

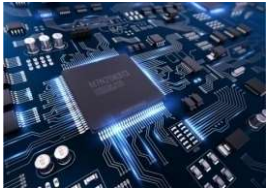
Cours Circuits Numériques

Fatiha El Hatmi

ISAMM

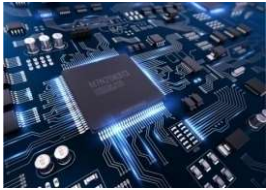
Cycle ingénieur en sciences appliquées et en technologie

2024/2025



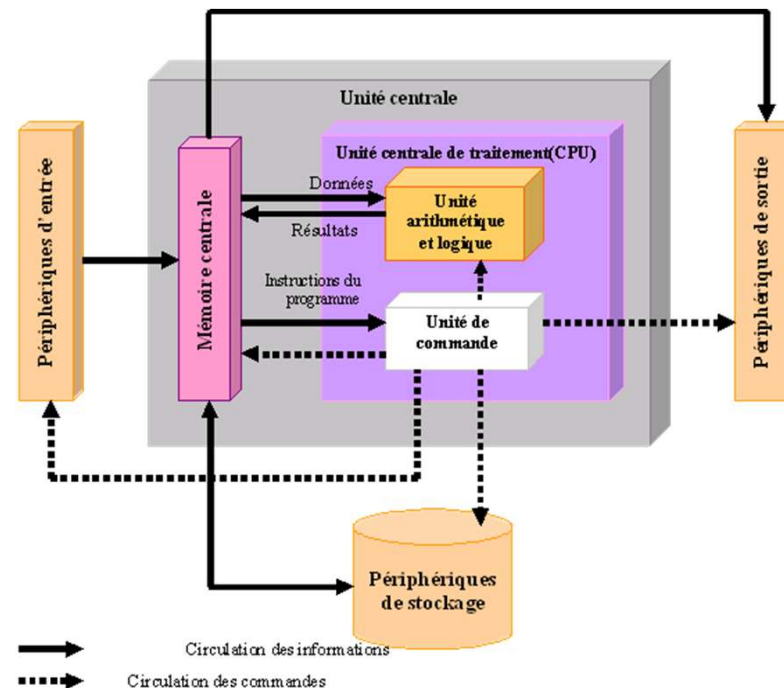
Plan du cours

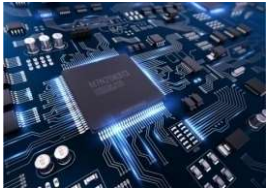
- ❑ Chapitre 1: Historique des calculateurs
- ❑ Chapitre 2: Le transistor et portes logiques
- ❑ Chapitre 3: Représentation (codage) des données
- ❑ Chapitre 4: Circuits combinatoires
- ❑ Chapitre 5: Architecture de base d'un ordinateur**
- ❑ Chapitre 6: Les mémoires
- ❑ Chapitre 7: Jeu d'instructions



Chapitre V

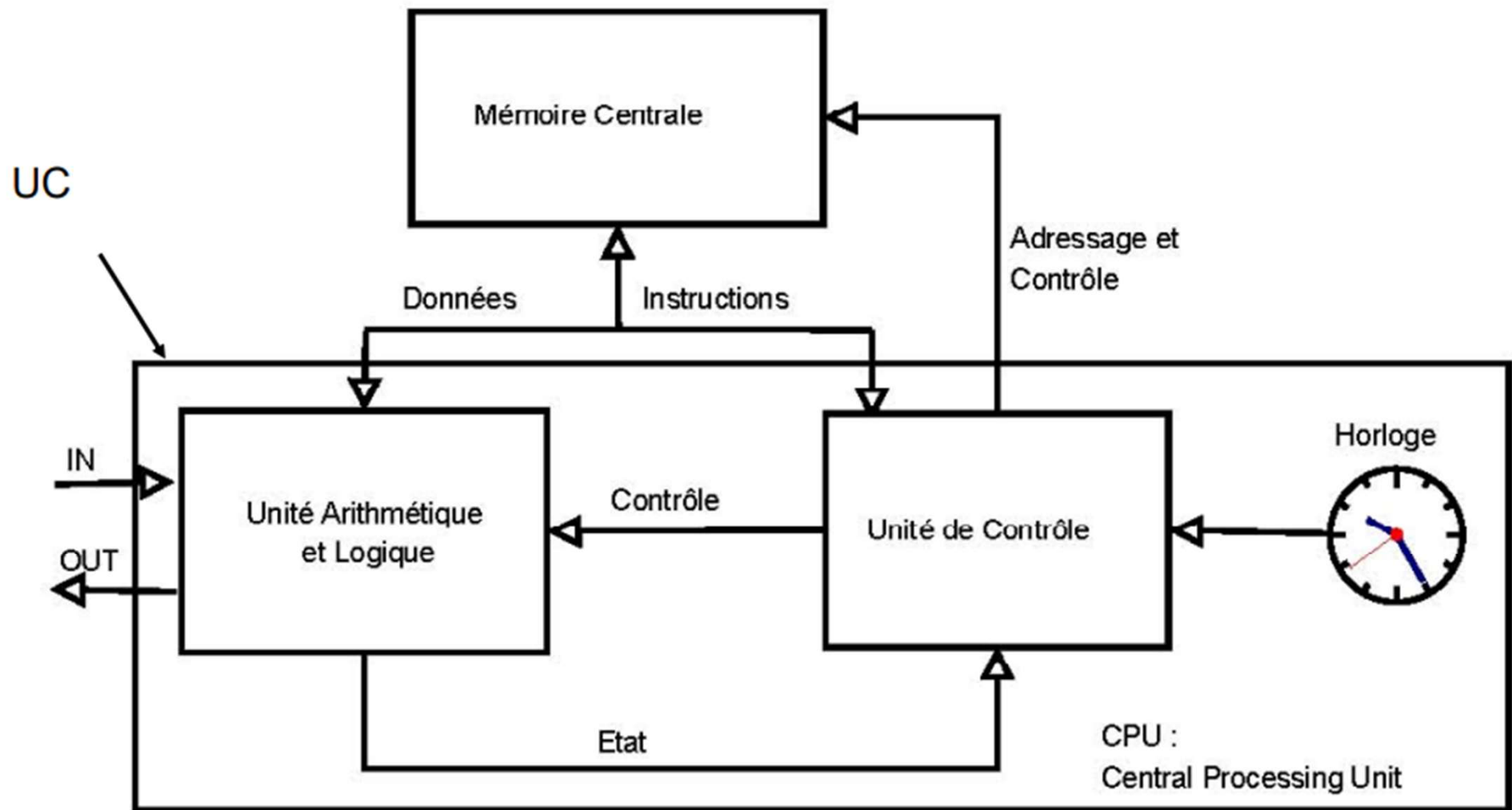
Architecture de base d'un ordinateur

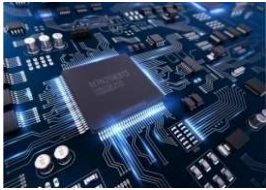




Architecture de Von Neumann

❑ Machine Von Neumann: architecture matérielle:

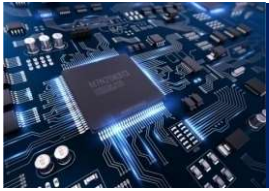




Le microprocesseur

□ Le processeur/microprocesseur:

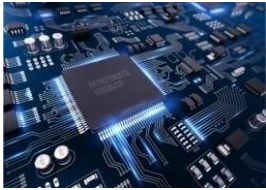
- Les premiers processeurs nécessitaient un espace important, puisqu'ils étaient construits à base de tubes électroniques ou de relais électromécaniques.
- Leur création a pour origine les travaux de John von Neumann, qui répondaient aux difficultés liées à la reprogrammation de calculateurs comme l'ENIAC où il était nécessaire de recâbler le système pour faire fonctionner un nouveau programme.
- Dans cette architecture, une unité de contrôle se charge de coordonner un processeur (ayant accès aux entrées/sorties) et la mémoire.
- Un **processeur** (ou **unité centrale de traitement, UCT** ou en anglais *central processing unit, CPU*) est un composant présent dans de nombreux dispositifs électroniques qui exécute les instructions machine des programmes informatiques.
- Avec la mémoire, c'est notamment l'une des fonctions qui existent depuis les premiers ordinateurs. Un processeur construit en un seul circuit intégré est un microprocesseur.



Le microprocesseur

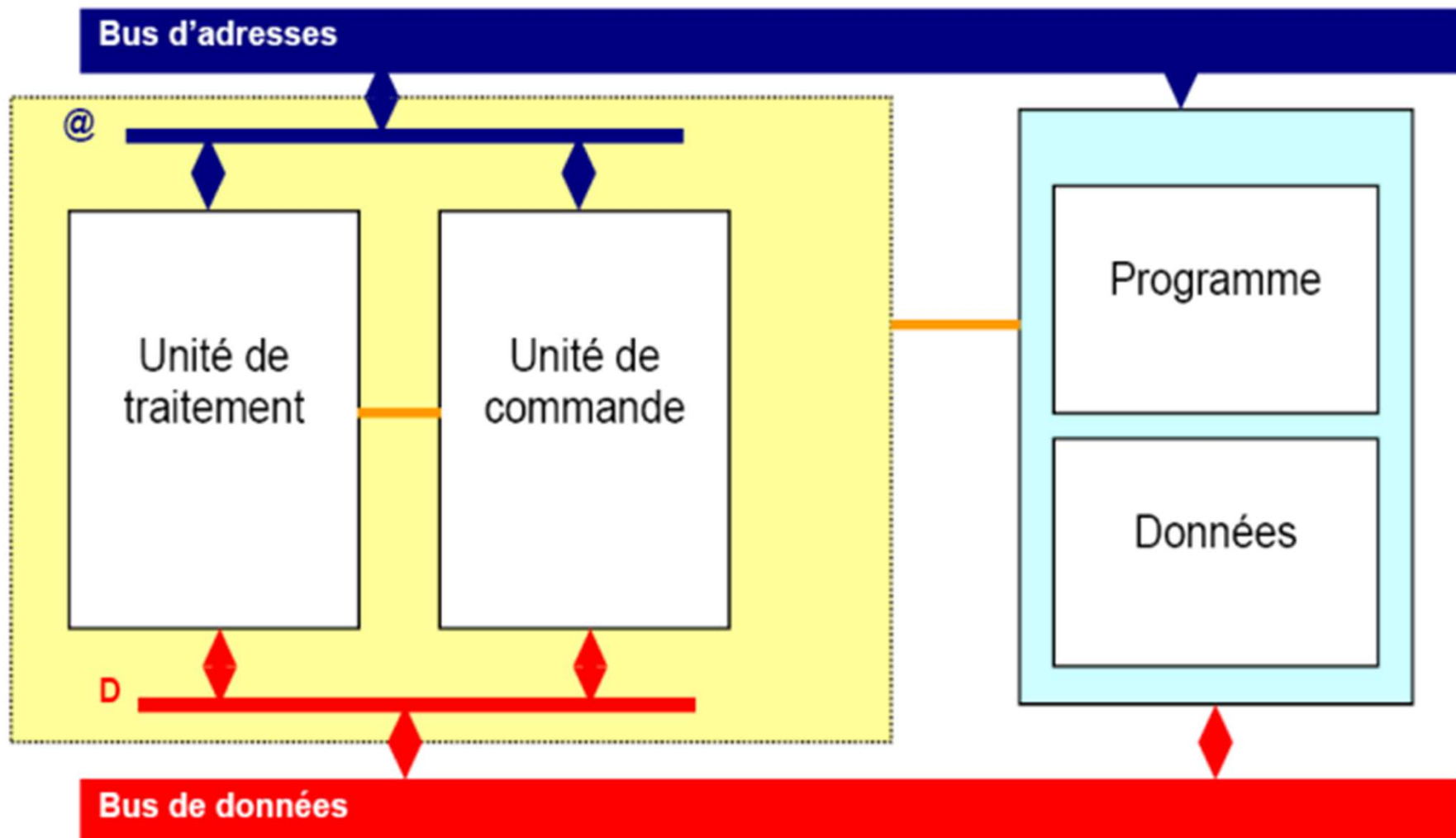
□ Le processeur/microprocesseur:

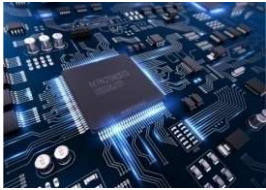
- L'invention du transistor, en 1948, ouvrit la voie à la miniaturisation des composants électroniques. Car, auparavant, les ordinateurs prenaient la taille d'une pièce entière du fait de l'utilisation de tubes à vide volumineux, gros consommateurs d'énergie et générant beaucoup de chaleur.
- L'introduction du microprocesseur dans les années 1970 a marqué de manière significative la conception et l'implémentation des unités centrales de traitement.
- Depuis l'introduction du premier microprocesseur (**Intel 4004**) en 1971 et du premier microprocesseur employé couramment (**Intel 8080**) en 1974, cette classe de processeurs a presque totalement dépassé toutes les autres méthodes d'implémentation d'unité centrale de traitement.
- Le microprocesseur est construit autour de 2 éléments principaux :
 - ▶ Une unité de commande
 - ▶ Une unité de traitement



Le microprocesseur

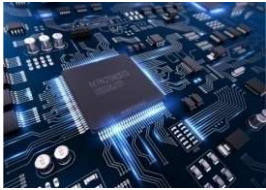
□ Architecture complète :





Unité de contrôle

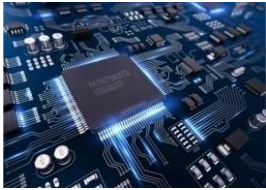
- Le rôle de l'unité de contrôle (ou unité de commande) est de :
 - **coordonner** le travail de toutes les autres unités (UAL , mémoire,....)
 - et d'assurer la **synchronisation** de l'ensemble.
 - Elle assure :
 - la **recherche** (lecture) de l'instruction et des données à partir de la mémoire,
 - le **décodage** de l'instruction et l'exécution de l'instruction en cours
 - et **prépare** l'instruction suivante.
- ➡ 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.



Unité de contrôle

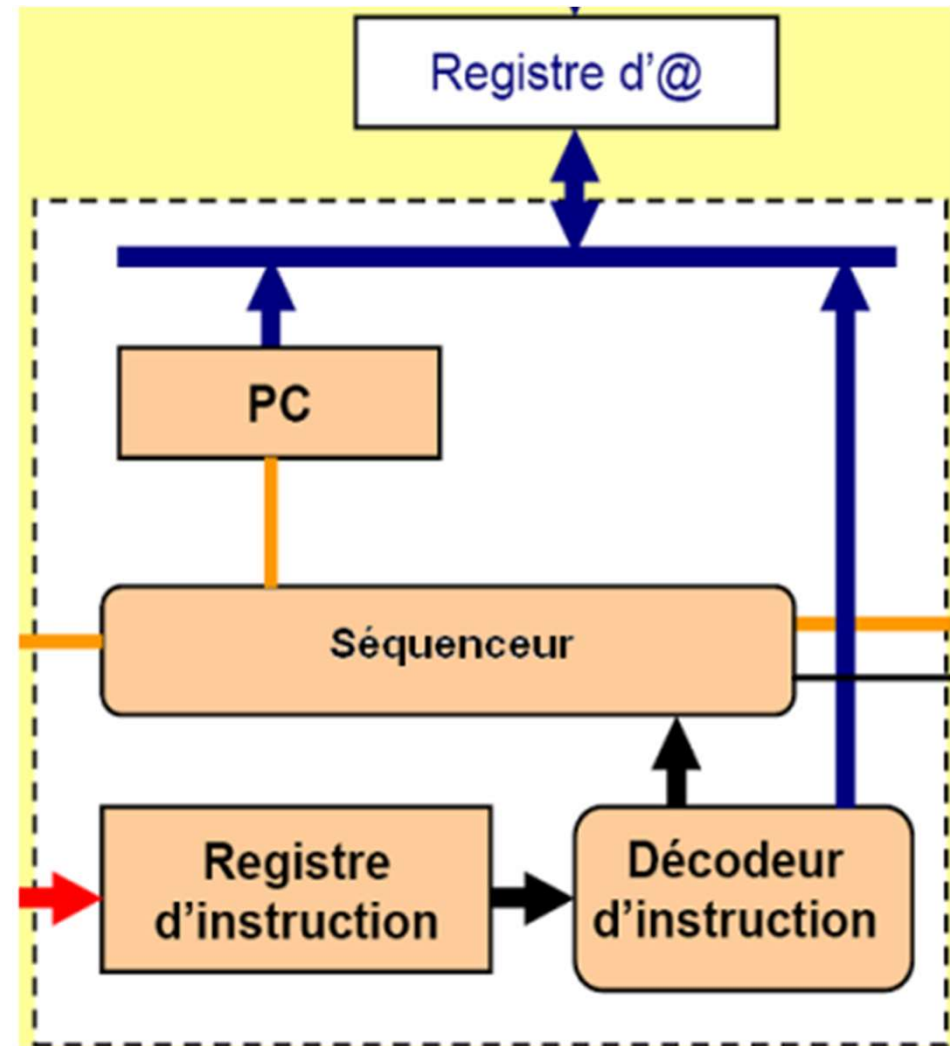
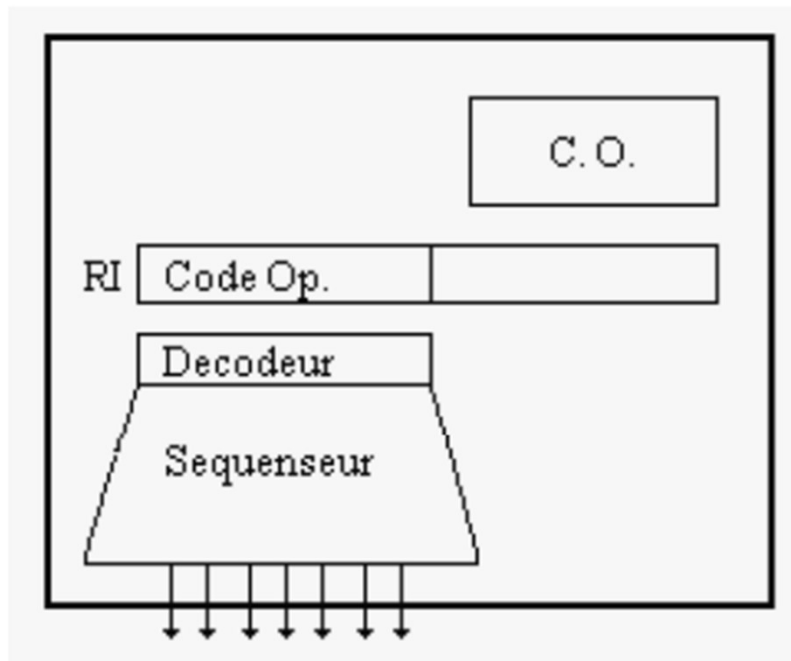
□ Blocs de l'unité de contrôle (commande) :

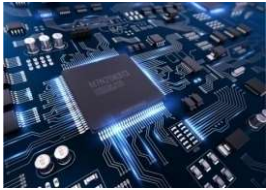
1. **Le compteur de programme (PC : Programme Counter) appelé aussi Compteur Ordinal (CO)** est constitué par un registre dont le contenu est initialisé avec l'adresse de la première instruction du programme. Il contient toujours l'adresse de la prochaine instruction à exécuter.
2. **Le registre d'instruction 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.
3. **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 micro-programmée.



Unité de contrôle

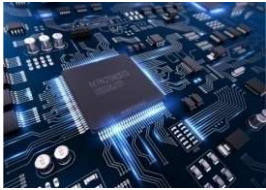
□ Schéma de l'unité de contrôle (commande) :





L'unité de traitement

- L'unité arithmétique et logique réalise une **opération élémentaire** (addition, soustraction, multiplication, . . .).
 - L'UAL regroupe **les circuits** qui assurent les fonctions logiques et arithmétiques de bases (ET,OU,ADD,SUS,.....).
 - L'UAL comporte un **registre accumulateur** (ACC) : c'est un registre de travail qui sert à stocker un opérande (données) au début d'une opération et le résultat à la fin.
- ➡ Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions



L'unité de traitement

□ Blocs de l'unité de traitement :

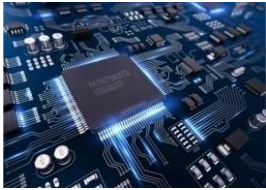
1. **Les accumulateurs** sont des registres de travail qui servent à stocker une opérande au début d'une opération arithmétique et le résultat à la fin de l'opération.
2. **L'Unité Arithmétique et Logique (UAL)** est un circuit complexe qui assure les fonctions logiques (ET, OU, Comparaison, Décalage, etc...) ou arithmétique (Addition, soustraction...).
3. **Le registre d'état** est généralement composé de 8 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**)

Débordement (**overflow : OV ou V**)

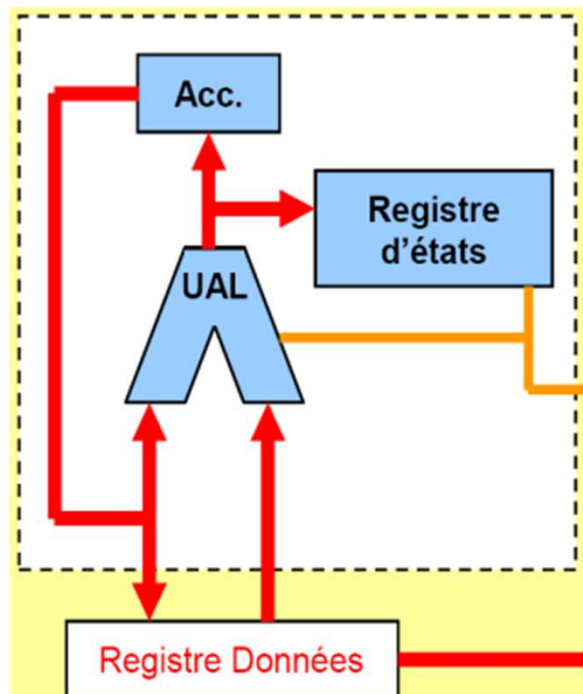
Zéro (**Z**)

...

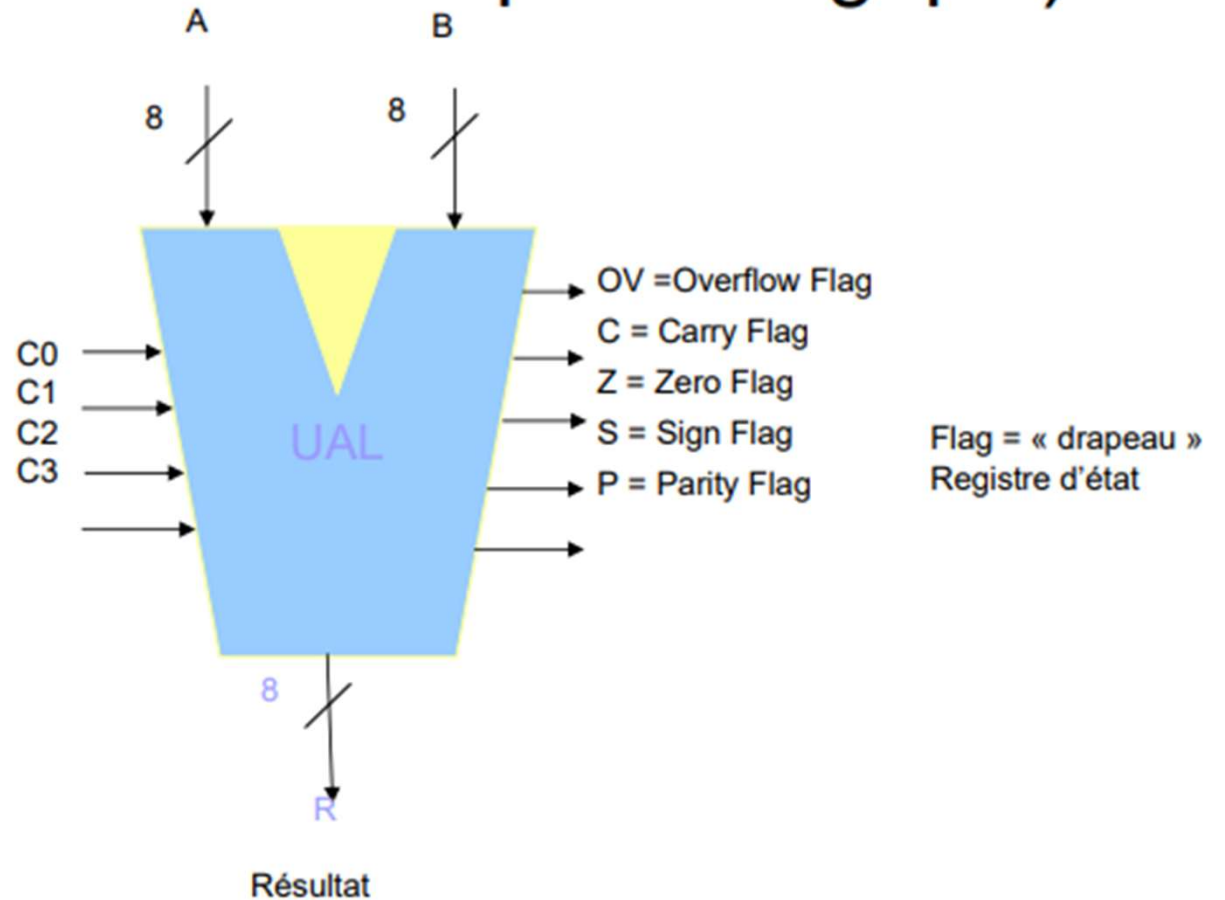


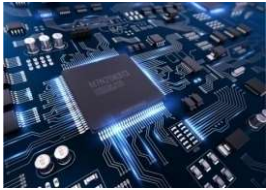
L'unité de traitement

UAL : Unité Arithmétique et Logique)



Sélection de l'opération





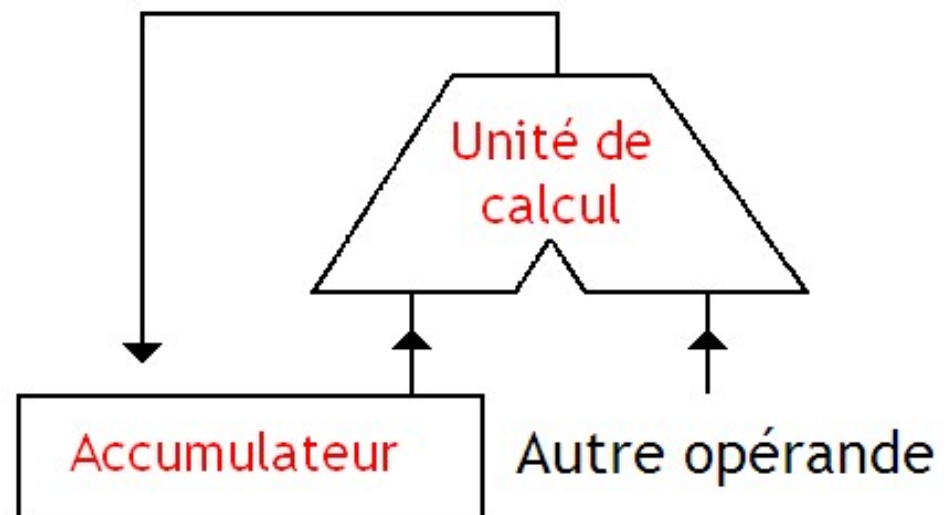
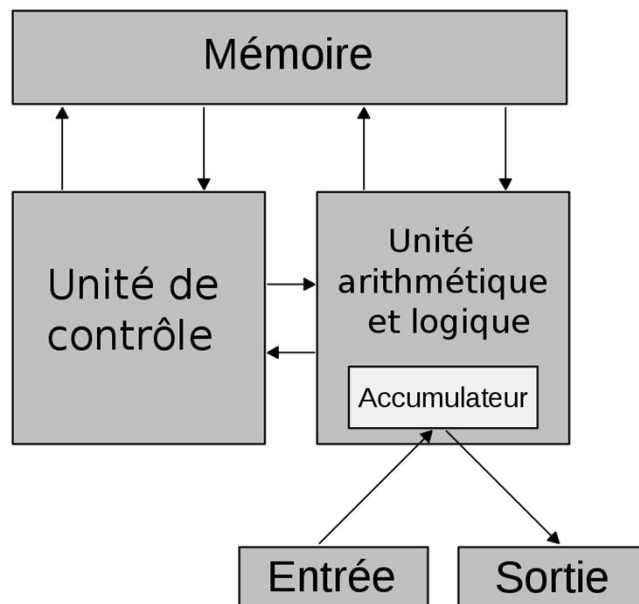
L'UAL

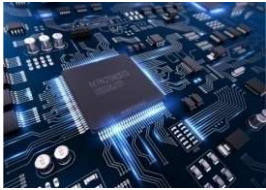
□ Microprocesseur à accumulateur:

● Exemple de programme :

Le code assembleur **8051** qui suit travaille avec l'accumulateur primaire :

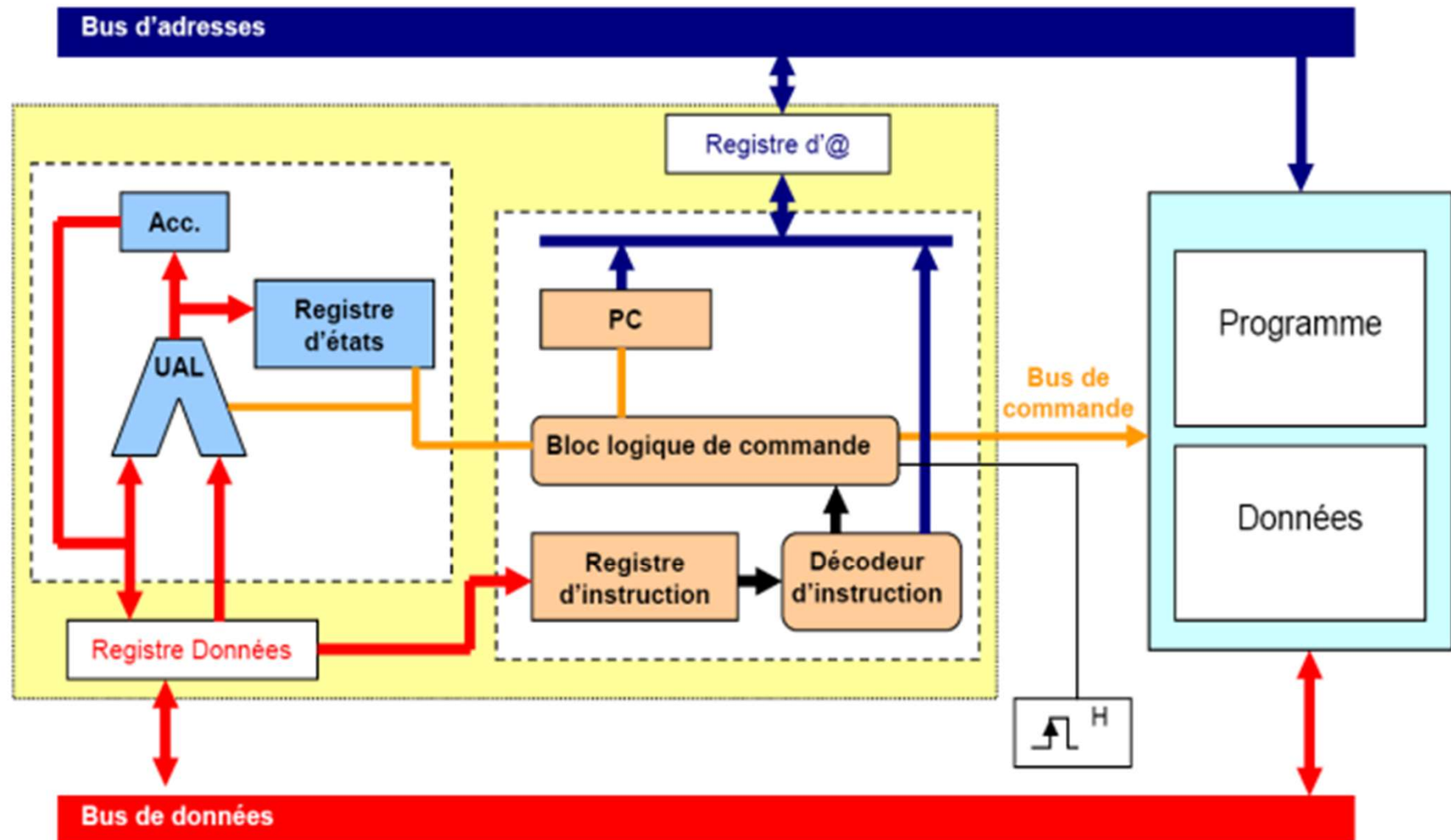
```
MOV A,R2 ;stocke R2 dans l'accumulateur  
ADD A,R4 ;ajoute la valeur de R4 dans l'accumulateur  
MOV R5,A ;stocke la valeur de l'accumulateur dans R5
```

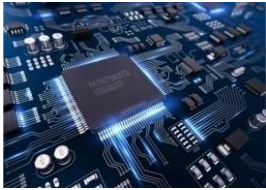




Le microprocesseur

□ Architecture détaillée :





La mémoire principale

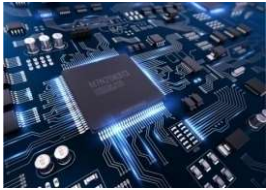
- La mémoire centrale (MC) représente l'espace de travail de l'ordinateur .
- C'est l'organe principal de rangement des informations utilisées par le processeur.
- Dans un ordinateur pour exécuter un programme il faut le charger (copier) dans la mémoire centrale .
- Le temps d'accès à la mémoire centrale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

→ Mémoire vive (volatile) : RAM

- statique
- dynamique

→ Mémoire morte (non-volatile) : ROM

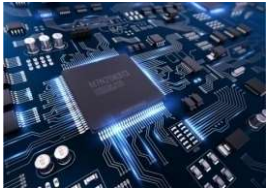
- PROM
- EPROM
- EEPROM
- flash



La mémoire principale

❑ Caractéristiques de la mémoire principale:

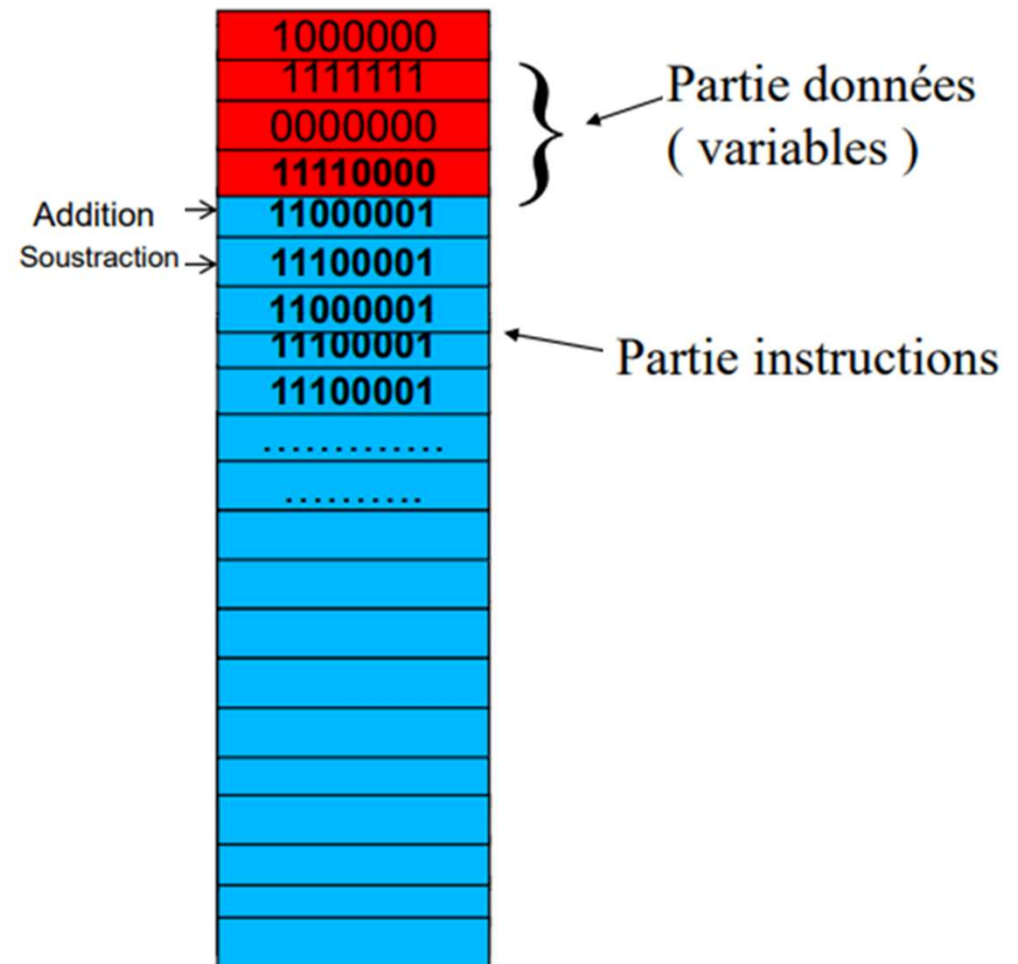
- ❑ Mémoire vive
- ❑ Accès aléatoire (RAM)
- ❑ A lecture-écriture
- ❑ Volatile
- ❑ Capacité limitée (possibilité d'extension)
- ❑ Communique au moyen des bus d'adresses et de données
- ❑ Types
 - Mémoires statiques (SRAM) : à base de bascules D
 - Mémoires dynamiques (DRAM) : à base de condensateurs

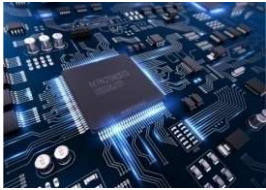


La mémoire principale

- La **mémoire volatile** (ou **non rémanente, temporaire** ou **à court terme**) est une mémoire informatique qui a besoin d'alimentation électrique continue pour conserver l'information qui y est enregistrée.
- Lorsque l'alimentation électrique est interrompue, l'information contenue dans la mémoire volatile est, quasi immédiatement, perdue.
- De façon générale, plus la quantité de mémoire est importante, plus d'applications simultanément peuvent être lancées.
- D'autre part, plus celle-ci est rapide, plus le système réagit vite. Le but est donc de mieux l'organiser au mieux pour en tirer le maximum de performances.

◆ Structure d'un programme en MP:

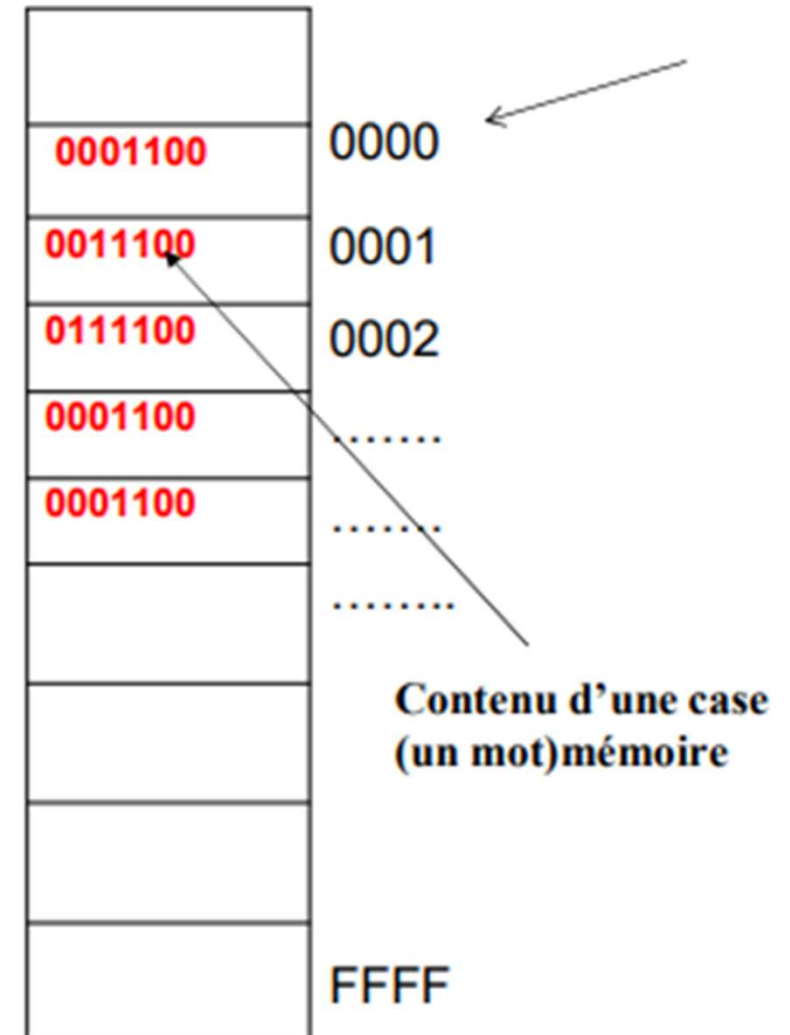


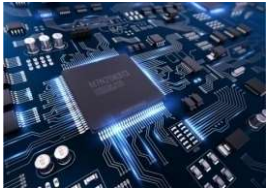


La mémoire principale

□ Caractéristiques de la MP:

- 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.

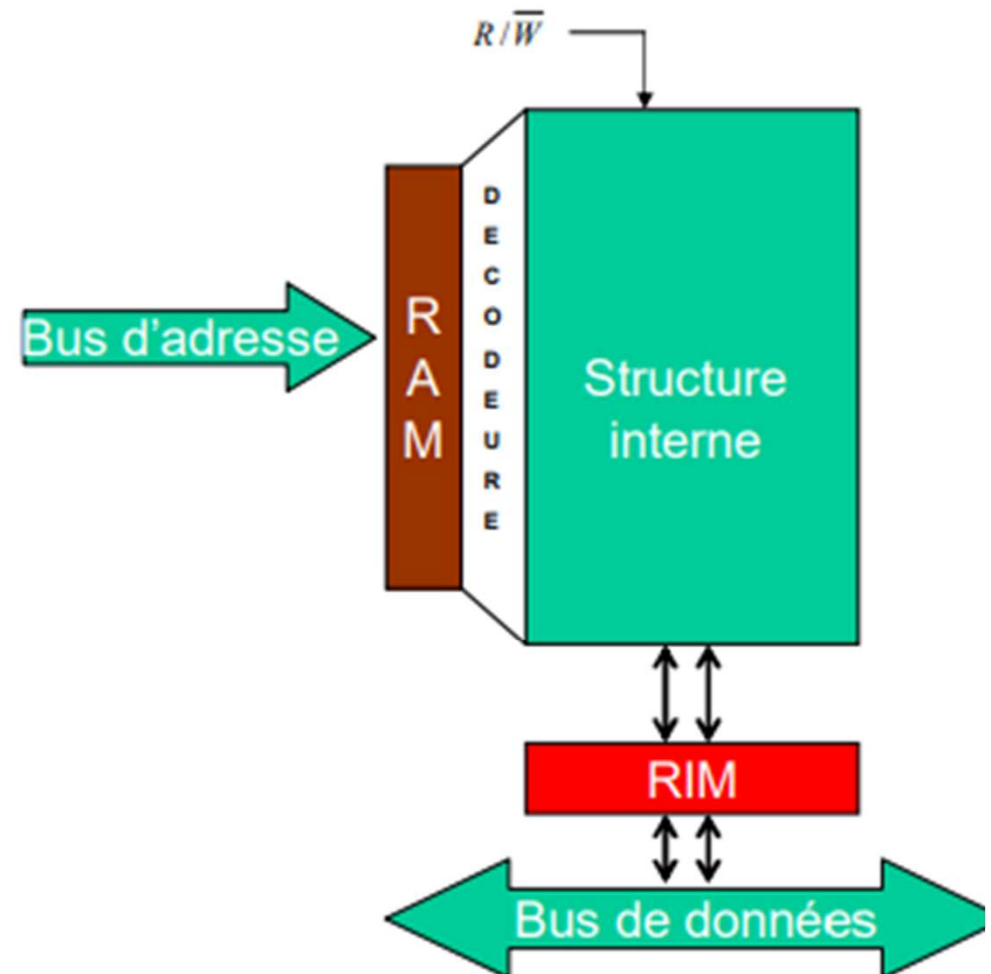


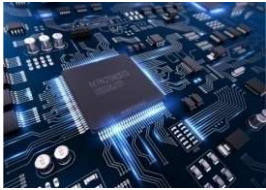


La mémoire principale

□ Structure physique de la MP:

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

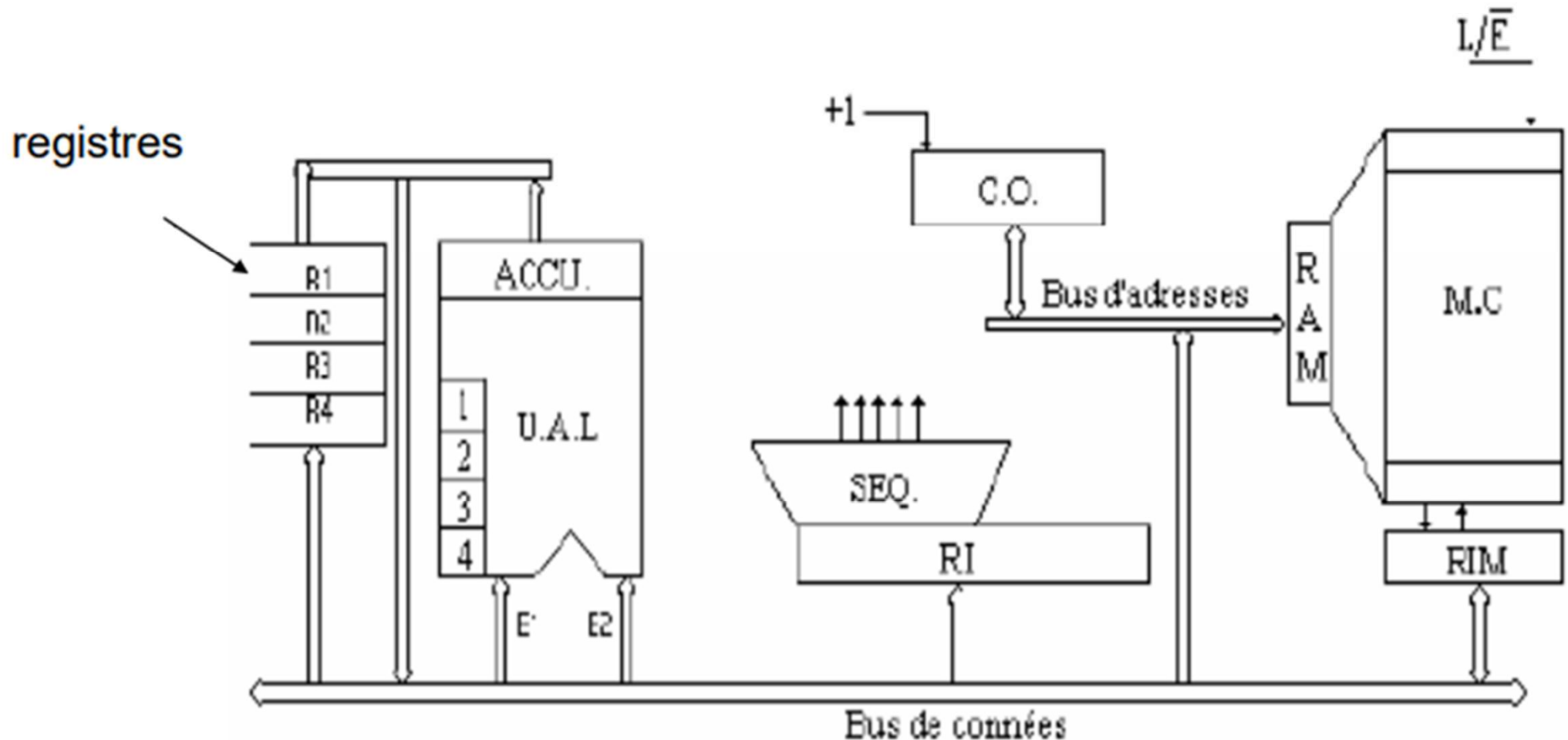


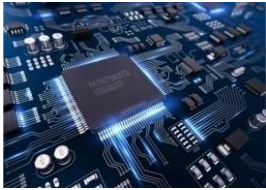


La mémoire principale

□ Structure physique de la MP:

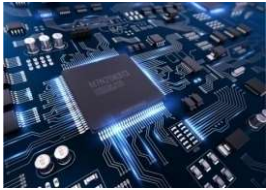
Une machine avec des registres de travail





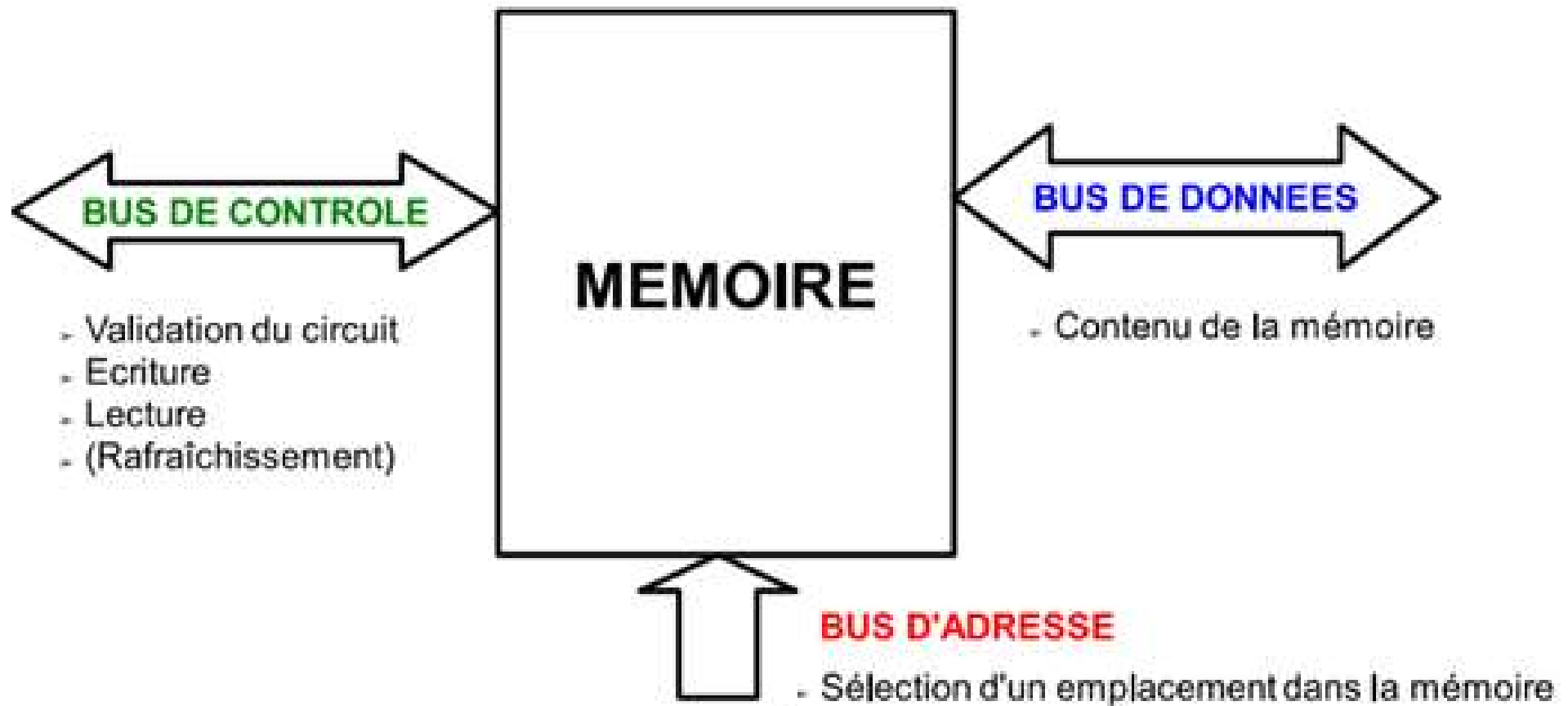
Opérations sur la mémoire

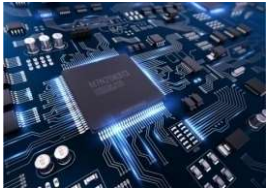
- ❑ Seul le processeur peut modifier l'état de la mémoire. Chaque emplacement mémoire conserve les informations que le processeur y écrit jusqu'à coupure de l'alimentation électrique, où tout le contenu est perdu (contrairement au contenu des mémoires externes comme les disquettes et disques durs). On parle de mémoire vive.
- ❑ Les seules opérations possibles sur la mémoire sont :
 - **Ecriture** : d'un emplacement : le processeur donne une valeur et une adresse, et la mémoire range la valeur à l'emplacement indiqué par l'adresse ;
 - **Lecture** : d'un emplacement : le processeur demande à la mémoire la valeur contenue à l'emplacement dont il indique l'adresse. Le contenu de l'emplacement auquel le processeur accède en lecture demeure inchangé.



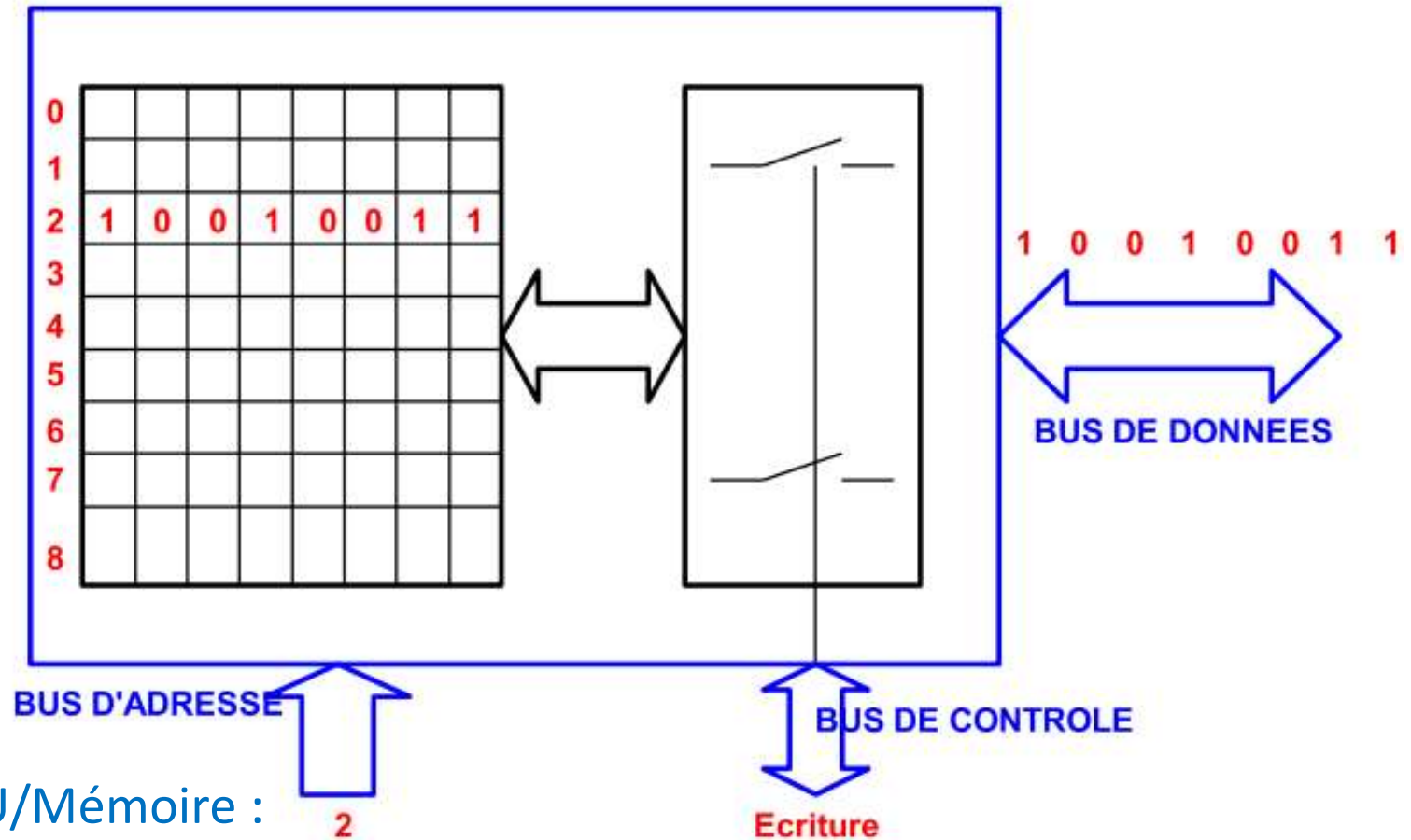
Opérations sur la mémoire

LA MEMOIRE

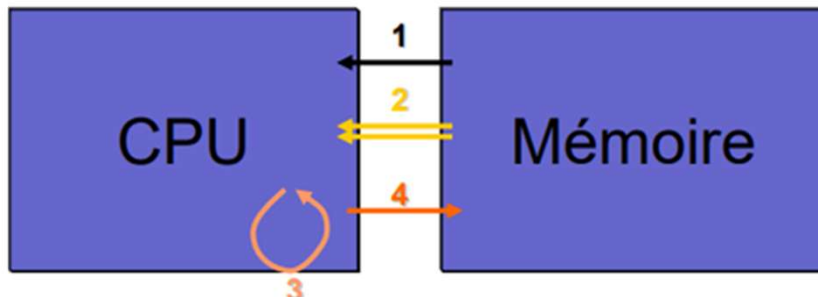




Opérations sur la mémoire



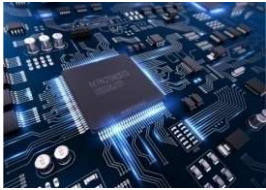
• Communication CPU/Mémoire :



- (1) Charger une instruction depuis la mémoire
- (2) Charger les opérandes depuis la mémoire
- (3) Effectuer les calculs
- (4) Stocker le résultat en mémoire

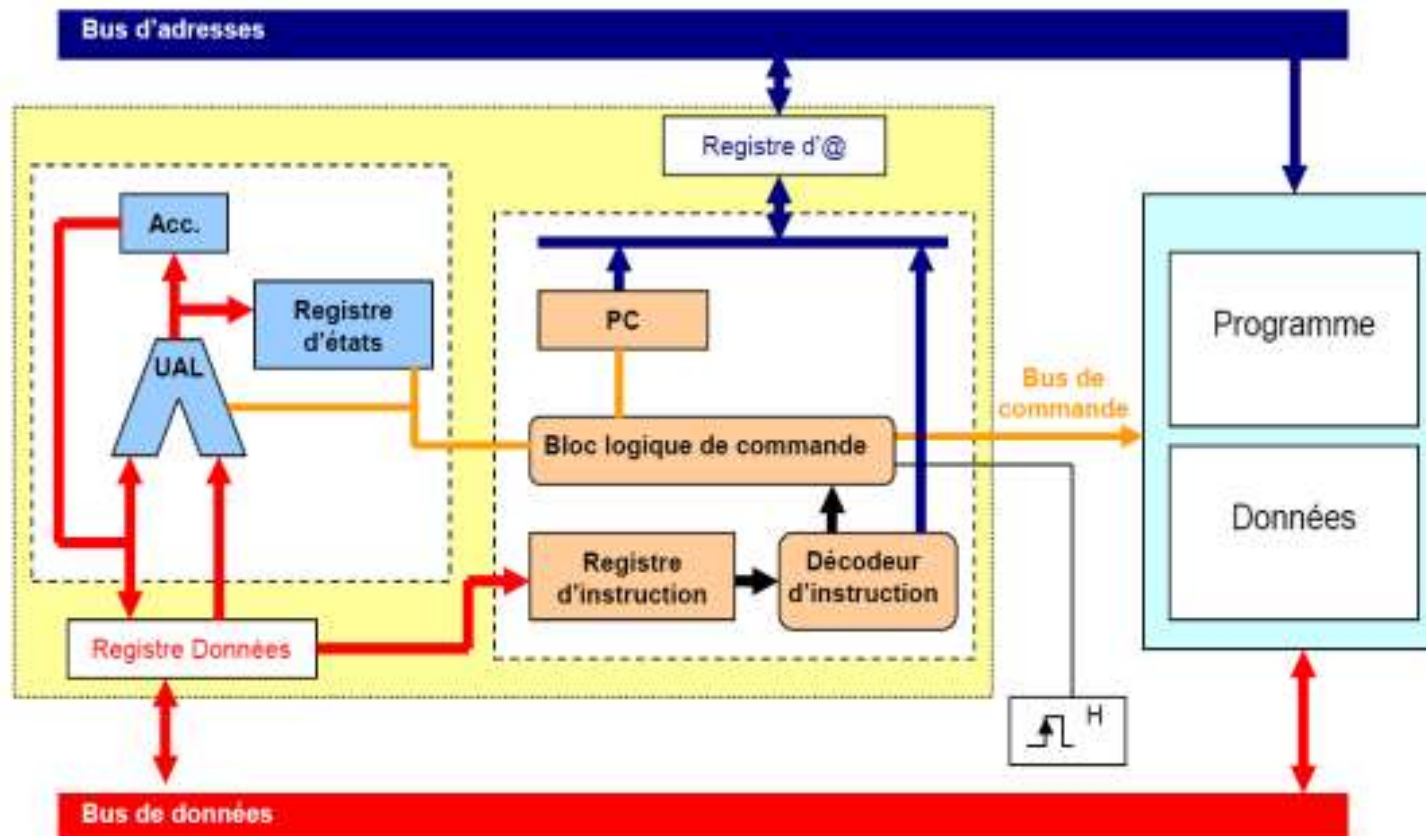


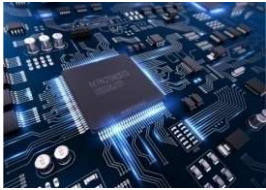
-
- The diagram illustrates the internal structure of a computer system and its connection to external buses. At the top, a dark blue bar represents the **Bus d'adresses** (Address Bus). Below it, a large yellow box contains the internal components of the system. Inside this box, a dashed line encloses the **Register d'@** (Address Register) and the **PC** (Program Counter). The **PC** is connected to the **Bus d'adresses** via a bidirectional arrow. Below the **PC** is the **Bloc logique de commande** (Control Logic Block), which is connected to the **PC** and the **Bus d'adresses**. To the left of the **Bloc logique de commande** is the **Register d'instruction** (Instruction Register), which is connected to the **Bloc logique de commande** and the **Bus d'adresses**. Below the **Register d'instruction** is the **Décodeur d'instruction** (Instruction Decoder), which is connected to the **Register d'instruction** and the **Bus d'adresses**. To the right of the **Décodeur d'instruction** is the **Register d'états** (Status Register), which is connected to the **Décodeur d'instruction** and the **Bus d'adresses**. To the left of the **Register d'états** is the **UAL** (Universal Arithmetic Logic), which is connected to the **Register d'états** and the **Bus d'adresses**. Below the **UAL** is the **Acc.** (Accumulator), which is connected to the **UAL** and the **Bus d'adresses**. At the bottom of the yellow box is the **Register Données** (Data Register), which is connected to the **Bus de données** (Data Bus) via a bidirectional arrow. The **Bus de données** is a red bar at the bottom of the diagram. To the right of the yellow box, a light blue box represents the **Programme** (Program) and **Données** (Data). The **Programme** is connected to the **Bus d'adresses** via a bidirectional arrow. The **Données** is connected to the **Bus de données** via a bidirectional arrow. A small square icon with a triangle and the letter **H** is located between the **Bus de données** and the **Programme/Données** box.



Communication CPU/Mémoire

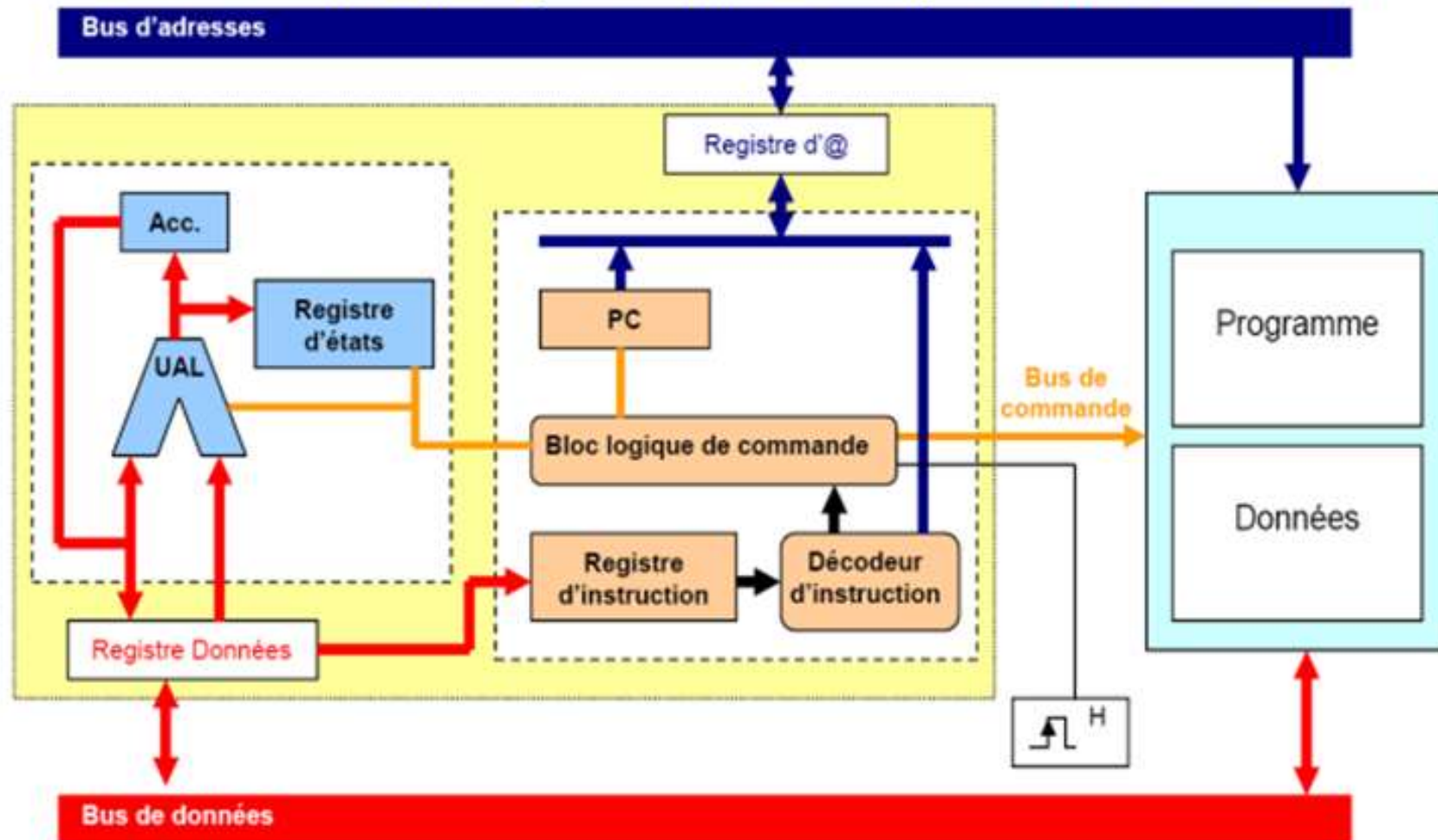
- L'unité de commande transforme l'instruction en une suite de commandes élémentaires nécessaires au traitement de l'instruction.
- Si l'instruction nécessite une donnée en provenance de la mémoire, l'unité de commande récupère sa valeur sur le bus de données.
- L'opérande est stocké dans le registre de données.

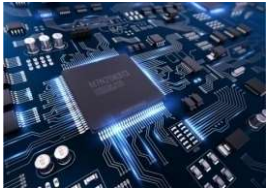




Communication CPU/Mémoire

- L'UAL réalise l'instruction.
- Les drapeaux sont positionnés (registre d'état).
- L'unité de commande positionne le PC pour l'instruction suivante.

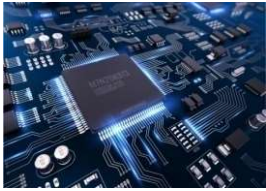




Les BUS

□ Liaison processeur-mémoire:

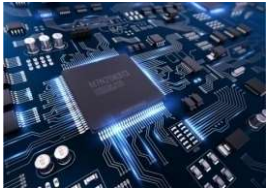
- ◆ En informatique, le mot bus, désigne l'ensemble des liaisons électrique (nappes, pistes de circuits imprimés, etc.) utilisées par plusieurs éléments matériels afin de communiquer entre eux.
- ◆ Si cette liaison relie deux éléments seulement, elle est appelée port matériel (port série, port parallèle, etc.)
- ◆ Les informations échangées entre la mémoire et le processeur circulent sur des bus.
- ◆ Un bus est simplement un ensemble de n fils conducteurs, utilisés pour transporter n signaux binaires.
- ◆ Un bus est formé en général de 50 à 100 lignes physiques distinctes, partagées en trois sous-ensembles fonctionnels :
 - ▶ **Le bus d'adresses (bus d'adressage ou bus mémoire)** : transporte les adresses des cases mémoires où le processeur souhaite lire ou écrire une donnée. C'est un bus unidirectionnel.
 - ▶ **Le bus de données** : transporte les instructions (opérations + données) à exécuter. C'est un bus bidirectionnel.
 - ▶ **Le bus de contrôle (bus de commandes)** : transporte les ordres et les signaux de synchronisation issus de l'UCC vers l'ensemble des composants matériels, et aussi les réponses de ces composants à l'UCC. C'est bus bidirectionnel.



Les BUS

□ Caractéristiques des BUS :

- ◆ Le bus d'adresse est un **bus unidirectionnel** : seul le processeur envoie des adresses. Il est composé de n fils ; on utilise donc des adresses de n bits. La mémoire peut posséder au maximum 2^n emplacements (adresses 0 à $2^n - 1$).
- ◆ Le bus de données est un **bus bidirectionnel**. Lors d'une lecture, c'est la mémoire qui envoie un mot sur le bus (le contenu de l'emplacement demandé) ; lors d'une écriture, c'est le processeur qui envoie la donnée.
- ◆ Le volume d'informations transmises simultanément:
- ◆ Cette quantité (en bits) correspond au nombre de lignes physiques servant à envoyer les données de manière parallèle. Une nappe de 32 fils permet de transmettre 32 bits en parallèle.
- ◆ Sa vitesse ou sa fréquence (en Hz): exprimant le nombre de paquets de données envoyés ou reçus par seconde. Il est possible de connaître la quantité de données qu'un bus peut transporter par unité de temps, en multipliant sa longueur de mot par sa fréquence.
- ◆ Exemple: Un bus de 16 bits, cadencé à une fréquence de 133 MHz possède donc un taux de transfert égal à : $16 * 133.10^6 = 2128 * 10^6 \text{ bit/s} = 266 * 10^6 \text{ octets/s}$.



Scénario d'exécution

Le microprocesseur ne comprend qu'un certain nombre d'instructions qui sont codées en binaire. Une instruction est composée de deux éléments :

- Le code opération : C'est un code binaire qui correspond à l'action à effectuer par le processeur
- Le champ opérande : Donnée ou bien adresse de la donnée.

La taille d'une instruction peut varier, elle est généralement de quelques octets (1 à 8), elle dépend également de l'architecture du processeur.

● Instruction Addition :

Accumulateur = Accumulateur + Opérande

Correspond à l'instruction ADD A,#2

Instruction (16 bits)	
Code opératoire (5 bits)	Champ opérande (11 bits)
ADD A	#2
11001	000 0000 0010

Cette instruction est comprise par le processeur par le mot binaire :

11001 000 0000 0010 = code machine