

# NFP 136 - Architecture des ordinateurs

Amélie Lambert

2017-2018

# Plan du cours

- 1 Rôle d'un ordinateur
- 2 Les composants de l'ordinateurs
- 3 Principe d'exécution des instructions
- 4 Les interruptions matérielles
- 5 Gestion des entrées/sorties

## Bibliographie :

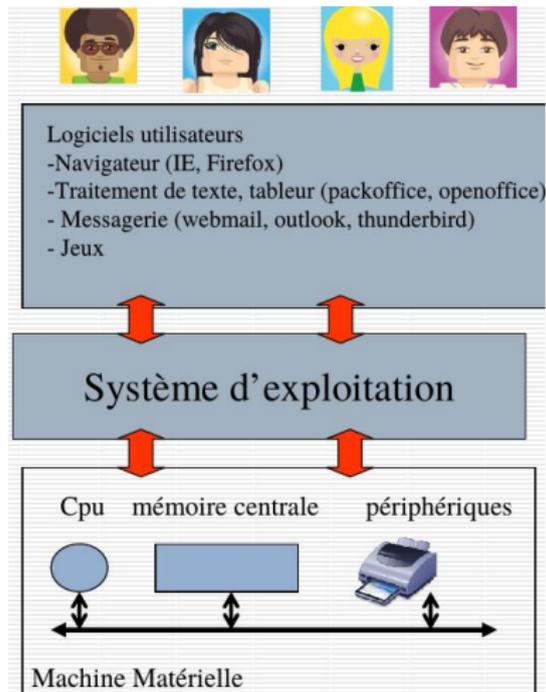
- DELACROIX J. Cours CNAM NFA 003, NFA 004
- CAZES A. DELACROIX J. Architecture des machines et des Systèmes informatiques Dunod 2003

# 1. Rôle d'un ordinateur

# Les différents niveaux de la machine informatique

Une machine informatique est composée de trois couches :

- Le matériel **hardware** (la machine physique : processeur, mémoire centrale, périphériques, l'ensemble communiquant par un bus)
- Le **logiciel de système d'exploitation** (ensemble de programmes qui se place à l'interface entre le matériel et les logiciels applicatifs)
- Les logiciels des utilisateurs **software** (programmes qui permettent à l'utilisateur de réaliser des tâches sur la machine).



# Le rôle d'un ordinateur

- **Traitement automatique d'informations**, par exécution d'un programme (description du traitement) sur des données (des informations) pour fournir des résultats (de nouvelles informations).
- **Fonctions fondamentales d'un ordinateur :**
  - ▶ Traitement des informations
  - ▶ Mémorisation des informations
  - ▶ Transmission des informations
- **Deux aspects :**
  - ① L'expression du problème à résoudre, de sa solution dans un langage compréhensible par l'ordinateur
  - ② La structure de l'ordinateur qui doit être composée d'éléments permettant le stockage, le traitement, la lecture ou l'écriture des données.

# Rôle d'un ordinateur

- 1 J'ai un problème à résoudre : Calculer le périmètre d'un rectangle de longueur  $a$  et de largeur  $b$ .
- 2 J'écris un algorithme :  
Perimetre :=  $2a + 2b$
- 3 Grâce à un langage de programmation, je code la solution pour la faire exécuter par l'ordinateur  $\implies$  Programme constitué d'instruction :

```
public static double perimetre (double a, double b){  
    double perimetre;  
    perimetre = 2*a +2*b;  
    return perimetre;  
}
```

# Le codage d'un problème

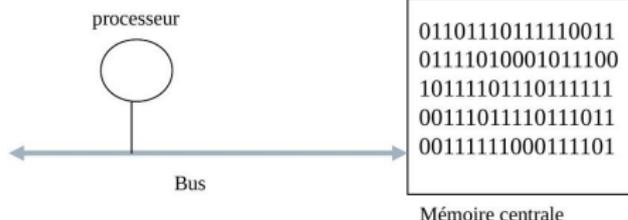
```
public static double perimetre (double a, double b){  
    double perimetre;  
    perimetre = 2*a +2*b;  
    return perimetre;  
}
```

Programme en langage  
de haut niveau : instruc-  
tions de haut niveau

---

## Compilateur

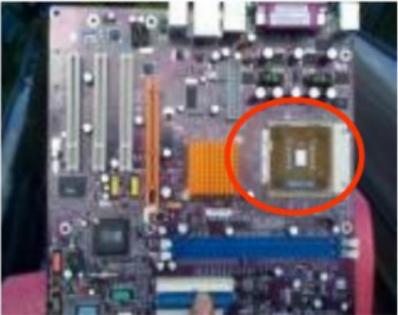
---



Programme à exécuter :  
instructions machine et  
valeurs en binaire

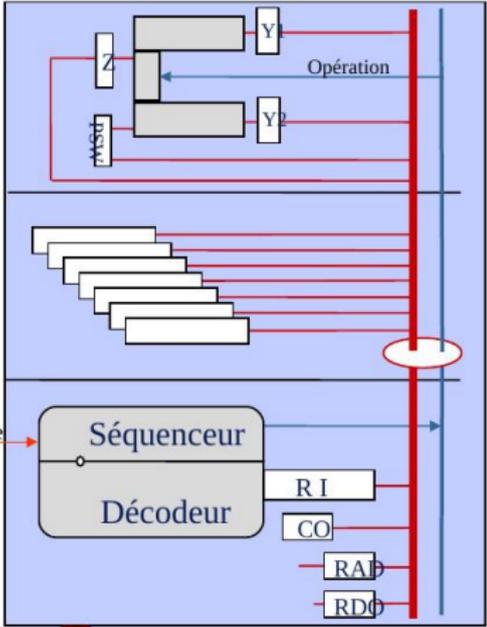
## 2. Les composants de l'ordinateur

# Les composants de l'ordinateur (1/2)

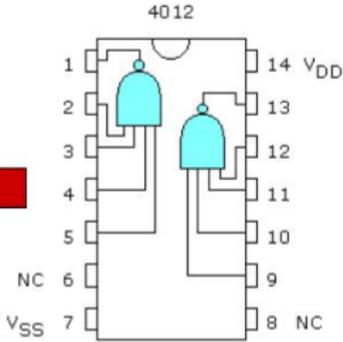
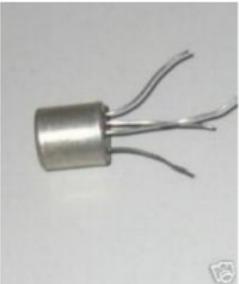


Unité Arithmétique et logique

Registres



TRANSISTORS

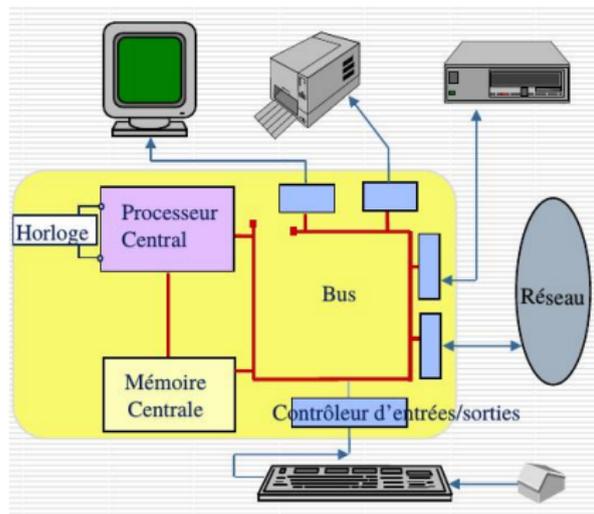


CIRCUITS LOGIQUES OU INTEGRES (PORTES)

## Les composants de l'ordinateur (2/2)

Les fonctions de l'ordinateur sont de permettre d'effectuer des calculs, de stocker des données et de communiquer. Pour cela, l'ordinateur a les composants physiques suivants :

- Des éléments permettant la communication entre l'ordinateur et les hommes : **les périphériques**
- Un élément permettant d'exécuter les instructions d'un programme : **le processeur**
- Des éléments permettant de stocker les données : **les mémoires**
- Des éléments permettant aux composants de communiquer : **les bus**



# Le processeur

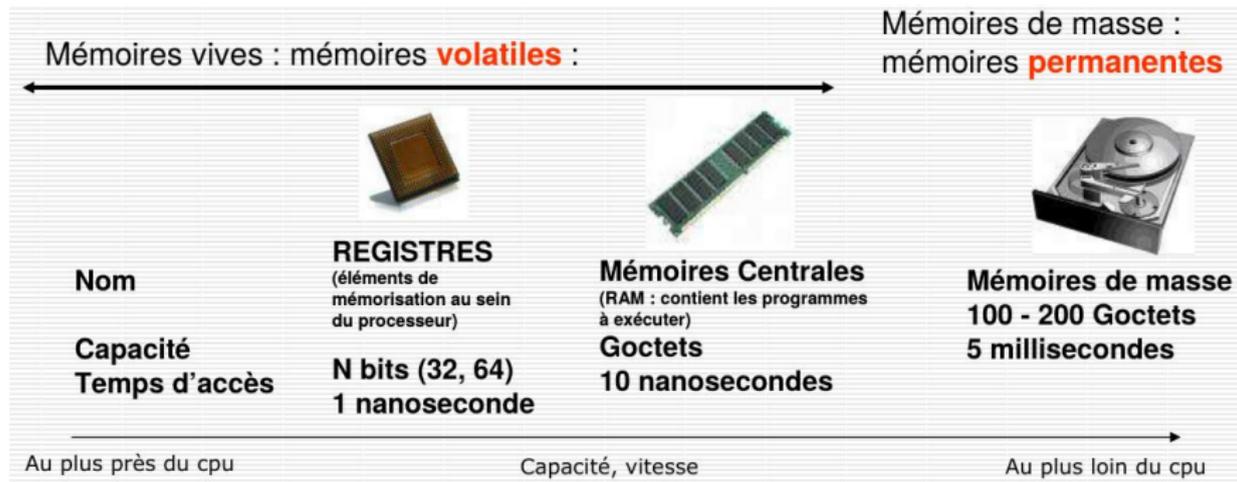
- **Le processeur** (CPU pour *Central Processing Unit*) est le cerveau de l'ordinateur. Il permet de manipuler les données et les instructions codées sous formes binaires.
- C'est un circuit électronique cadencé au rythme d'une horloge interne qui envoie des impulsions. La **fréquence d'horloge** est le nombre d'impulsions par seconde, son unité est le Hertz.  
*Exemple* : Un ordinateur à 200MHz a une horloge qui envoie 200 000 000 impulsions par seconde.

# Les mémoires de l'ordinateur (1/2)

- Une **mémoire** est un composant électronique capable de stocker temporairement des informations.
- Une **mémoire** est caractérisée par :
  - ▶ Sa **capacité** représentant le volume global d'information (en bits) que la mémoire peut stocker.
  - ▶ Son **temps d'accès**, correspondant à l'intervalle de temps entre la demande de lecture/écriture et la disponibilité de la donnée.
- L'ordinateur possède différents niveaux de mémoire organisés selon une **hiérarchie mémoire**

## Les mémoires de l'ordinateur (2/2)

L'ordinateur possède différents niveaux de mémoire organisés selon une **hiérarchie mémoire** :



- **Mémoire volatile** : le contenu de la mémoire n'existe que si il y a une alimentation électrique.
- **Mémoire permanente, de masse** : mémoire de grande capacité dont le contenu demeure même sans alimentation électrique

# Les Périphériques

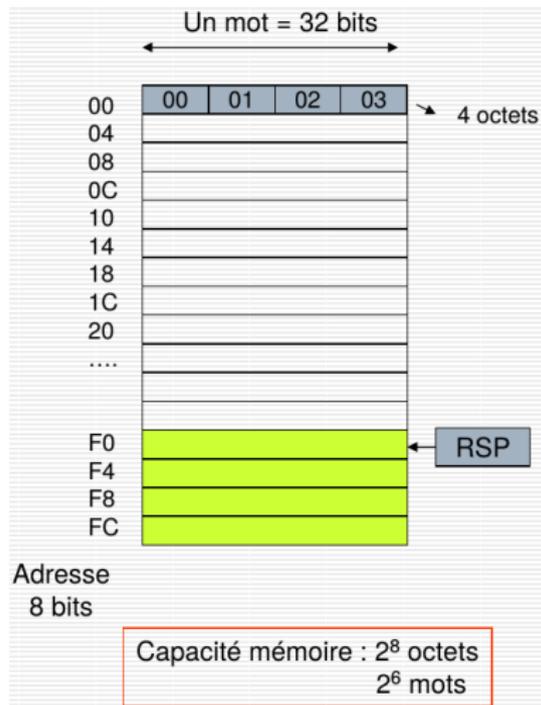
Un **périphérique** : est un matériel électronique pouvant être raccordé à un ordinateur via une **interface entrée/sortie** comme un connecteur. Cette interface est pilotée par un **driver**.

On distingue plusieurs catégories de périphériques :

- **Périphériques de sortie** permettant à l'ordinateur d'émettre des informations vers l'extérieur (écran, imprimante,...)
- **Périphériques d'entrée** permettant uniquement d'envoyer des informations à l'ordinateur (souris, clavier,...)
- **Périphériques d'entrée-sortie** permettant à la fois d'envoyer et de recevoir des informations (disque dur, modem,...)
- **Périphériques de stockage** permettant de stocker des données de manière permanente (disque dur, DVD,...)

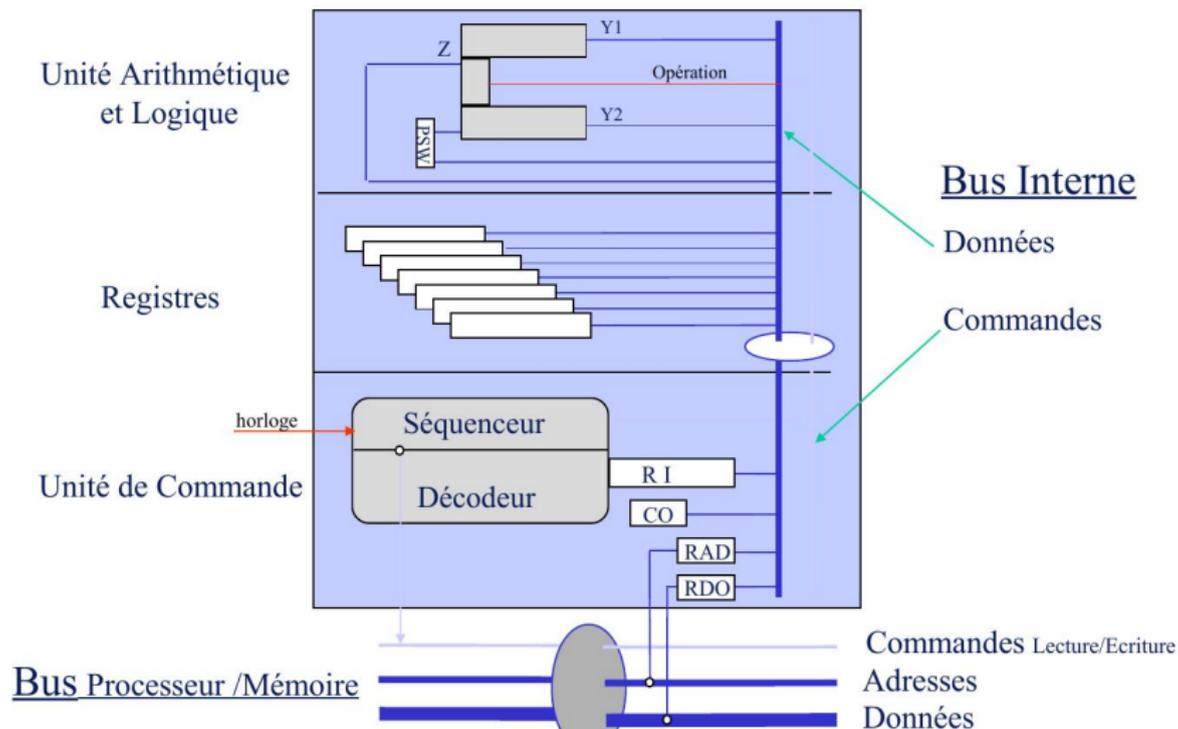
# 3. Principe d'exécution des instructions

# La mémoire centrale

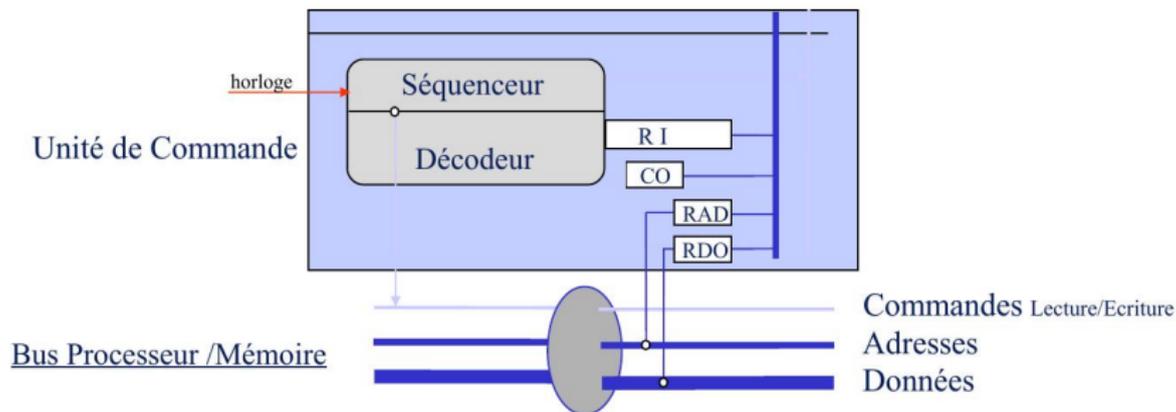


- La mémoire centrale est constituée d'un ensemble de **mots** mémoires.
- Elle contient les instructions et les données des programmes à exécuter.
- Un mot est constitué par un ensemble d'octets, et chaque octet est repéré de manière unique par une adresse.
- Elle s'interface avec le CPU via le bus CPU-mémoire.
- Une zone particulière de la mémoire est gérée comme une structure de pile et est repéré par le registre RSP du processeur.

# La structure du Processeur (1/4)



## La structure du Processeur (2/4)

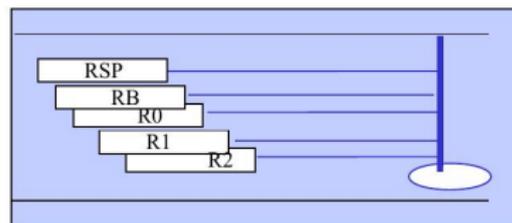


L'unité de commande est chargée de la reconnaissance des instructions et de leur exécution par l'unité de traitement au rythme de l'horloge.

Elle contient les registres :

- **RI** (registre instruction) : contient l'instruction en cours d'exécution
- **CO** (compteur ordinal) : contient l'adresse en MC de la prochaine instruction
- **RAD** (registre adresse) et **RDO** (registre de données) : registres d'interfaçage avec la mémoire centrale

## La structure du Processeur (3/4)

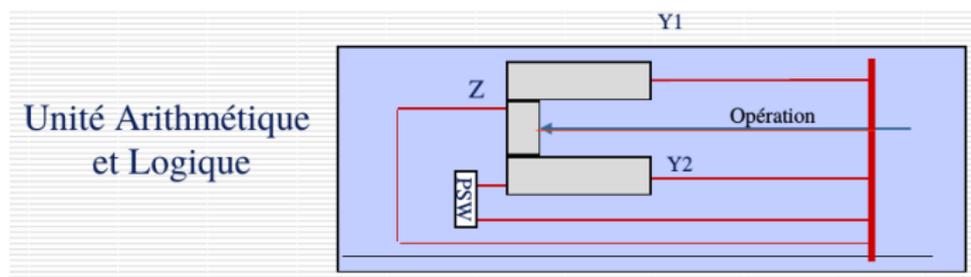


- Le registre est l'entité de base manipulée par le processeur.
- Aucune opération n'est directement réalisée sur les cellules mémoires.

Les Registres sont :

- les registres généraux **R0**, **R1**, **R2**
- le registre de pile **RSP** (Register Stack Pointer)
- les registres d'adressage : **RB** (registre de base)

# La structure du Processeur (4/4)



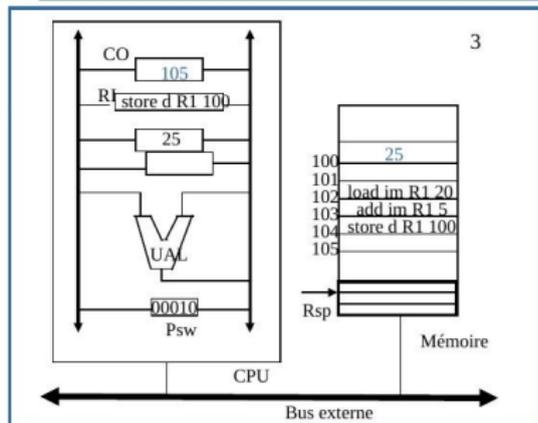
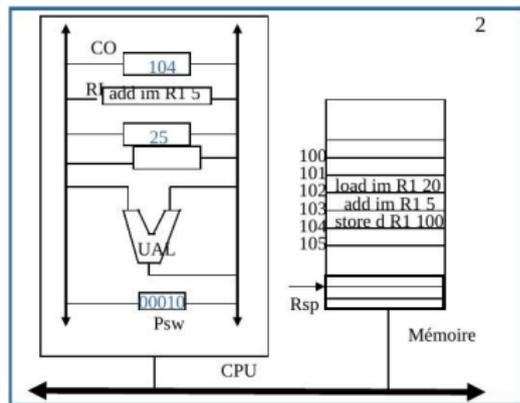
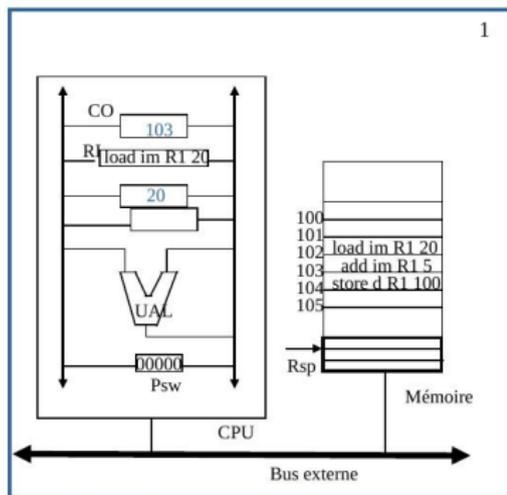
L'**unité Arithmétique et logique** constitue l'unité d'exécution du processeur. Elle est composée de :

- l'ensemble des circuits permettant de réaliser les opérations arithmétiques (addition, multiplication, division, ...) et les opérations logiques (complément à 2, inverse, OU, ET, ...) sur les opérandes **Y1** et **Y2**.
- un registre d'état **PSW** qui contient des indicateurs positionnés par le résultat **Z** des opérations effectuées :
  - ▶ **Z** : positionné à 1 si résultat opération nul, 0 sinon
  - ▶ **S** : positionné à 0 si résultat opération positif, 1 sinon
  - ▶ ...

# Exécution d'un programme

- Le programme composé d'instructions machine et de données a été chargé en mémoire centrale par un outil **CHARGEUR**
- Le compteur ordinal **CO** est chargé avec l'adresse en Mémoire centrale du mot contenant la première instruction du programme
- L'exécution du programme s'effectue instruction par instruction, sous le pilotage de l'unité de commande du processeur.
- Le traitement d'une instruction par le processeur se découpe en trois étapes :
  - ① **FETCH** : l'instruction est lue en mémoire centrale et copiée dans le registre RI du processeur
  - ② **DECODAGE** : l'instruction est reconnue par l'unité de décodage
  - ③ **EXECUTION** : l'opération correspondant à l'instruction est réalisée

# Exécution d'un programme : Exemple



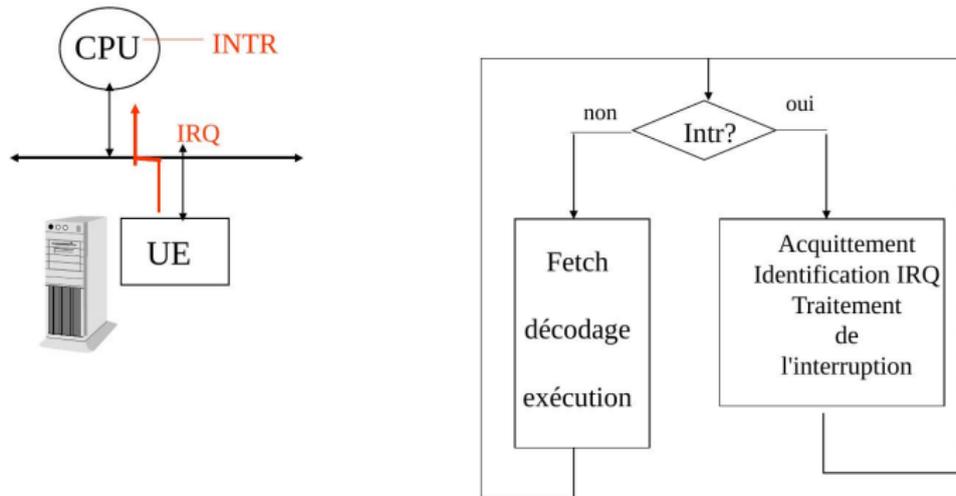
- ✓ CO : compteur ordinal
- ✓ PSW : registre d'état
- ✓ RI : registre instruction

## 4. Les interruptions matérielles

## Notion d'interruption

- Une **interruption** est un mécanisme permettant de stopper l'exécution du programme en cours afin d'aller exécuter une tâche jugée plus prioritaire.
- Elle évite au processeur de **scruter** tous les périphériques.
- Elle est caractérisée par un numéro et un traitement associé (la routine ou traitant d'interruption).
- On distingue principalement deux types d'événements :
  - ① Les interruptions externes ou matérielles sont émises par les périphériques du processeur (fin d'écriture disques, plus de papier imprimante...). Ce sont les **IRQs**.
  - ② Les interruptions internes ou logicielles sont émises par le processeur lui-même lorsqu'il rencontre une erreur dans l'exécution du programme (division par zéro, accès mémoire illégal). Ce sont les **trappes**.

# Le mécanisme des interruptions (1/3)

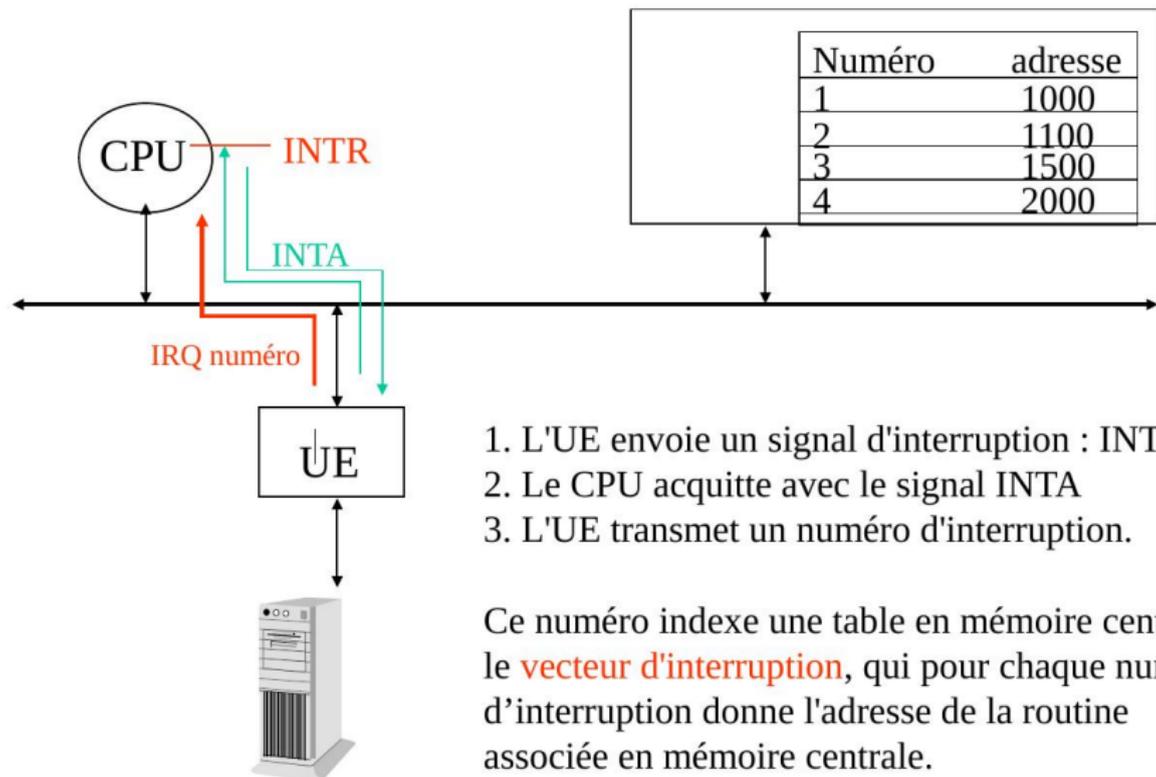


Un périphérique signale un événement au processeur en émettant une interruption matérielle.

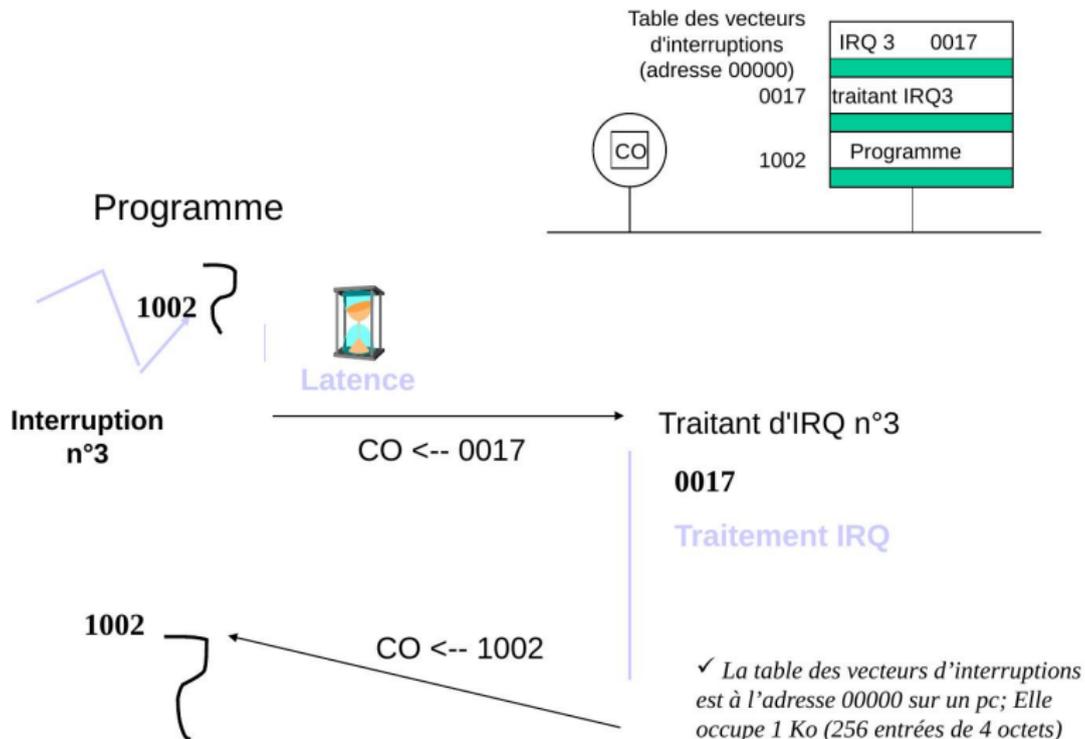
Le processeur teste sa ligne d'entrée d'interruption (INTR) avant de commencer le traitement de l'instruction suivante du programme qu'il exécute.

## Le mécanisme des interruptions (2/3)

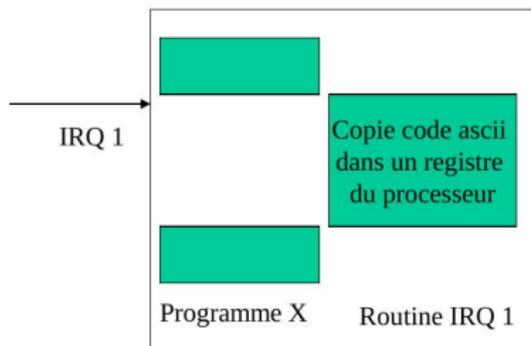
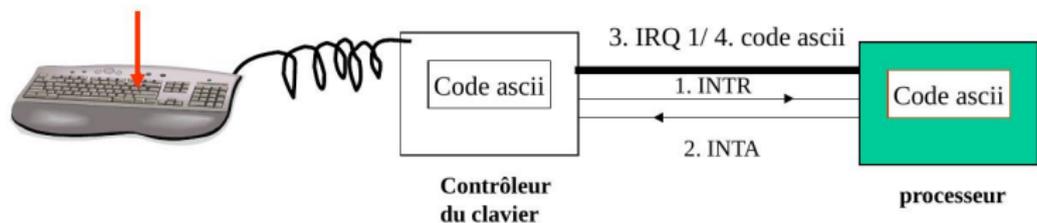
### Mémoire centrale



# Le mécanisme des interruptions (3/3)



# Le mécanisme des interruptions : Exemple



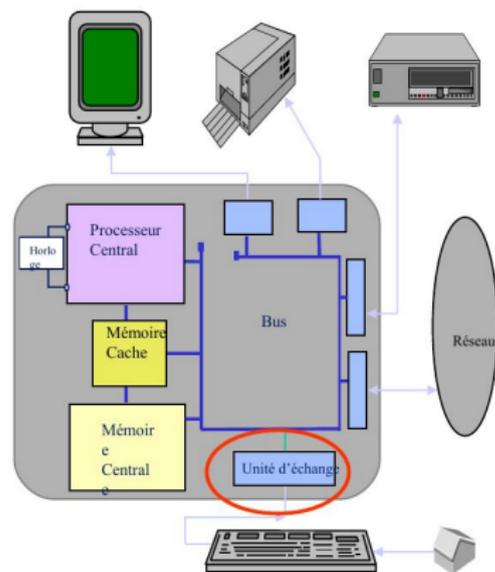
Exécutions du CPU

# 5. Gestion des entrées sorties

# Périphériques de l'ordinateur

- Un **périphérique** est un matériel électronique pouvant être raccordé à un ordinateur par l'intermédiaire de l'une de ses interfaces d'entrée-sortie.
- On distingue habituellement les catégories de périphériques suivantes :
  - ▶ **périphériques de sortie** permettant à l'ordinateur d'émettre des informations vers l'extérieur (un écran, une imprimante,...)
  - ▶ **périphériques de stockage** capable de stocker les informations de manière permanente (disque dur, clé usb, DVD,...)
  - ▶ **périphériques d'entrée** capables uniquement d'envoyer des informations à l'ordinateur (la souris, le clavier,...)

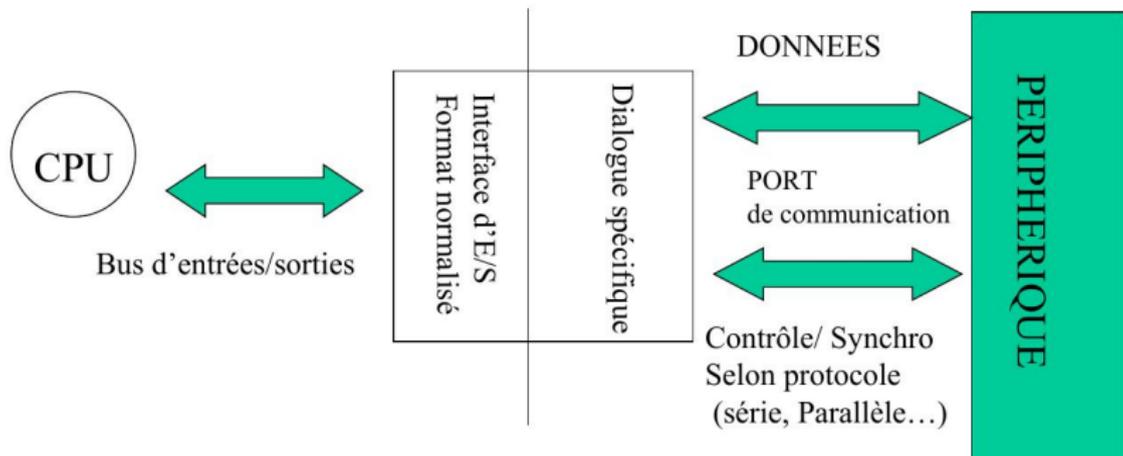
## L'unité d'échange (1/2)



### Rôle de l'unité d'échange :

- connexion des unités périphériques au bus
- gestion des échanges entre le processeur et les périphériques

## L'unité d'échange (2/2)

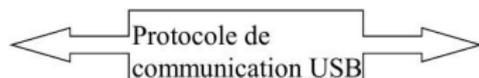


Interface  
d'entrées-sorties  
USB (contrôleur)

Connecteur USB



✓ Port de communication :  
Voies en entrées et sorties entre UE  
et périphériques



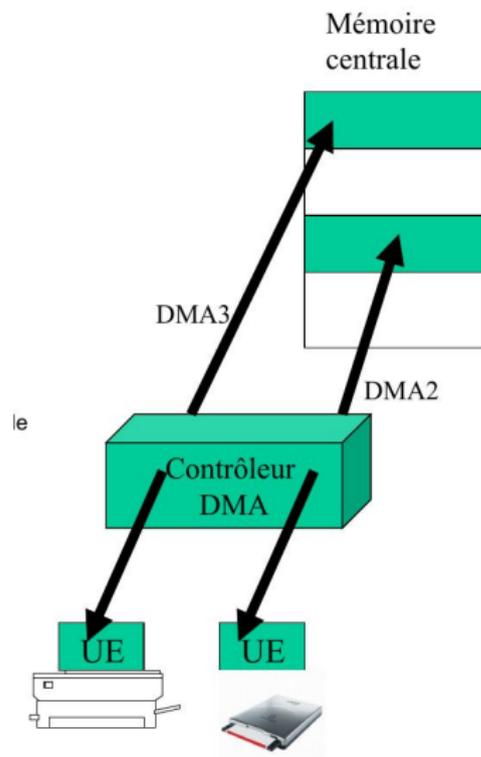
## Le Direct Memory Access (1/3)

Par défaut, une unité d'échange ne sait pas accéder à la mémoire centrale :

- Le processeur doit lire chaque mot depuis la mémoire centrale et le placer dans le registre de données de l'unité d'échange (ou vice versa)
- Il faut permettre à L'UE d'accéder à la mémoire centrale sans recourir au processeur : le **DMA (DIRECT MEMORY ACCESS)**

Le DMA est un contrôleur d'entrées-sorties qui permet aux unités d'échanges d'accéder à la mémoire centrale sans intervention du processeur.

## Le Direct Memory Access (2/3)



- Le DMA gère les dispositifs de contrôleur de disquettes, contrôleurs de disques durs, La carte son, Le port parallèle (imprimante) entre autres.
- Chaque dispositif utilise un canal DMA identifié par une adresse de début et une adresse de fin en mémoire centrale qui désigne la zone mémoire accessible pour le transfert.
- Le contrôleur DMA d'un PC gère 8 canaux DMA.

## Le Direct Memory Access (3/3)

L'algorithme est le suivant :

- 1 le processeur (pilote) initialise le DMA
- 2 le DMA effectue l'entrée/sortie ;
- 3 le DMA signale la fin d'entrée-sortie par une IRQ