

# Structure d'un ordinateur

---

# Structure d'un ordinateur

---

## *Plan de la phase*

*Introduction*

*Blocs fonctionnels*

*Etude de l'unité centrale de traitement*

*Fonctionnement de l'unité centrale*

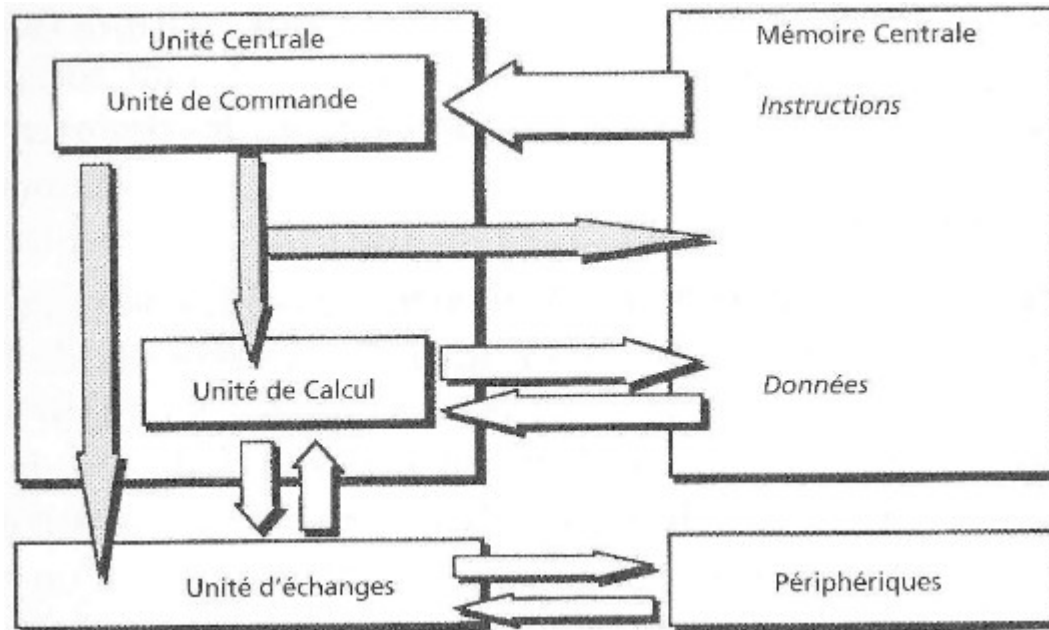
*Exercices*

# Structure d'un ordinateur

## *Introduction*

Un système informatique peut être représenté par 4 unités fonctionnelles :

- Une unité centrale
- La mémoire centrale
- L'unité d'échange
- Les périphériques



**Schéma fonctionnel d'un système informatique**

# Structure d'un ordinateur

## *Blocs fonctionnels – l'unité centrale*

### Unité centrale de traitement :

- microprocesseur ou
- microprocesseur + mémoire centrale

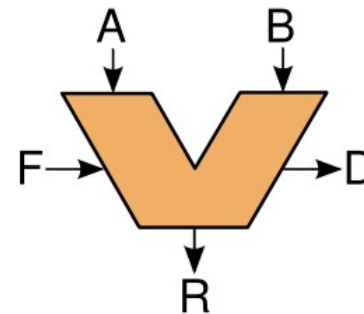


Boîtier central du système



Unité centrale

- a) l'unité de calcul (UAL) ou unité arithmétique et logique réalise :
- Les opérations arithmétiques (addition, multiplication...)
  - les opérations logiques
  - les comparaisons



- b) l'unité de commande :

- Gère le décodage et l'exécution des instructions
- En fonction des instructions, repartit les ordres vers les divers organes de la machine.
- Cherche l'instruction suivante.

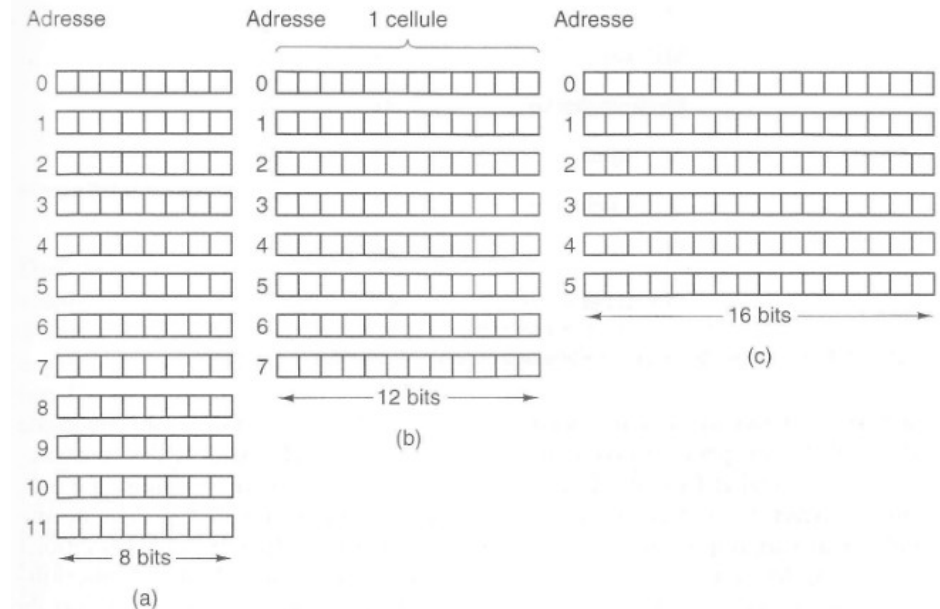
# Structure d'un ordinateur

## Blocs fonctionnels – la mémoire centrale

La **mémoire centrale** est composée d'un ensemble de **cellules** pouvant chacune **stocker des informations** qui auront toutes la même taille, le **mot mémoire**.

Les mots mémoires **représentent** soit des **instructions** ou des **données**.

Chaque cellule possède un numéro, **l'adresse**, qui permet aux programmes de la référencer.



Si une **adresse** s'exprime **sur m bits**, le nombre maximum de **cellules adressables** est donc de  $2^m$ .  
Sur le schéma :

- (a)  $m = 4$
- (b)  $m = 3$
- (c)  $m = 3$

Le nombre de bits de l'adresse est indépendant du nombre de bits présent dans chaque cellule.

# Structure d'un ordinateur

---

## *Blocs fonctionnels*

**L'unité d'échange** : gère les transferts d'informations entre l'uc et les périphériques

**Les périphériques :**

- D'entrée (écran...)
- De sortie (clavier...)
- Mémoire auxiliaire ou de masse (disque...)
- Spécialisés (sondes de température, de pression...)

# Structure d'un ordinateur

## *Etude de l'unité centrale de traitement - instruction*

Une instruction est une unité élémentaire d'un langage de programmation

### **Exemple :**

En C : `VAR = A + B;`

Toutes les instructions présentent deux types d'informations:

- ce qu'il faut faire (addition..)
- avec quelles données (A,B)

- L'instruction est transformée en un langage compréhensible par la machine (série binaire) :

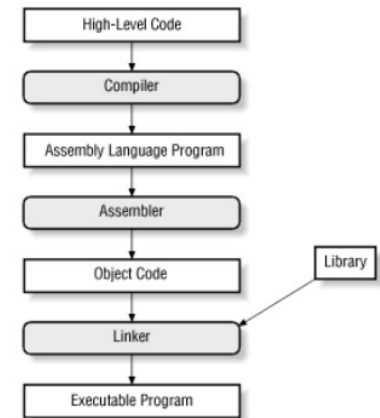
- **langage évolué** (C,Java,Perl...) : **interpréteur, compilateur.**
- **langage non évolué** ou langage d'assembleur : dépend du  $\mu P$ , **assembleur**

- Les données sont représentées sous forme ASCII,ISO 8859xx... ou représentées sous une forme virgule flottante, entier binaire....

Exemple : **ADD A,C** : instruction du langage d'assemblage va se traduire après assemblage par la suite binaire 1000 0001.

Quand l'unité reçoit une telle instruction, elle sait quel « travail » elle a faire et avec quelles données

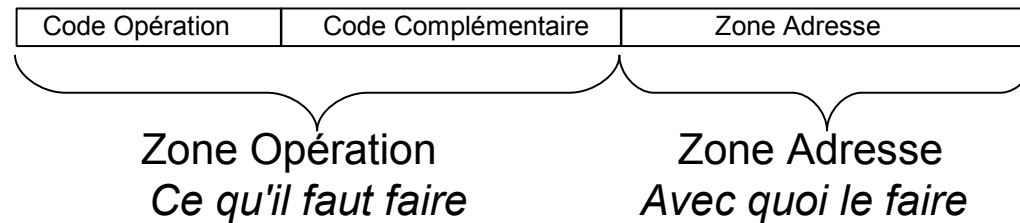
Elle déclenche une suite de **signaux de commandes** (microcommandes)



# Structure d'un ordinateur

## *Etude de l'unité centrale de traitement - instruction*

Une **instruction** peut se décomposer en **2 champs**



- **La zone opération** : indique quelle opération doit être réalisée.

Sera plus ou moins longue selon **le jeu d'instruction du  $\mu P$** .

Une zone opération sur un octet autorise 256 instructions.

Les **opérations** d'un **même type** possèdent le même **code opération** et sont dissociées par leur **code complémentaire**.

Pour une opération, on peut distinguer l'opération du contenu d'une zone mémoire avec elle-même et sur deux zones mémoires différentes.

- **La zone adresse** : Contiens la donnée à traiter.

Une **instruction** nécessite souvent le **traitement** de **deux données** et donc de connaître deux adresses. L'une de ces deux adresses est une **adresse implicite**, celle d'un registre particulier de l'UAL, l'**accumulateur**.

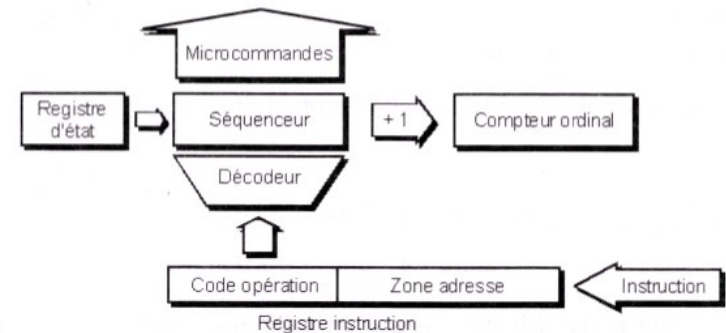


# Structure d'un ordinateur

## *Etude de l'unité centrale de traitement – unité de commande*

### Rôle :

- **décoder** les instructions en cours.
- **lancer les ordres** aux composants qui participent à la réalisation de l'instruction.
- **chercher** les instructions et recommencer le processus.



### Composants internes :

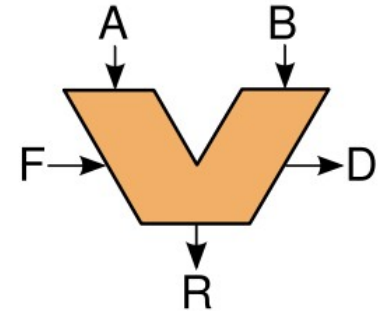
- **Le registre instruction** : charge l'instruction à traité.
- **Le séquenceur** : envoie une séquence de **microcommandes** aux composants impliqués par l'instruction en respectant une **chronologie** (séquencement) rythmée par une **horloge**.
- **Le registre d'état**, mémorise les informations :
  - sur l'état d'autres composants
  - sur les opérations qui ont déjà été exécutées (retenue préalable...)
- **Le compteur ordinal** (CO) :
  - au lancement du programme, le CO pointe sur la première instruction.
  - par la suite **pointe sur l'instruction suivante**

# Structure d'un ordinateur

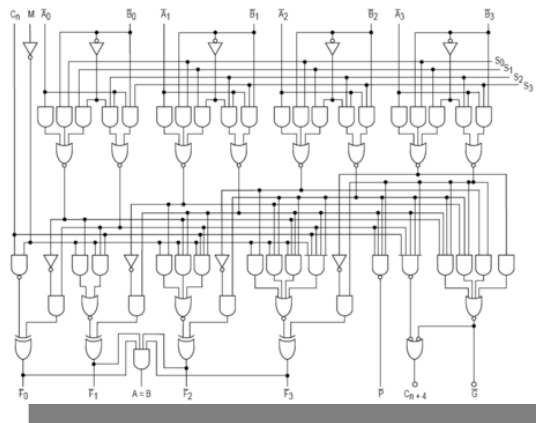
## *Etude de l'unité centrale de traitement – UAL*

L'**unité arithmétique et logique** est composé de circuits logiques :

- additionneur
- soustracteur
- comparateur logiques...



Celle-ci possède **deux entrées A et B** sur lesquelles on présentera les données à traiter. L'entrée **F** désignera l'**opération** à effectuer. Enfin, celle-ci possède **deux sorties, R** qui sera le **résultat** de l'opération et généralement **stocker** dans un registre dit **accumulateur**, et **D** les **drapeaux** qui indiqueront soit qu'il y a eu erreur (division par zéro, dépassement de capacité etc.), soit des codes conditions (supérieur, inférieur, égal à zéro etc.).



Le schéma de l'UAL 4 bits 74181.

# Structure d'un ordinateur

## *Etude de l'unité centrale de traitement – Bus*

Les **bus** permettent la **connexion** entre **L'UC** et **l'extérieur**.

**Le bus de données** : ce sont par ses lignes que **transitent** toutes les **données** (données et instructions) échangées entre les **composants** du système.

- **Largeur de bus** : 8, 16, 32 ou 64 bits.
- **Bus bidirectionnel** : les données peuvent circuler dans les deux sens.

**Le bus d'adresse** : destiné à **véhiculer** des **adresses**

- **Largeur de bus** détermine la taille de la **mémoire** qui sera **directement adressable**.

Un bus de 32 bits on peut adresser  $2^{32}$  (4Go) de mémoire physique

- **Bus unidirectionnel** : circulation des adresses que dans le sens unité centrale vers mémoire

**Le bus de commande** : permet aux **microcommandes générées** par le **séquenceur** de **circuler** vers les divers **composants** du système. Le bus de commande permet d'envoyer les requêtes associées avec l'envoi des données et des adresses dans les deux autres bus.

Par exemple, les requêtes de lecture ou d'écriture lors de transfert entre processeur et mémoire.

<http://www.arcanapercipio.com/cpu/bus/buspro.php#fsb>

# Structure d'un ordinateur

## Fonctionnement de l'unité centrale

Pour chaque **instructions** :

- Phase de **recherche** d'instructions
- Phase de **traitement** d'instructions

Étudions un exemple simplifié de fonctionnement :

Mnémonique = microcomande (correspond à une action élémentaire)  
 flèches = microcomande issue du séquenceur  
 Zone grise = passage information entre les composants

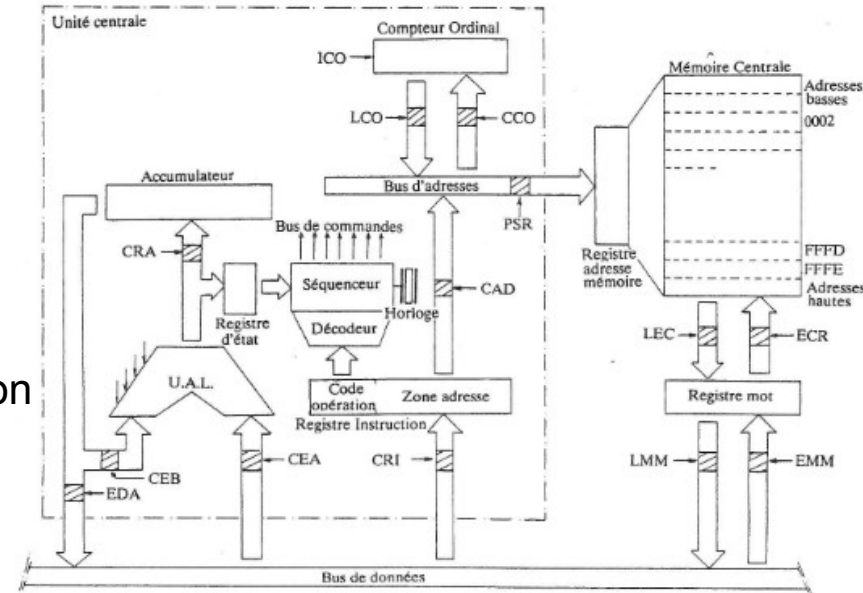


Schéma complet d'une unité centrale

**Exemple** : La microcomande LCO autorise le transfert du contenu du compteur ordinal sur le bus adresse.

# Structure d'un ordinateur

## *Fonctionnement de l'unité centrale – phase de recherche d'instruction*

Le programme est chargé en mémoire centrale, ou il occupe un certain nombre de cellules mémoires référencées par des adresses.

Le séquenceur, dans la phase de recherche, génère les microcommandes destinées à placer l'instruction dans le registre instruction.

Suite d'instruction :

- LCO : CO -> Bus d'adresse (première adresse du programme)
- PSR : Bus adresse -> Registre Adresse mémoire (décodeur, indique quel case mémoire lire)
- LEC : Mémoire -> Registre Mot (recopie de l'instruction)
- LMM : Registre Mot -> Bus de donnée (l'instruction se trouve sur le bus de donnée)
- CRI : Bus de donnée -> registre d'instruction (l'instruction se trouve dans le RI)  
l'instruction est décodée et exploitée par le séquenceur qui génère les microcommande nécessaire à son traitement.
- ICO : compteur ordinal +1 : une dernière microcomande provoque l'incrémentation du compteur ordinal qui pointe maintenant sur l'instruction suivante.

# Structure d'un ordinateur

## *Fonctionnement de l'unité centrale – phase de traitement de l'instruction*

Le **registre** d'instruction **soumet** au décodeur **l'instruction**.  
Le **séquenceur analyse** la zone opération et **génère** les **microcomandes** en fonction du code opération.

### **Exemple :**

soit un programme chargé de l'addition de deux nombres :

- 0x8 -> stocké en 0xF800
- 0x4 -> stocké en 0xF810
- Le résultat 0xC en 0xF820

• Étape 1 :

chargé 0x8 dans le registre l'accumulateur (A)

• Étape 2 : Faire l'addition du contenu de A avec la seconde donnée (0x4) et le résultat est remis dans A.

• Étape 3 :

Ranger le résultat (0xC) en mémoire à l'adresse 0xF820

En langage d'assemblage cela donne :

```
LD A,(F800H)
ADD A,(F810H)
LD (F820H),A
```

Le code machine généré →

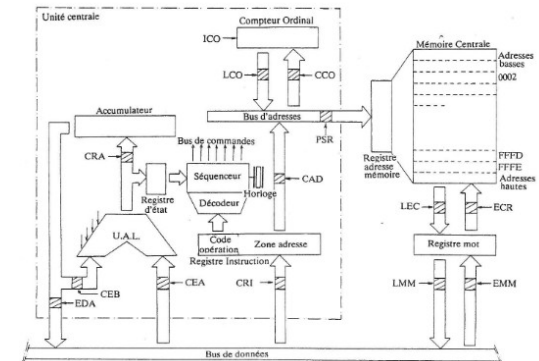
```
3A F8 00
C6 F8 10
32 F8 20
```

On stocke ses instructions en FB00H, FB01H et FB02H. (un mot devrait contenir la même quantité d'information, donc l'instruction 3A F8 00, devrait être en réalité stockée sur trois mots mémoires. Mais par souci de simplification, on stocke toute l'instruction dans un seul mot mémoire.



# Structure d'un ordinateur

### *Fonctionnement de l'unité centrale – phase de traitement de l'instruction*

[illegible]

LCO Lecture Compteur Ordinal	(Compteur Ordinal) -> Bus adresses
CCO Chargement Compteur Ordinal	(Bus adresses) -> Compteur Ordinal
PSR Pointage Sur Registre	(Bus d'adresses) -> Registre Adresse Mémoire
LEC LECTure	(Mémoire) -> Registre Mot
ECR ÉCRiture en mémoire	(Registre Mot) -> Mémoire
LMM Lecture Mot Mémoire	(Registre Mot) -> Bus de Données
EMM Écriture Mot Mémoire	(Bus de Données) -> Registre Mot
CAD Chargement Adresse	(Reg instr. adresse) -> Bus d'Adresses
CRA Chargement Registre Accumulateur	(UAL sortie) -> Accumulateur
CRI Chargement Registre Instruction	(Bus de Données) -> Registre Instruction
CEB Chargement Entrée B	(Accumulateur) -> Entrée B de l'UAL
CEA Chargement Entrée A	(Bus de Données) -> Entrée A de l'UAL
EDA Envoi de Donnée Accumulateur	(Accumulateur) -> Bus de données
ICO Incréméntation du Compteur Ordinal	(Compteur Ordinal) + 1
NOP No OPeration	la donnée passe de l'entrée A à la sortie sans opération
ADD Addition, SUB Soustraction, ET, OU logique, etc	

### Tableau des commandes générées par le programme d'addition



# Structure d'un ordinateur

## *Exercices*

Réaliser un tableau de fonctionnement pour le programme qui **soustrait** le nombre **3H**, rangé à l'adresse **F820H** du nombre **9H**, rangé à l'adresse **F810H** et range le **résultat** à l'adresse **F820H**.  
Considérons pour l'exercice que les instructions en pseudo-assembleur et leur équivalent en langage machine sont :

```
LD A,(F800H)
ADD A,(F810H)
LD (F820H),A
```

Le code machine généré

```
3A F8 00
C6 F8 10
32 F8 20
```