utiliser sur une machine.

- Une instruction est découpée en champs :
 - ✓ Un code opération (**Opcode**)
 - ✓ Des informations complémentaires sur l'emplacement des données sources et de la destination (**Opde**)

- Format général :

Instruction:	Taille (<i>n,m</i>)
<i>Code opération</i>	<i>Désignation des opérandes</i>

Le microprocesseur: Jeu d'instruction

(2/2)

- Objectifs différents selon les classes d'applications
 - ✓ Vitesse maximale (PC, serveurs)
 - ✓ Taille de code minimale (**embarqué**)
 - ✓ Consommation
 - essentiel pour **embarqué**
 - important pour **toute application**

- Taille des instructions
 - ✓ Fixe
 - ✓ Variable

- Modèles d'exécution :
Architecture processeur (RISC-CISC)

Le microprocesseur: Modèle d'exécution

(1/3)

CISC :

- ✓ Plusieurs modes d'adressages mémoires possible;
- ✓ Grand nombre d'instructions
- ✓ Codage des instructions variables : le nombre d'octets nécessaire pour le codage des instructions est variable
- ✓ Pour une tache donnée, une machine CISC exécute ainsi un petit nombre d'instructions mais chacune nécessite un plus grand nombre de cycles d'horloge.

Le microprocesseur: Modèle d'exécution

(2/3)

RISC :

- ✓ Peu d'instructions (simples) :une diminution de la complexité de la partie unité de commande
- ✓ Peu de modes d'adressage
- ✓ Taille fixe des instructions
- ✓ Chacune de ces instructions s'exécutent ainsi en un cycle d'horloge
- ✓ Au moins 16 registres + qlqs registres flottants
- ✓ Seul deux instructions accèdent à la mémoire load/store;

Le microprocesseur: Modèle d'exécution

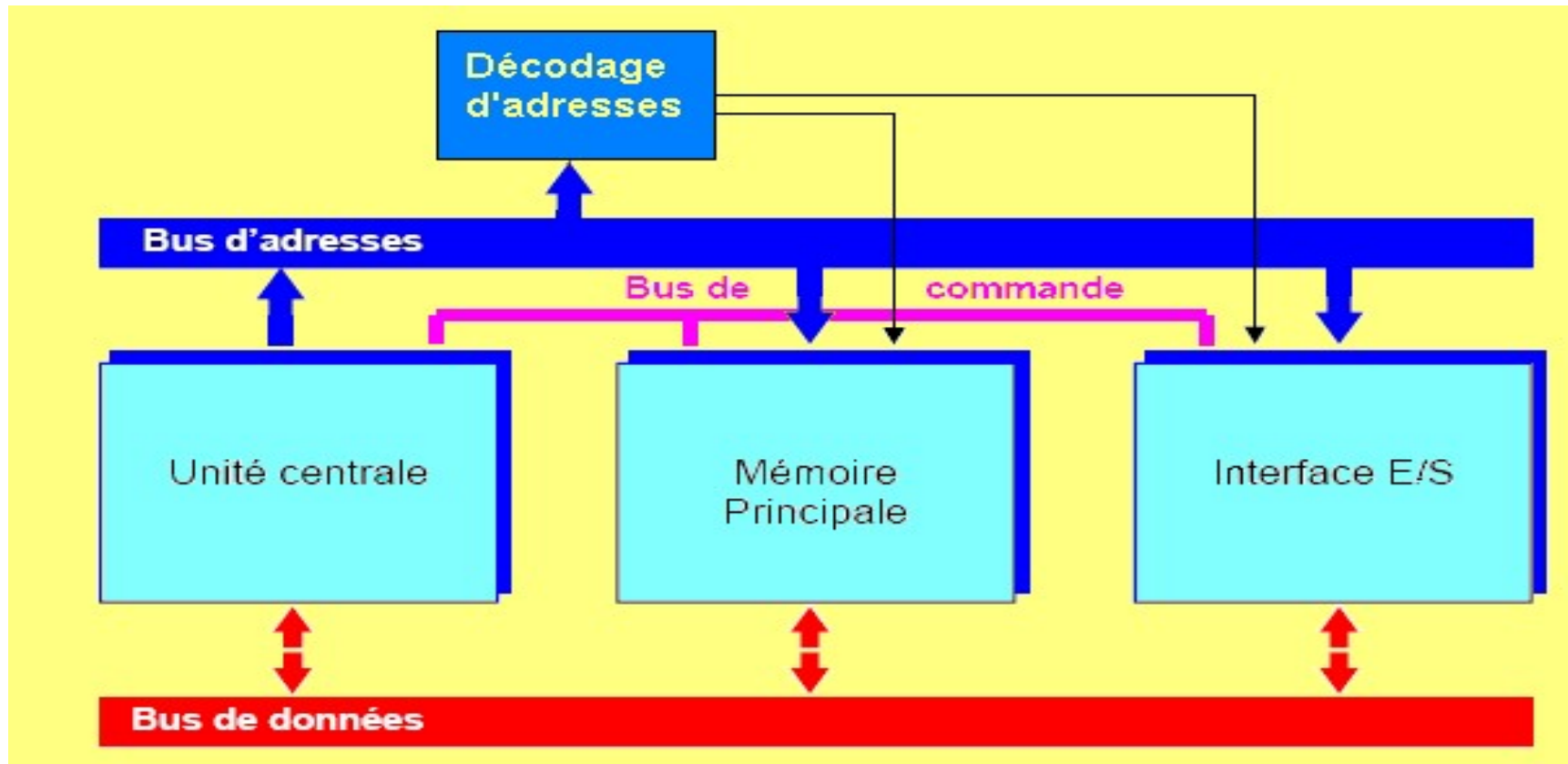
(3/3)

Architecture RISC	Architecture CISC
Instructions simples ne prenant qu'un seul cycle	Instructions complexes prenant plusieurs cycles
Instructions au format fixe	Instructions au format variable
Décodeur simple (câblé)	Décodeur complexe (microcode)
Beaucoup de registres	Peu de registres
Seules les instructions LOAD et STORE ont accès à la mémoire	Toutes les instructions sont susceptibles d'accéder à la Mémoire
Peu de modes d'adressage	Beaucoup de modes d'adressage
Compilateur complexe	Compilateur simple

Le microprocesseur: Les bus

(1/2)

Ils relient électriquement le micro-processeur et les périphériques (mémoires et interfaces E/S).



Le microprocesseur: Les bus

(2/2)

Le bus de données est bidirectionnel:

Il assure le transfert des informations entre le microprocesseur et son environnement, et inversement. Son nombre de lignes est égal au format des mots de données du microprocesseur.

Le bus d'adresses est unidirectionnel.

Il permet la sélection des informations à traiter dans un espace mémoire (ou espace adressable) qui peut avoir 2^n emplacements, avec n = nombre de conducteurs du bus d'adresses.

Le bus de commande (ou bus de contrôle):

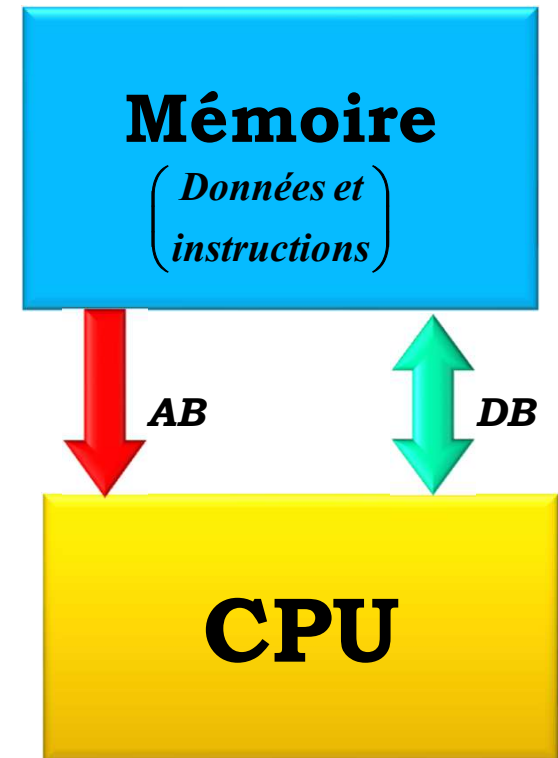
Il est constitué par quelques conducteurs qui assurent la synchronisation des flux d'informations sur les bus de données et d'adresses.

Le microprocesseur: Deux architectures (1/3)

➤ L'architecture Von Neumann :

- ✓ Un seul chemin d'accès à la mémoire
 - Un bus de données (programme et données),
 - Un bus d'adresse (programme et données)
- ✓ Architecture des processeurs d'usage général
- ✓ Goulot d'étranglement pour l'accès à la mémoire

☞ **Bande passante limitée**

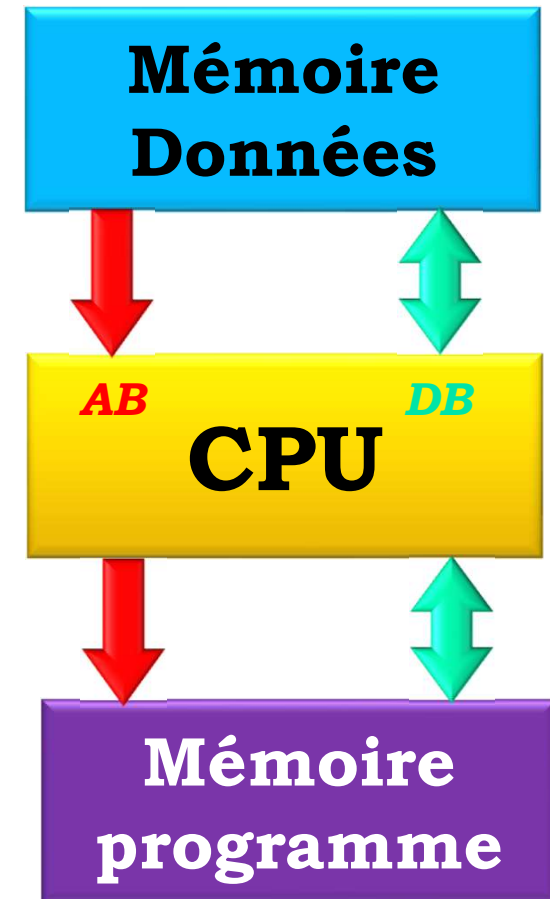


Le microprocesseur: Deux architectures (2/3)

➤ L'architecture Harvard :

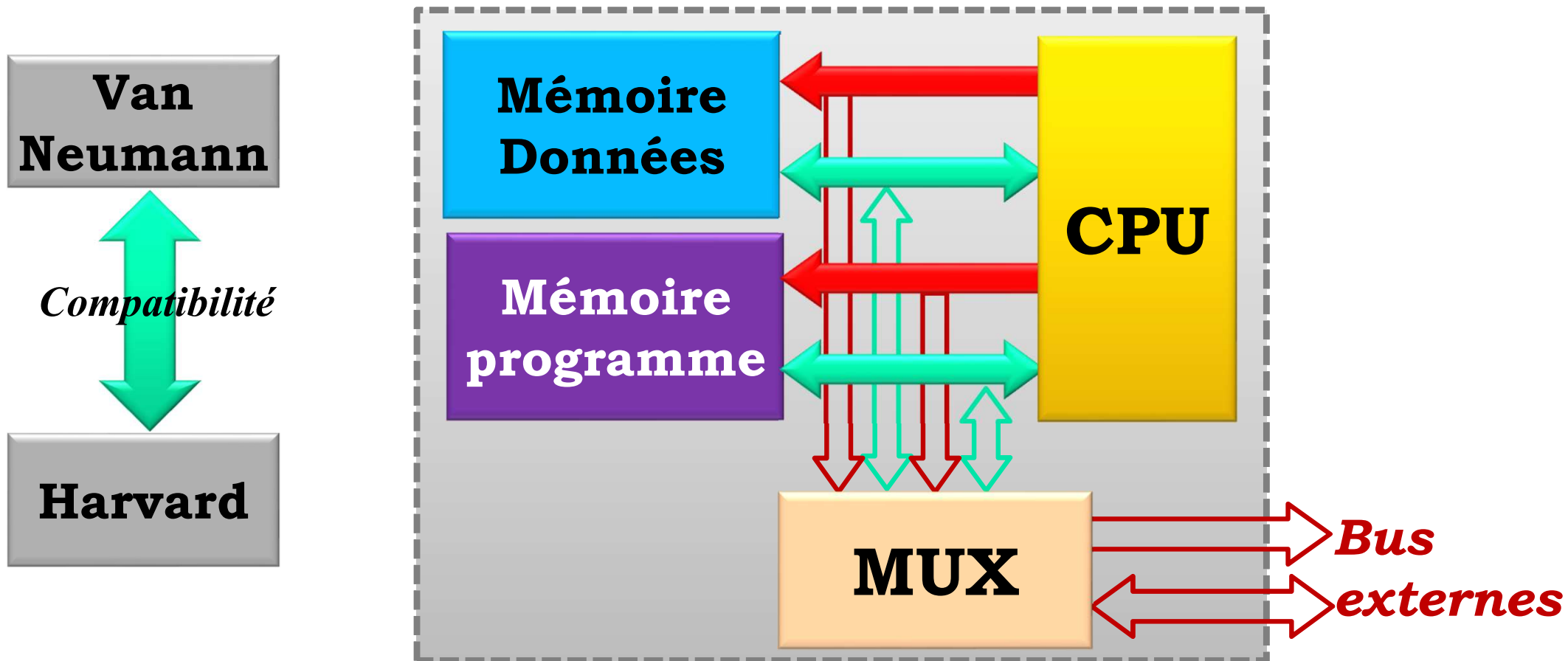
- ✓ Une Séparation des mémoires programme et données
 - Un bus de données programme,
 - Un bus de données pour les données,
 - Un bus d'adresse programme,
 - Un bus d'adresse pour les données.
- ✓ Meilleure utilisation du CPU :
 - Chargement du programme et des données en parallèle

☞ **Exécution des opérations en parallèle**



Le microprocesseur: Deux architectures (3/3)

➤ L'architecture Harvard modifiée :



Le microprocesseur: Caracteristiques

Le format des données
= Nombre de bit du bus de donnée

- 8bits
- 16bits
- 32bits
- 64 bits

La puissance de traitement

S'exprime en MIPS



La taille de l'espace adressable
= Nombre de bit du bus d'adresse

- 16 bits = 65.536 adresses
- 32 bits = 4.294.967.296 adresses



Vitesse
d'exécution

Le jeu d'instructions

La puissance consommée

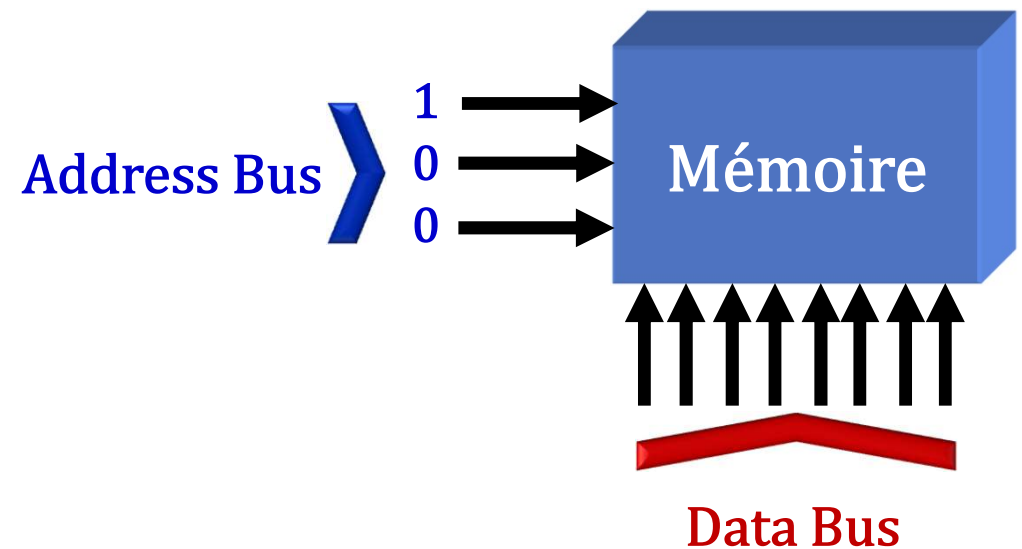
- Pd (w) fonction de la vitesse
- Pd (w) fonction de la tension

La mémoire:

(1/12)

Définition :

Une mémoire est un circuit capable de stocker des informations, de les conserver et de les restituer à temps voulu.

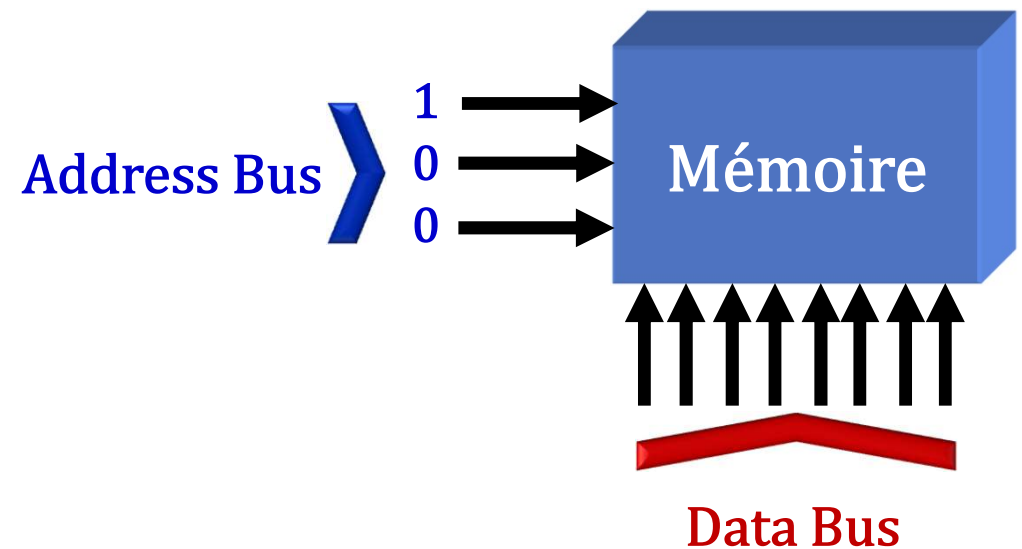
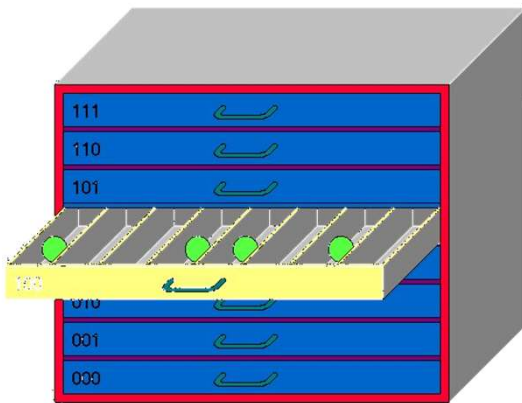


La mémoire:

(2/12)

Définition :

Une mémoire est un circuit capable de stocker des informations, de les conserver et de les restituer à temps voulu.



La mémoire: Classification

(3/12)

Selon la volatilité :

Morte

- ROM
- PROM
- EPROM
- EEPROM
- Flash

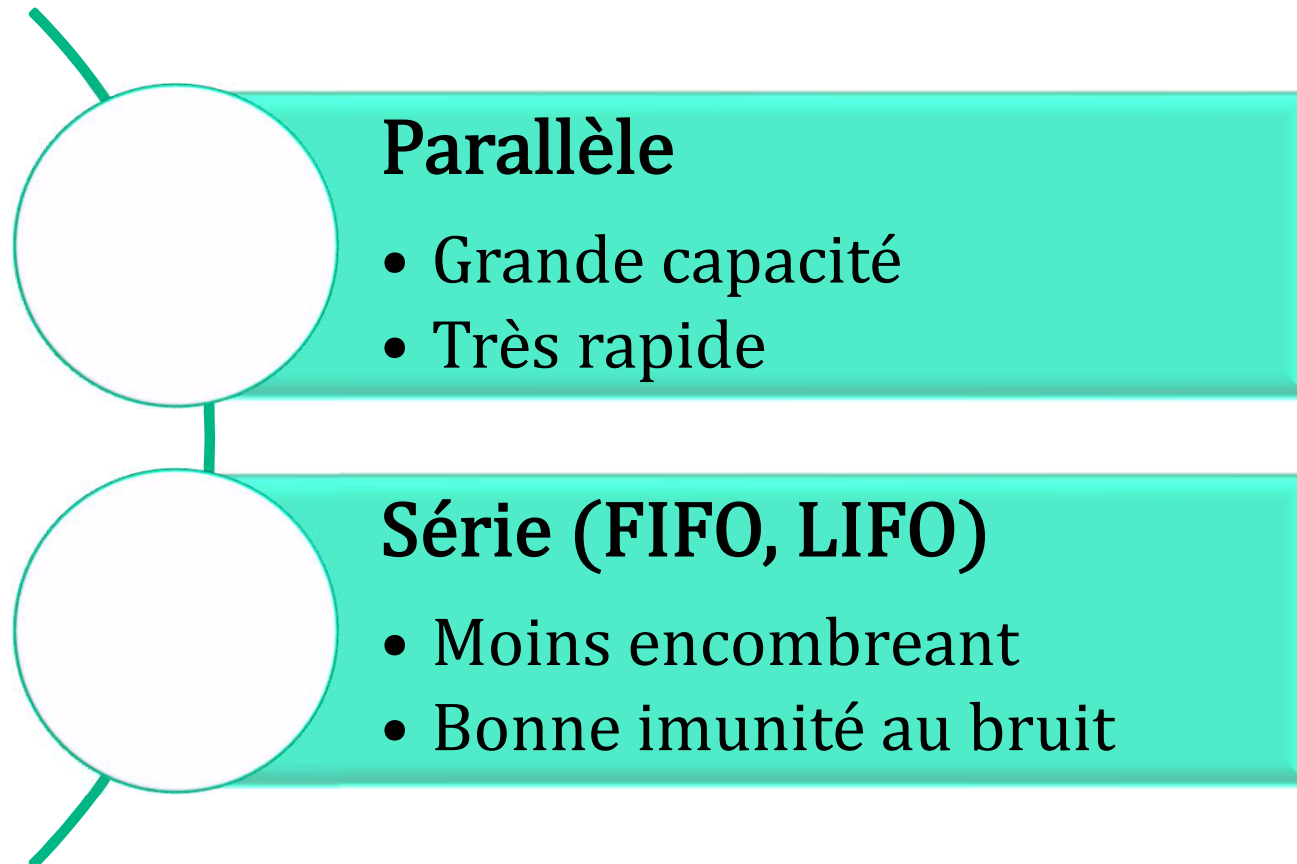
**Vive
(RAM)**

- Statique
(plus rapide)
- Dynamique
(moins encombrants,
moins chères)

La mémoire: Classification

(4/12)

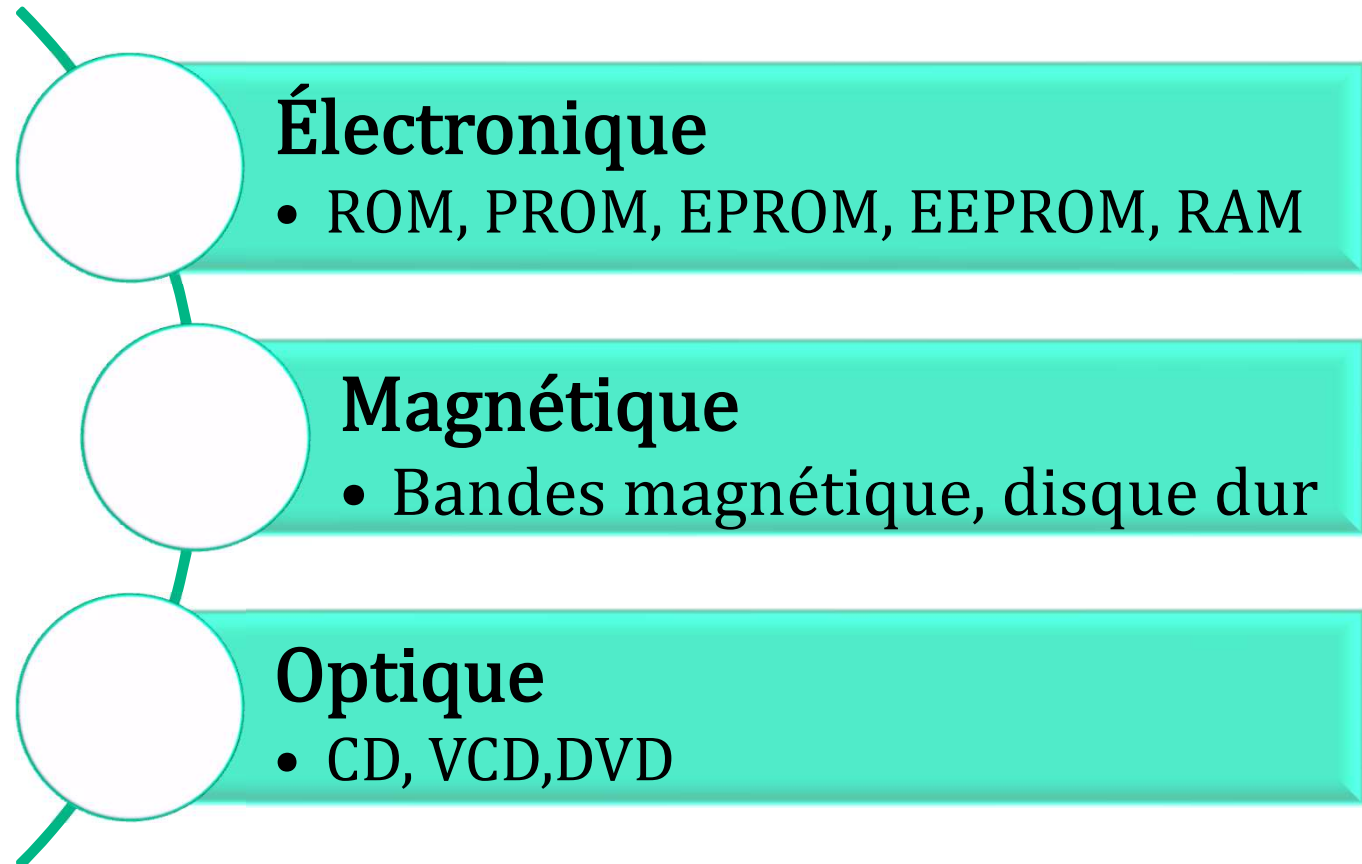
Selon le mode d'accès :



La mémoire: Classification

(5/12)

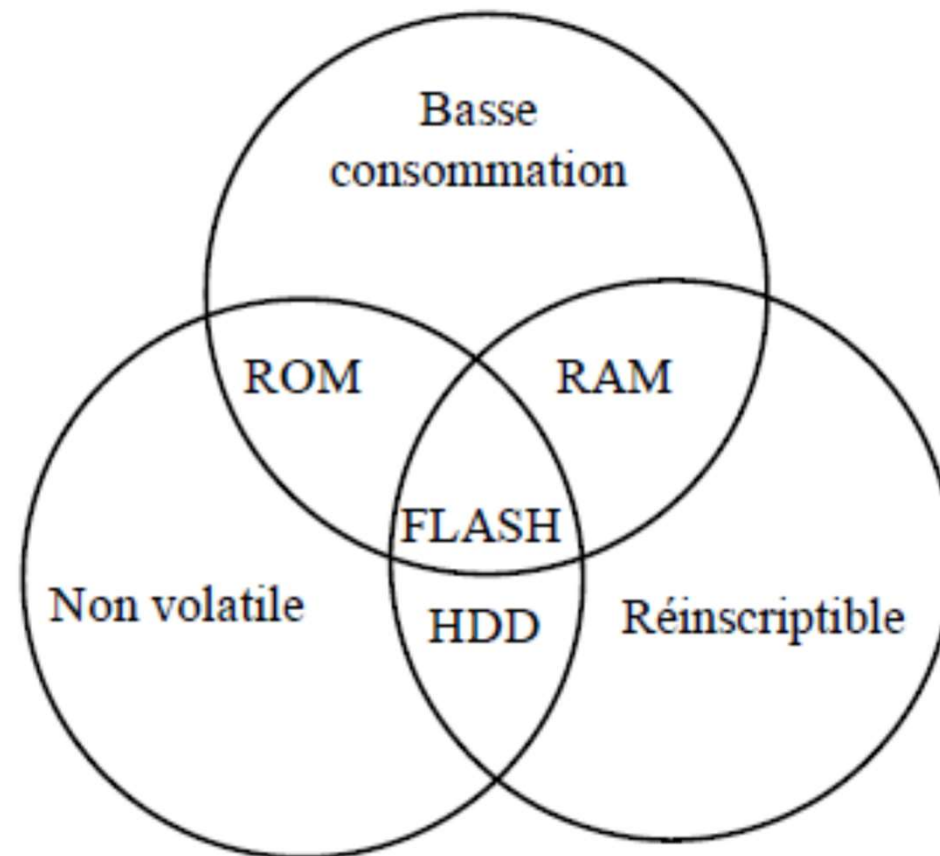
Selon la nature de l'information :



La mémoire: Classification

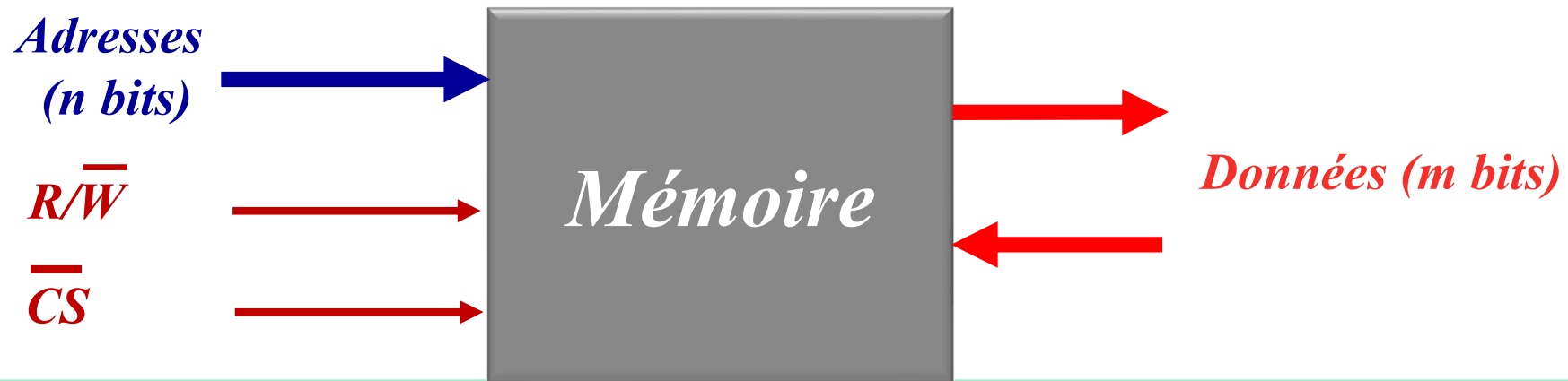
(6/12)

Comparaison :



La mémoire: Analyse d'un boîtier

(7/12)



- Des connexions de données (D_0 à D_{m-1}) : le nombre de ces fils correspond au nombre de bits de chaque mémoire.
- Des connexions d'adresses ($A_0 \dots A_{n-1}$): le nombre des ces fils correspond au nombre total des adresses de la mémoire.
- Des connexions de contrôle, permettant de commander une lecture ou une écriture d'une donnée de la mémoire.
- Une ou plusieurs entrée de sélection (CS), permettant de sélectionner un boîtier parmi plusieurs.

La mémoire: Analyse d'un boîtier

(8/12)

Caractéristiques:

- ✓ **La capacité** : c'est le nombre total de bits que contient la mémoire. Elle s'exprime aussi souvent en octet.
- ✓ **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.
- ✓ **Le débit** : c'est le nombre maximum d'informations lues ou écrites par seconde.
- ✓ **Volatilité** : L'information stockée est volatile si elle risque d'être altérée par un défaut d'alimentation électrique et non volatile dans le cas contraire.

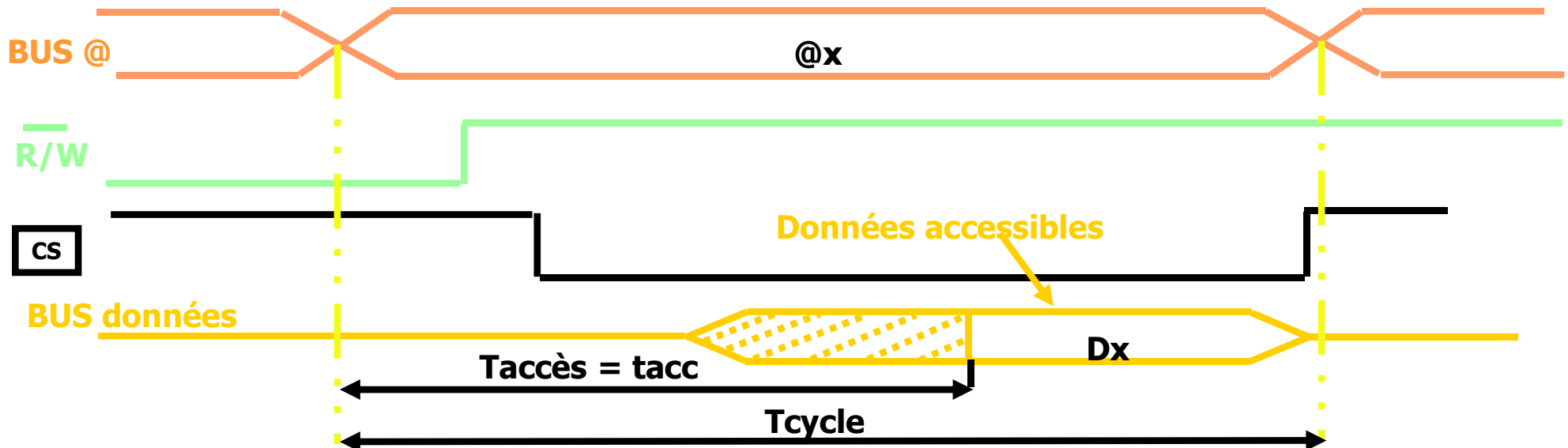
La mémoire: Analyse d'un boîtier

(9/12)

Accès à la mémoire:

Pour accéder à ce boîtier, il faut :

- Sélectionner le boîtier
- Présenter l'adresse du mot sur les fils d'adresse ;
- Commander l'opération désirée (lecture ou écriture) ;
- L'information pourra alors transiter par l'intermédiaire des fils de données

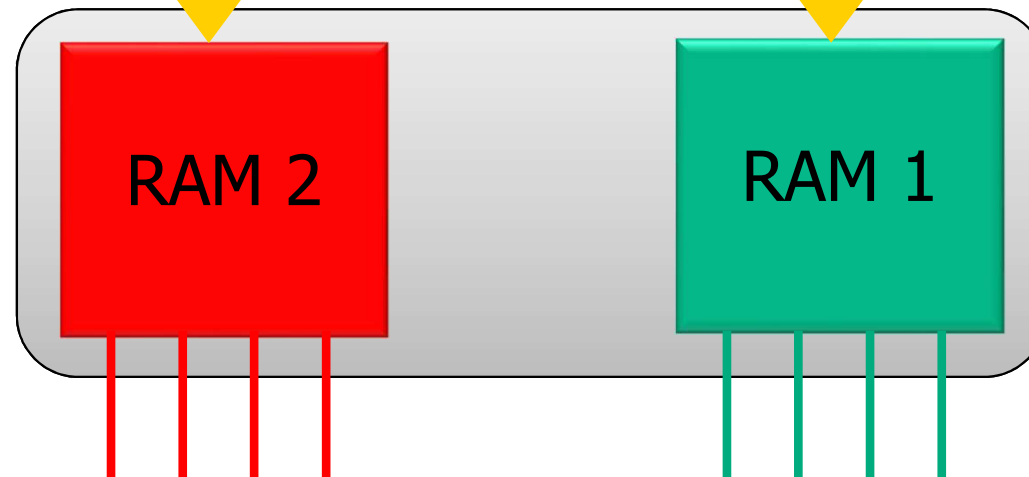


La mémoire: Analyse d'un boîtier

(10/12)

Extension du bus de données:

Bus d'adresse



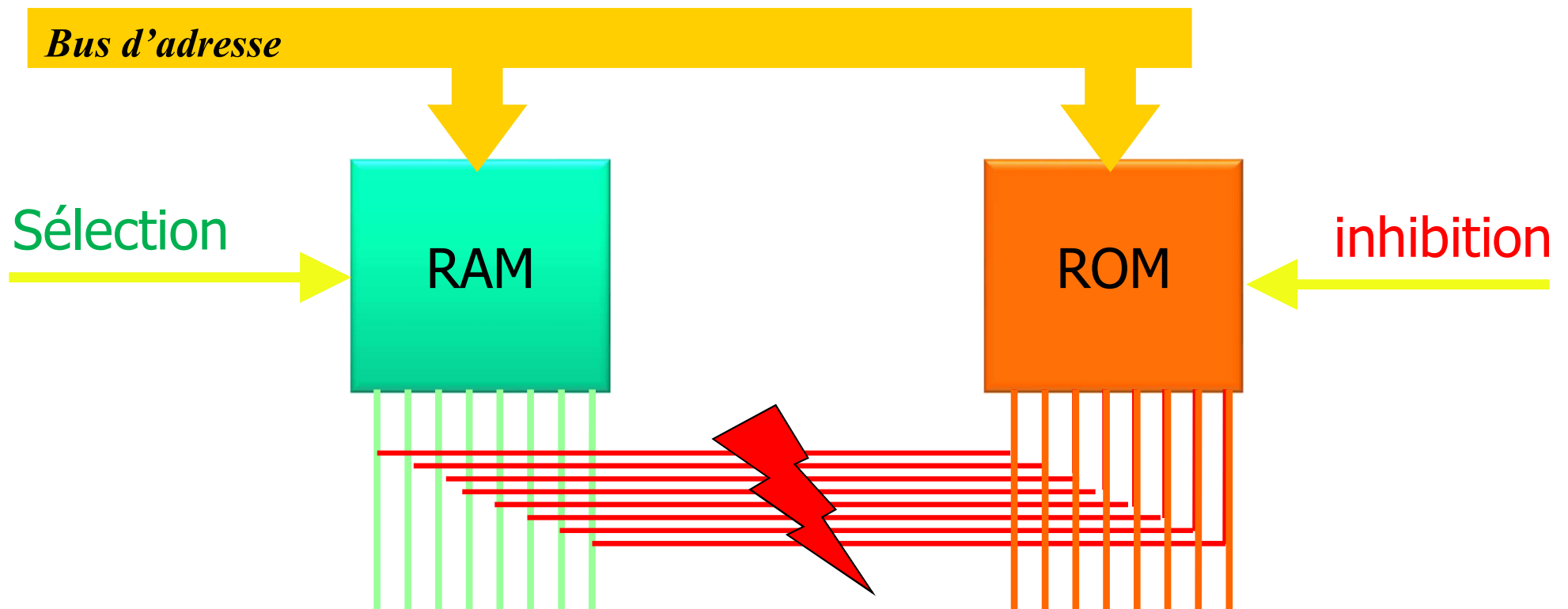
Bus de données réunis → 8 bits

Mot de un octet : 4 bits de poids faible en Ram 1 et 4 de poids fort en RAM 2.

La mémoire: Analyse d'un boîtier

(11/12)

Gestion de l'espace adressable:



Bus de données mis en parallèles
→ 1 seul bus de données 8 bits

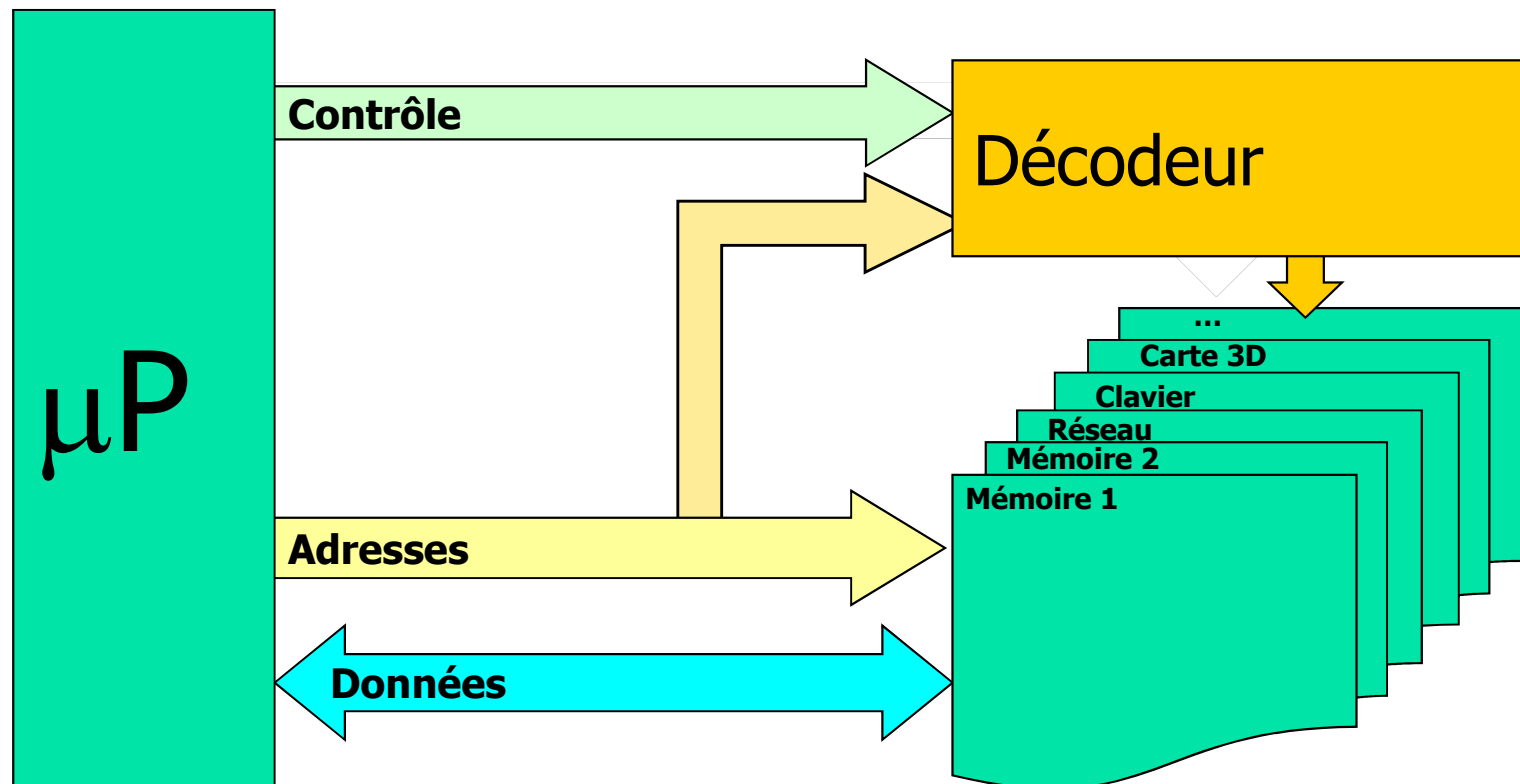
La mémoire: Décodage d'adresse

(12/12)

Objectif:

Distribuer Espace d'adressage entre plusieurs Circuits.

Exemple:



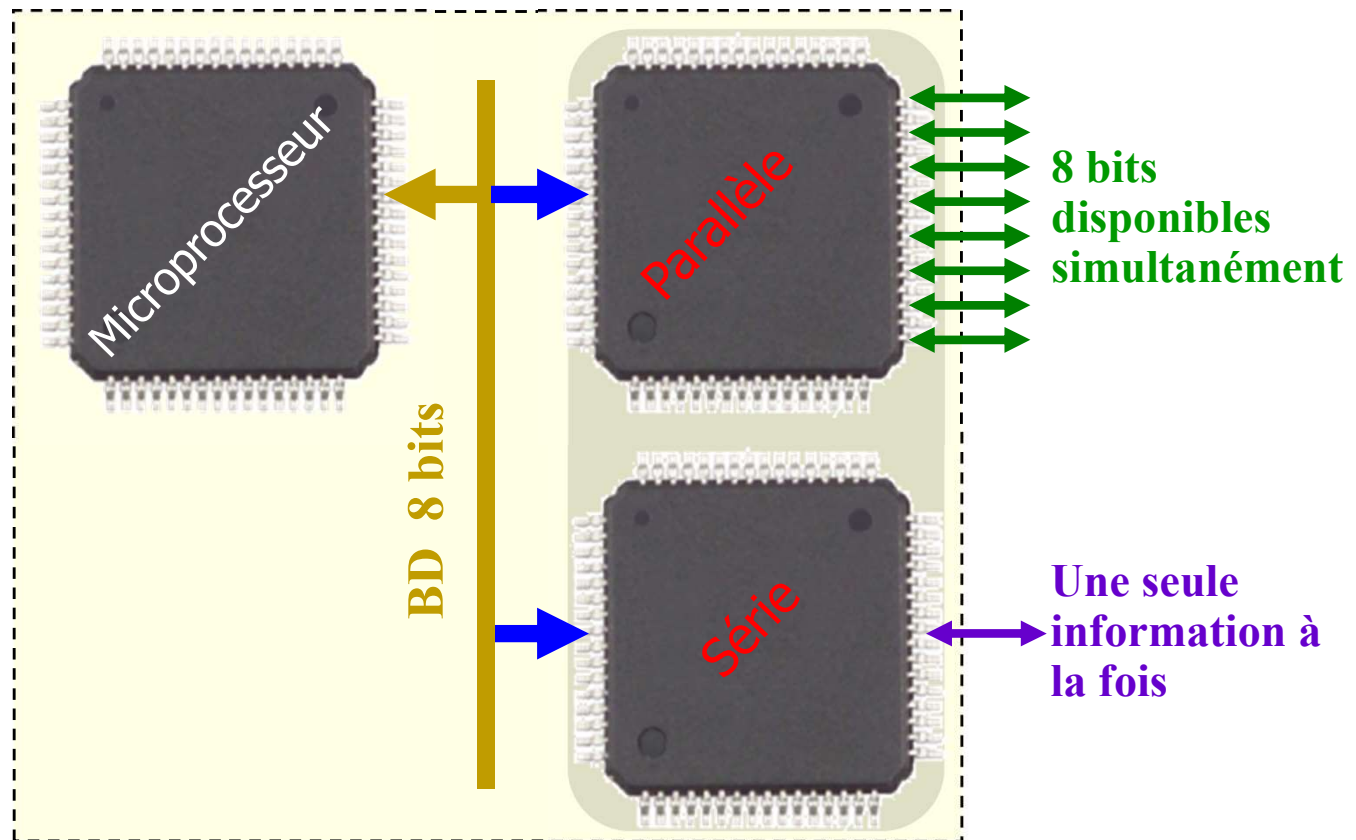
Les interfaces d'entrée-sortie:

(1/6)

Interfaçage:

Une interface est un organe nécessaire entre le μp et le périphérique et ce pour assurer une compatibilité :

- ✓ au niveau du type de transmission (série ou parallèle) ;
- ✓ au niveau du code ;
- ✓ au niveau de la vitesse de transmission ;
- ✓ Au niveau logique (adaptation de tension).



Les interfaces d'entrée-sortie:

(2/6)

INTERFACE PARALLÈLE.

Ce type d'interface, répartie sur plusieurs ports (maximum 8 bits), permet de prendre en compte des états logiques appliqués en entrée (état de capteurs) ou de générer des signaux binaires en sortie (commande d'actionneurs).

Les broches de ces ports peuvent donc être configurées en entrée ou en sortie.

On trouve généralement :

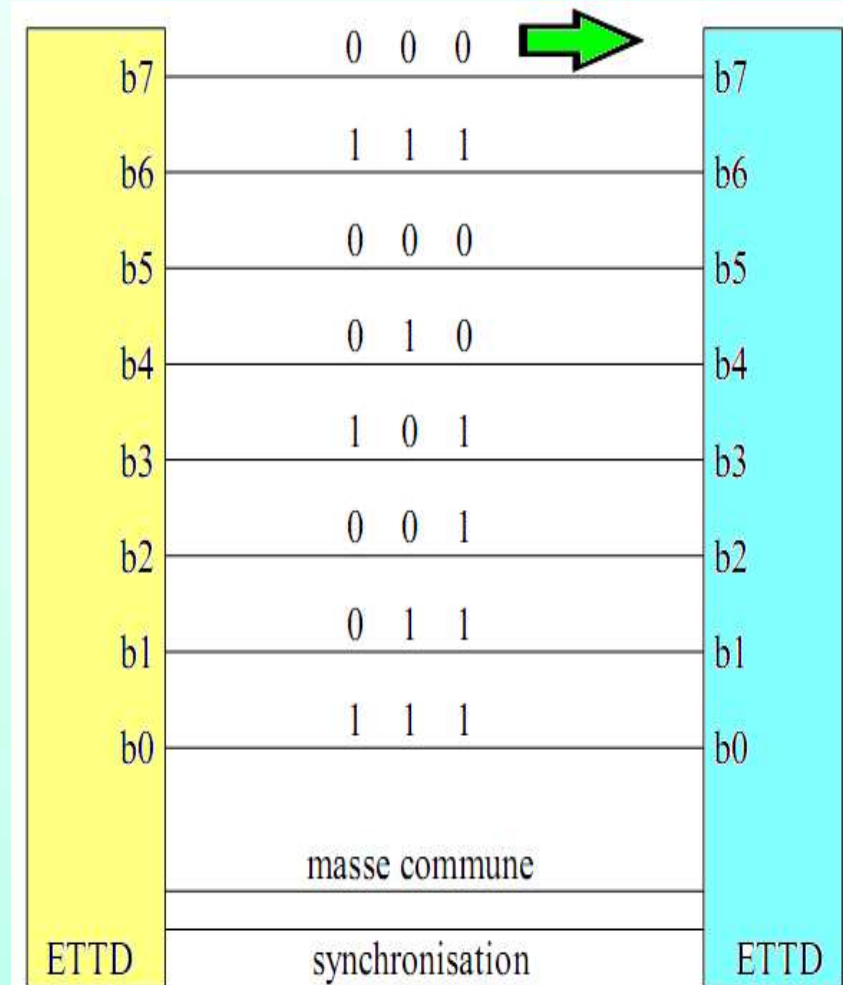
- Un registre de direction pour une configuration en entrée ou en sortie,
- Un registre de donnée recopiant les états logiques de chaque broche de port,
- Un registre d'option permettant plusieurs configurations en entrée ou en sortie.

Les interfaces d'entrée-sortie:

(3/6)

INTERFACE PARALLÈLE.

- Caractérisée par un transfert simultané de tous les bits d'un même mot.
- Nécessite autant de conducteurs qu'il y a de bits à transmettre et un conducteur commun (liaison asymétrique) ou autant de paires que de fils si la masse n'est pas commune (liaison symétrique).
- Un conducteur supplémentaire peut être utilisé pour transmettre un signal qui assurera la synchronisation entre les intervalles d'émission et ceux de réception
- Autorise une grande vitesse de transmission (débit).
- Un coût élevé (nombre de conducteurs)
- Une distance franchissable limitée réservent la transmission parallèle aux liaisons de processeur à processeur



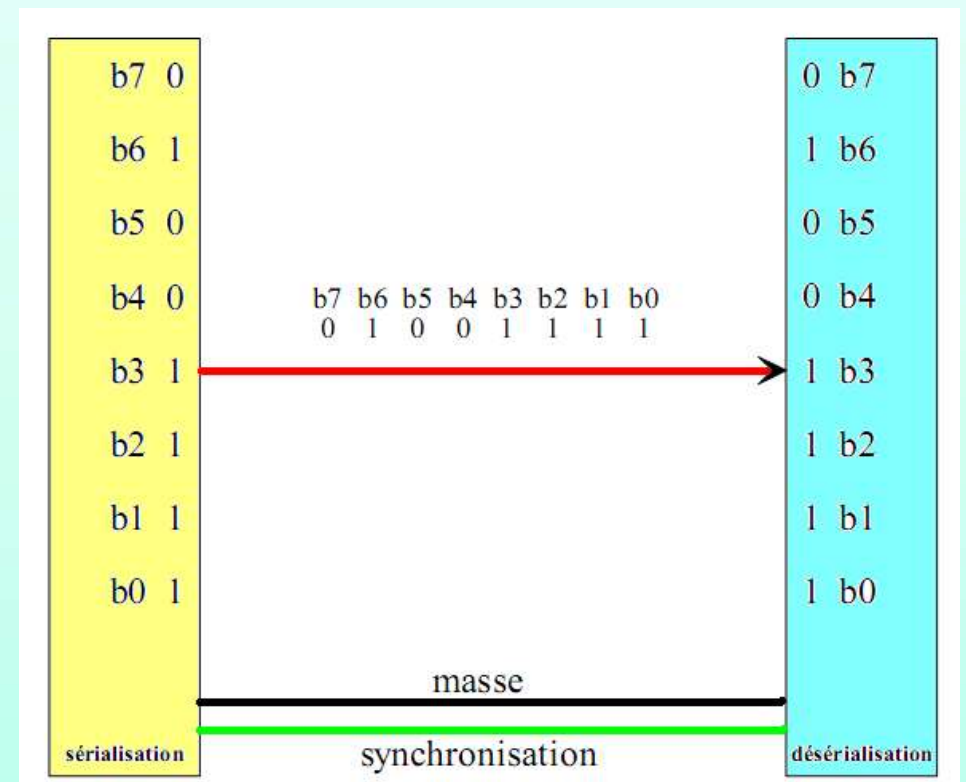
Les interfaces d'entrée-sortie:

(4/6)

INTERFACE SÉRIE.

Les données envoyées ou reçues se présentent sous la forme d'une succession temporelle (sur un seul bit) de valeurs binaires images d'un mot.

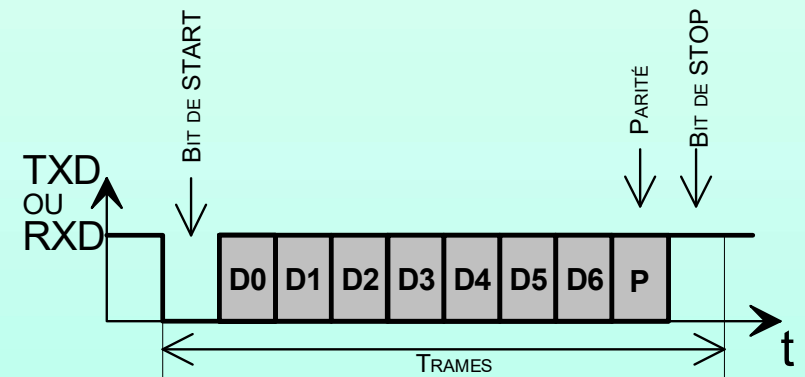
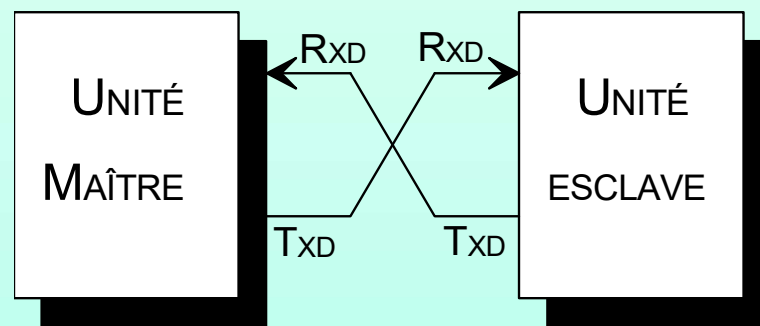
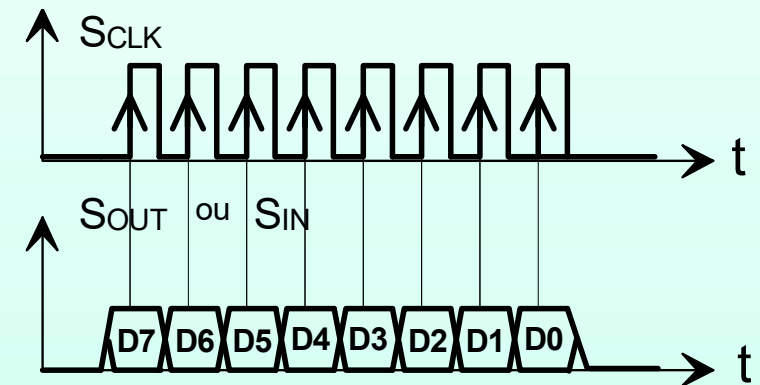
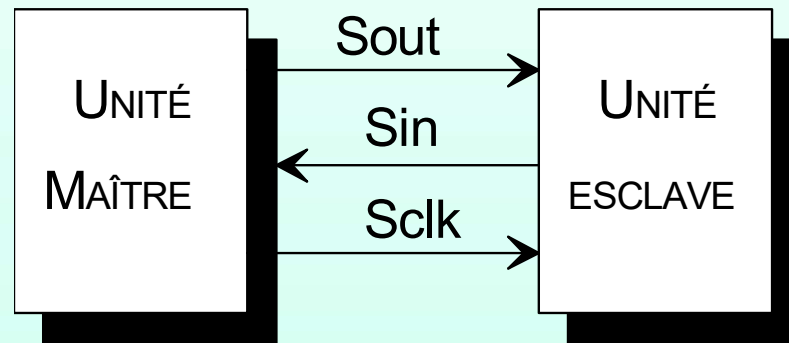
- Nécessite une interface de conversion pour sérialiser les bits à l'émission (conversion parallèle/série) et les désérialiser à la réception (conversion série/parallèle).
- Ne nécessite, pour la transmission des données, que deux conducteurs, d'un coût moins important, elle est utilisée pour les transmissions sur des distances importantes.
- Les vitesses de transmission et de réception doivent être identiques (bits/s ou bauds).
- *Modes simplex, half-duplex et full-duplex*



Les interfaces d'entrée-sortie:

(5/6)

INTERFACE SÉRIE.



Les interfaces d'entrée-sortie: Adressage (6/6)

Pour adresser un périphérique il faut :

- ✓ sélectionner le boîtier d'interface (CS) ;
- ✓ sélectionner l'un des deux ou trois tampons disponibles par boîtier (port).

Deux solutions sont offertes:

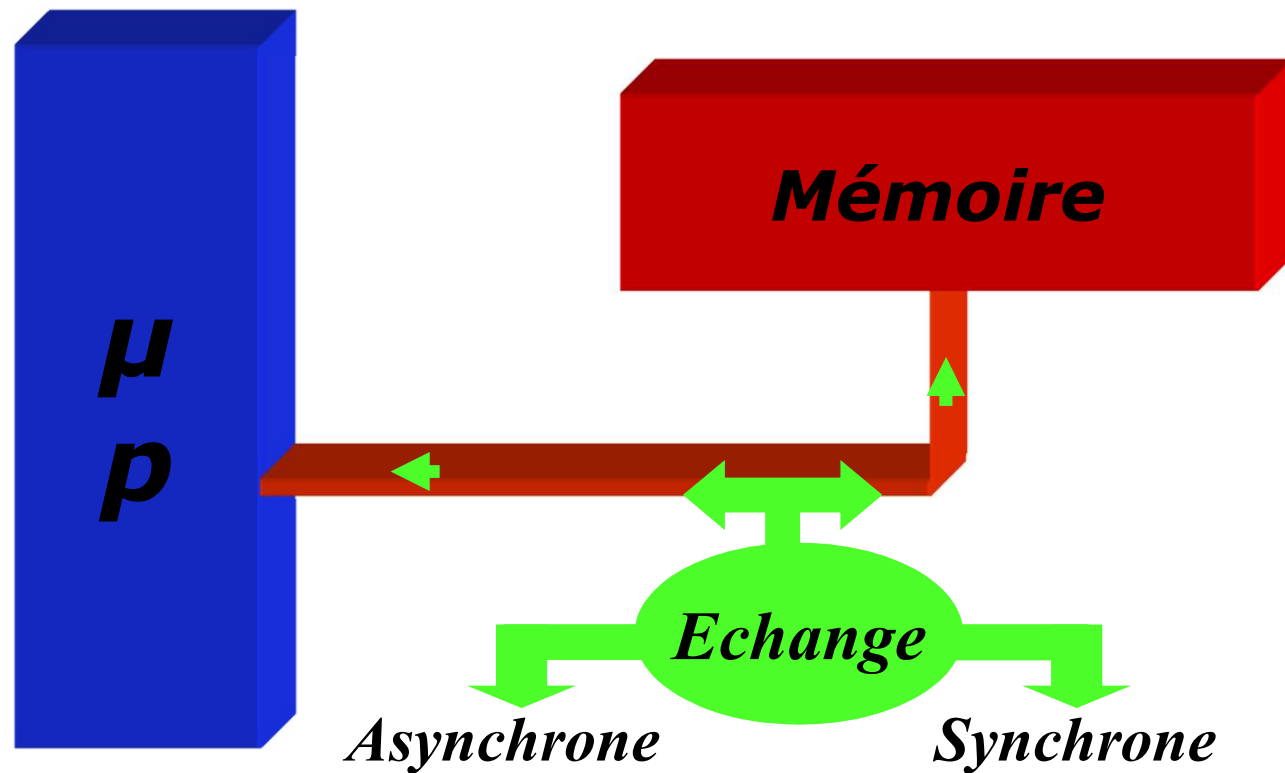
- ✓ Structure d'entée/sortie par instructions mémoires :
Memory Mapped I/O
- ✓ Structure d'entrée/sortie par instruction E/s :
Isolated I/O



Sommaire:

- Objectifs
- Informatique Industrielle
- Du système câblé au microprogrammé
- Systèmes à microprocesseur
 - ✓ Système minimal
 - ✓ Microprocesseur
 - ✓ Mémoire
 - ✓ Interfaces d'entrée/sortie
- **Exploitation d'un système à μp**
- Du microprocesseur au microcontrôleur

Modes d'échange du μp avec la mémoire : (1/2)



Modes d'échange du μp avec la mémoire : (2/2)

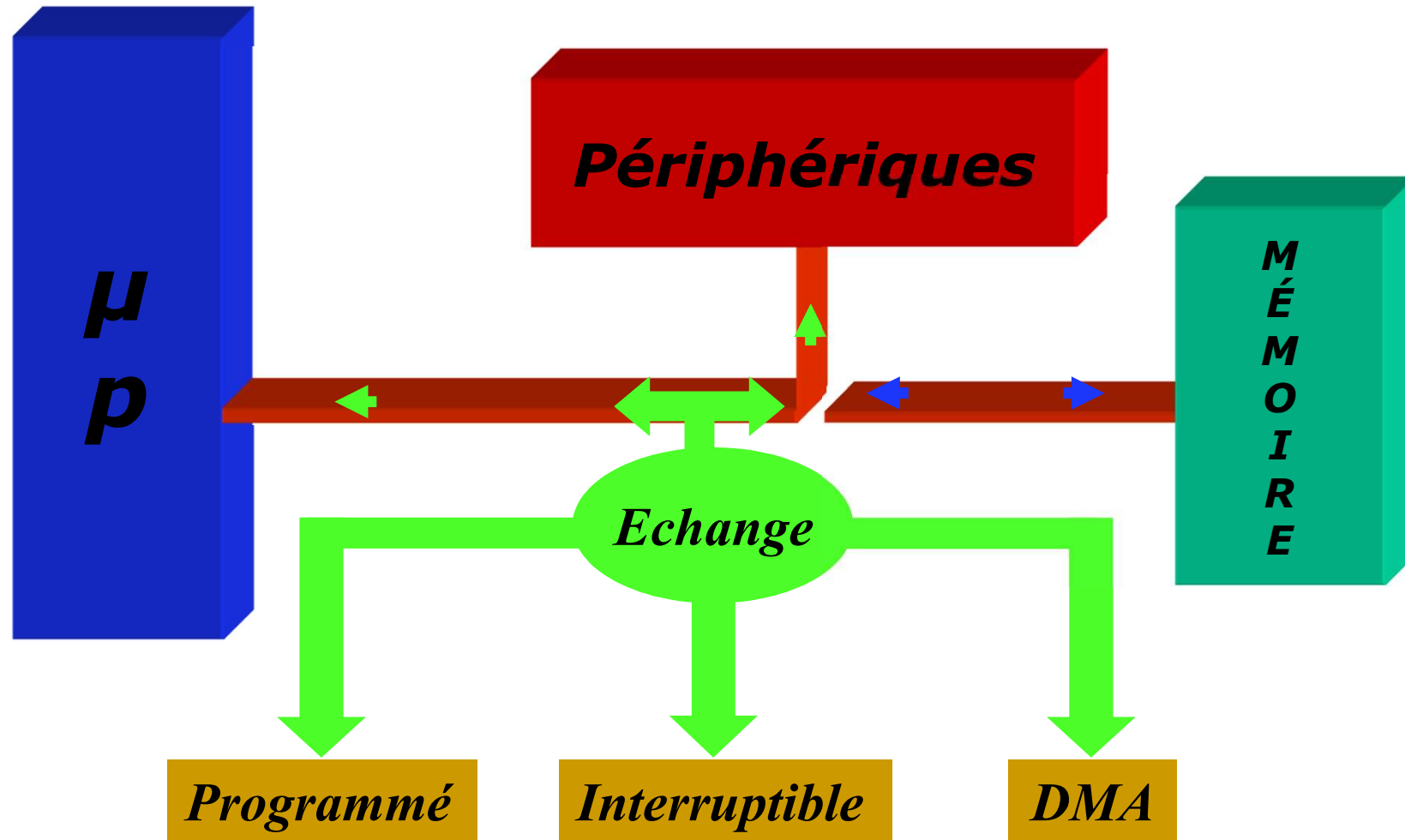
Échange synchrone :

Le μp va chercher en mémoire les adresses ou les données, il communique donc sans arrêt avec la mémoire soit pour une écriture, soit pour une lecture. Le temps pour une opération de lecture ou d'écriture est connu, fixe et caractéristique du μp , c'est le temps de cycle, qui est également appelé cycle mémoire. Durant un tel cycle, le μp doit pouvoir effectuer son opération de lecture ou d'écriture. cela suppose que la mémoire soit au moins aussi rapide que le μp . dans ce cas le μp gère les échanges avec la mémoire et ces échanges se font en synchronisme avec l'horloge du μp : c'est le mode synchrone.

Échange asynchrone :

Si la mémoire a un temps d'accès supérieur au temps de cycle du μp , les échanges se font alors avec un autre mode, qui comprend une demande de service du μp à la mémoire.

Modes de commande des transferts E/S : (1/3)

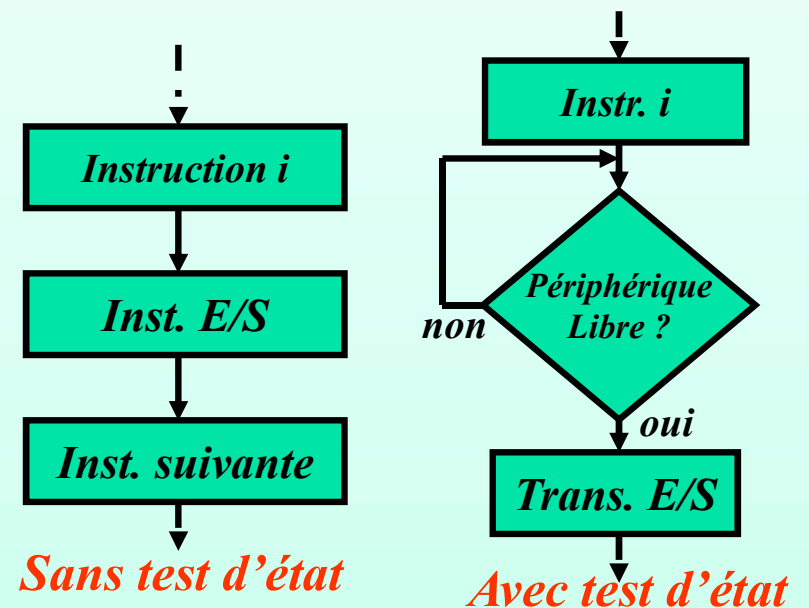


Modes de commande des transferts E/S : (2/3)

Mode programmé avec ou sans test d'état :

Les transferts E/S sont prévus à des moments bien précis dans le programme utilisateur.

C'est donc le μ p qui a l'initiative de commander le transfert et ceci par la rencontre d'une instruction de transfert : il s'agit du mode programmé.



On parle aussi de méthode par scrutation (*polling*) qui permet d'interroger régulièrement les périphériques afin de savoir si une nouvelle donnée est présente.

? Coûteux en temps (multiplier par le nombre de périphérique à interroger)

Modes de commande des transferts E/S : (3/3)

Mode interruptible :

Les transferts peuvent également s'effectuer à l'initiative du périphérique et non plus du μp . Ainsi, le périphérique envoie une demande d'interruption au μp qui répond par l'envoi d'un signal pour indiquer qu'il a reçu la demande, termine l'exécution de l'instruction en cours de traitement et se mettra à la disposition du périphérique.

Accès direct à la mémoire :

Lorsqu'un périphérique a un nombre important d'informations à faire rentrer en mémoire ou lorsqu'il doit en recevoir une quantité importante de la mémoire, le transfert par interruption est long. Un mode de transfert E/S particulier est alors possible : c'est l'accès direct à la mémoire.

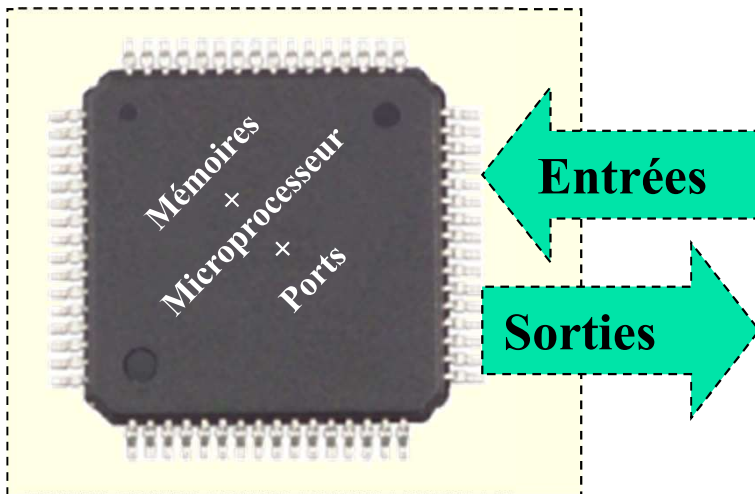
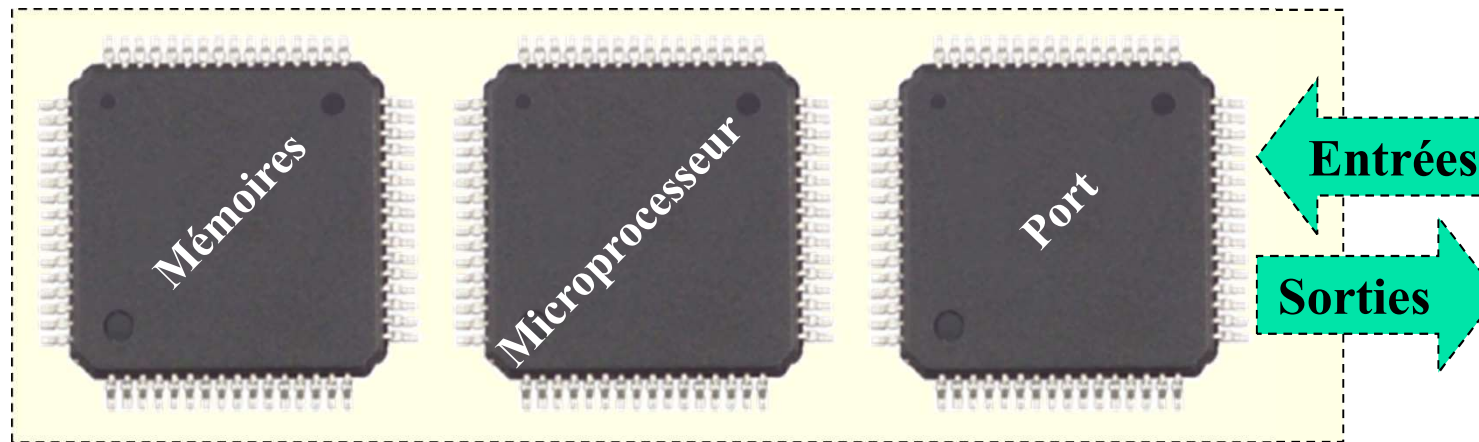
Le périphérique peut alors échanger directement avec la mémoire, les bus d'adresses, de données et de commande étant déconnectés du μp par leur mise à l'état haute impédance



Sommaire:

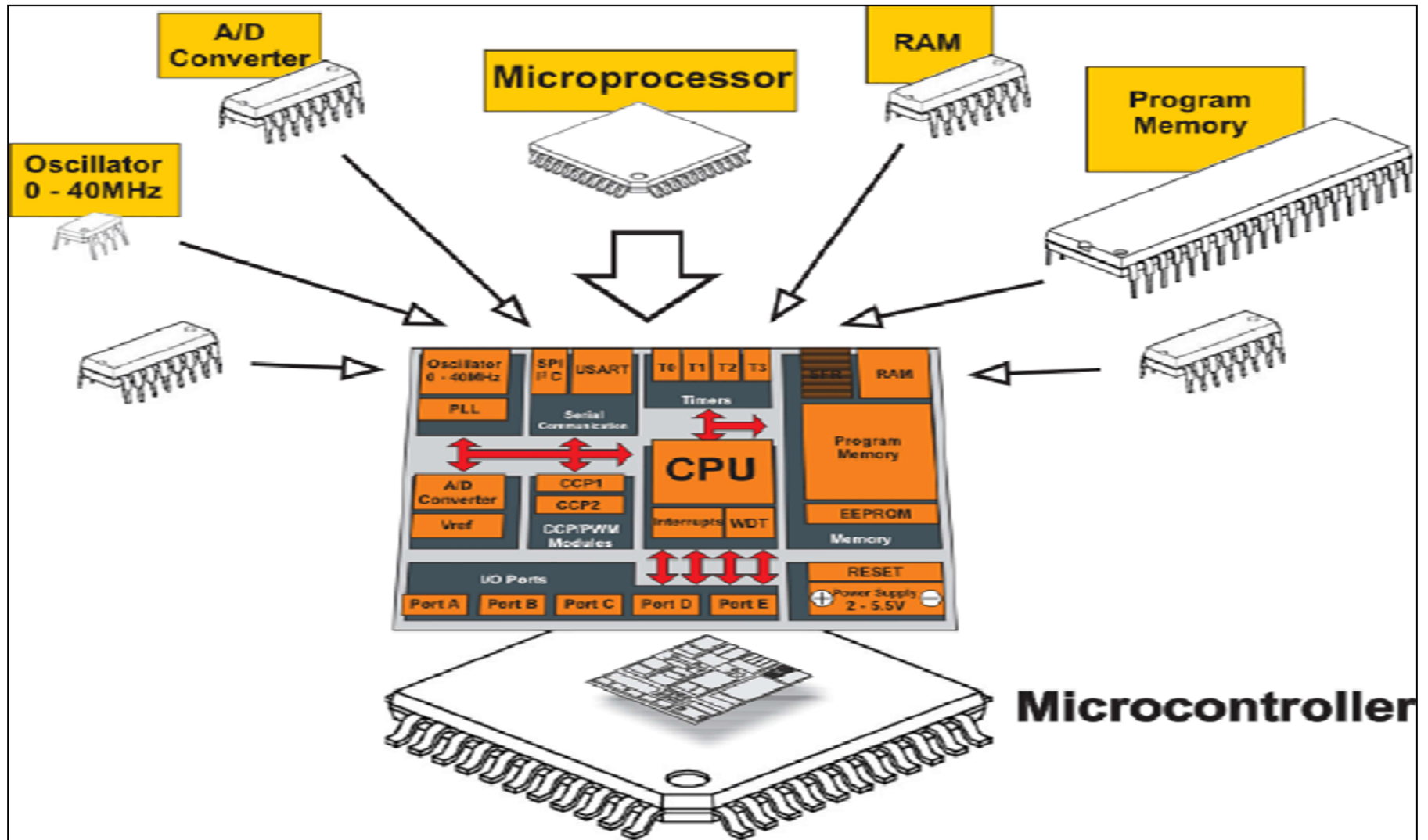
- Objectifs
- Informatique Industrielle
- Du système câblé au microprogrammé
- Systèmes à microprocesseur
 - ✓ Système minimal
 - ✓ Microprocesseur
 - ✓ Mémoire
 - ✓ Interfaces d'entrée/sortie
- Exploitation d'un système à μp
- **Du microprocesseur au microcontrôleur**

Du microprocesseur au microcontrôleur: (1)



Microcontrôleur
=
Microprocesseur + Mémoires + Ports
intégrés dans le même boîtier

Du microprocesseur au microcontrôleur: (2)



Du microprocesseur au microcontrôleur: (3)

Avantage des μC :

Les μC = avantage des μP mais limités aux applications ne nécessitant pas trop de puissance de calcul et un nombre de composant très réduit.

Les avantages des microcontrôleurs :

- ✓ Diminution de l'encombrement du matériel et du circuit imprimé
- ✓ Simplification du tracé du circuit imprimé p(lus besoin de tracer de bus !)
- ✓ Augmentation de la fiabilité du système
 - nombre de composants réduit
 - connexions composants/supports et composant circuit imprimé
- ✓ Intégration en technologie MOS, CMOS, ou HCMOS
 - diminution de la consommation
- ✓ Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux:
 - moins cher que les composants qu'il remplace
 - réduction des coûts de main d'oeuvre (conception et montage)
- ✓ Environnement de programmation et de simulation évolués

Défauts des μ C:

- ✓ Le microcontrôleur est souvent surdimensionné devant les besoins de l'application
- ✓ Investissement dans les outils de développement
- ✓ Écrire les programmes, les tester et tester leur mise en place sur le matériel qui entoure le microcontrôleur
- ✓ Incompatibilité possible des outils de développement pour des microcontrôleurs de même marque.
- ✓ Les microcontrôleurs les plus intégrés et les moins coûteux sont ceux disposant de ROM programmables par masque
Fabrication uniquement en grande série >1000
- ✓ Défaut relatif car il existe maintenant systématiquement des versions OTPROM un peu plus chères.