

Introduction à l'architecture des ordinateurs

- ❑ Rôle et composants
- ❑ Les mémoires de l'ordinateur
- ❑ Principes d'exécution des instructions
- ❑ Le mécanisme des interruptions

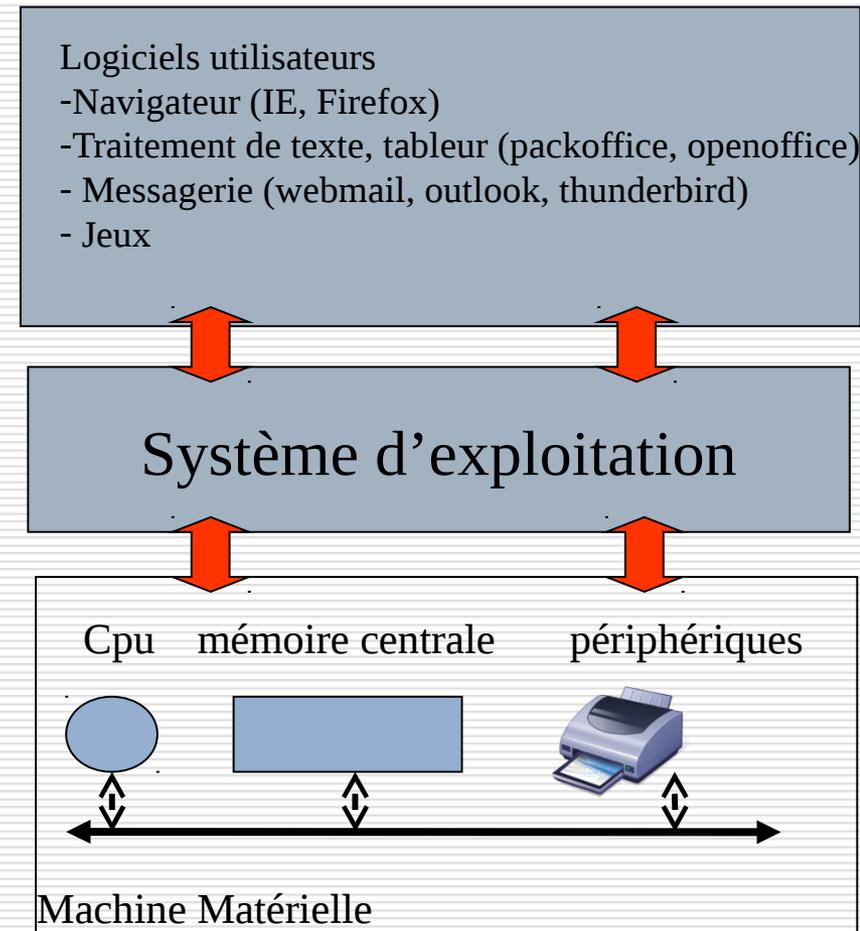
Rôle et composants

Les différents niveaux de la machine informatique



On distingue généralement trois couches dans la composition d'une machine informatique :

- **Le matériel « hardware »** : il correspond à la machine physique, notamment composée du processeur, de la mémoire centrale et des périphériques, l'ensemble communiquant par un bus.
- Le **logiciel de système d'exploitation** : c'est un ensemble de programmes qui se place à l'interface entre le matériel et les logiciels applicatifs. Il permet notamment à ces logiciels applicatifs d'utiliser les ressources matérielles de la machine. Les principaux OS (*Operating System*) sont notamment
- Les **logiciels des utilisateurs « software »** : ce sont des programmes qui permettent à l'utilisateur de réaliser des tâches sur la machine.



Le rôle d'un ordinateur

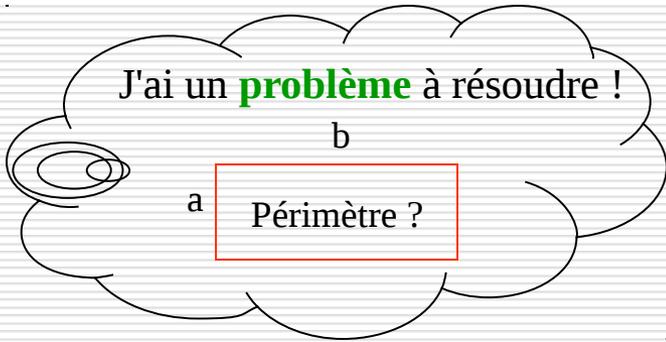
- Exécuter un programme de traitement sur des données en vue de résoudre un problème.

- Pour cela il faut :
 - L'expression du problème à résoudre, de sa solution dans un **langage compréhensible par l'ordinateur**

 - Que la **structure de l'ordinateur** soit composée d'éléments permettant le stockage, le traitement , et la lecture ou l'écriture des données

Le rôle d'un ordinateur

1.



2.



J'écris un ALGORITHME

Périmètre := $2a + 2b$

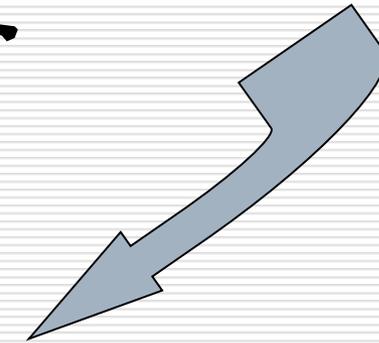


3.

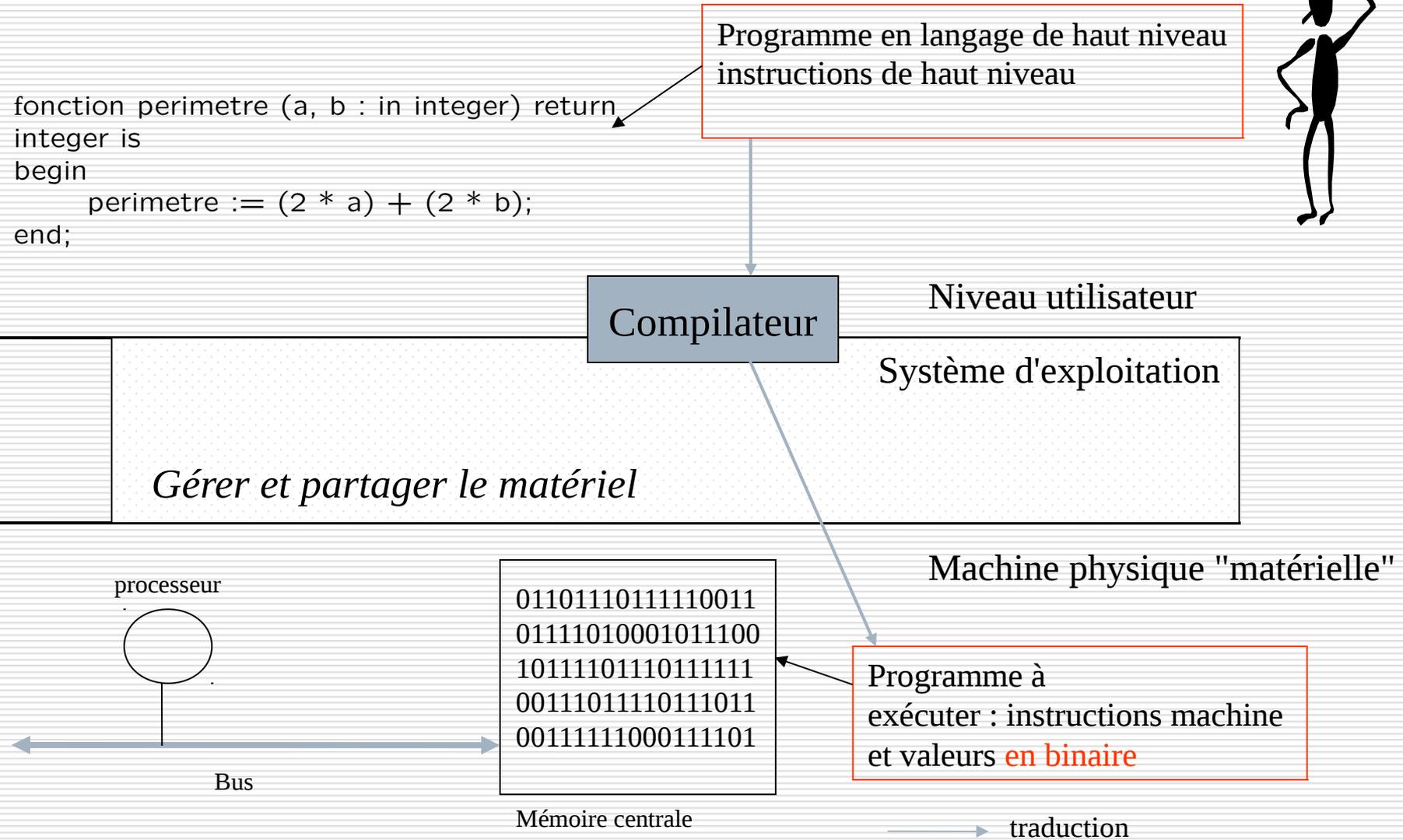
En utilisant un langage de programmation, je code l'algorithme pour le faire exécuter par l'ordinateur

→ PROGRAMME constitué d'instructions

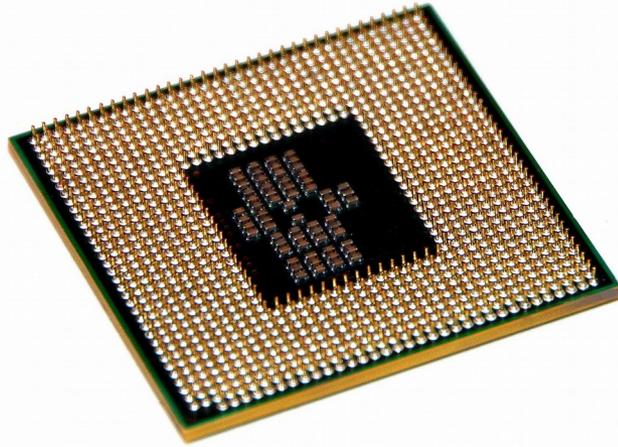
```
fonction perimetre (a, b : in integer) return integer is
begin
    perimetre := (2 * a) + (2 * b);
end;
```



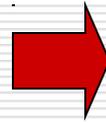
Le codage d'un problème ...



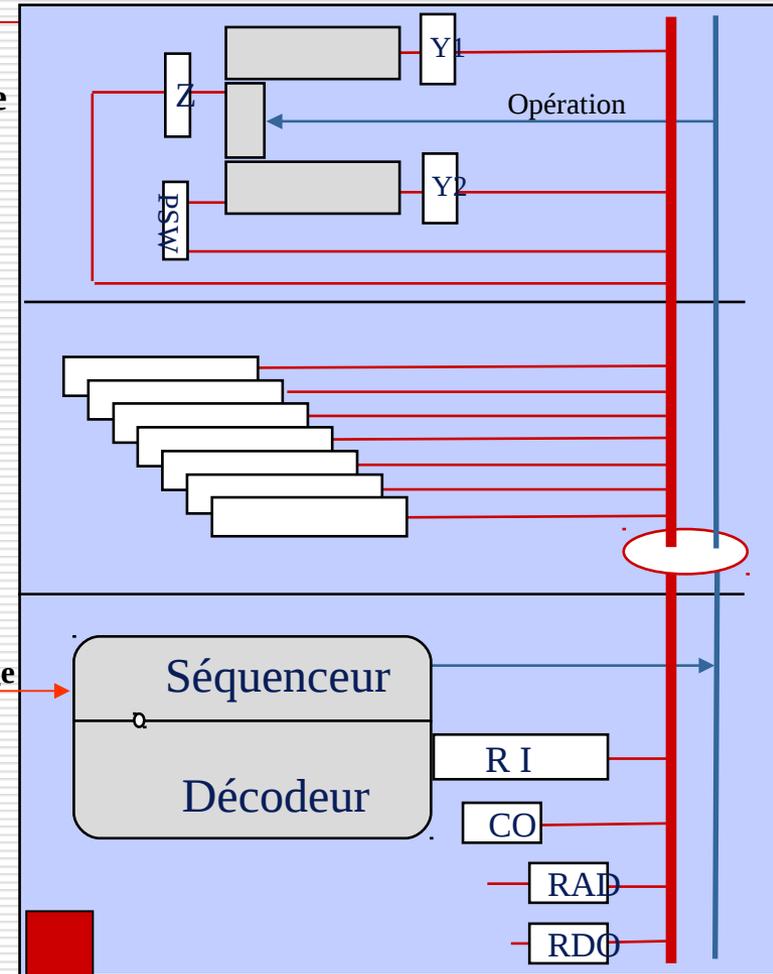
Composants de l'ordinateur



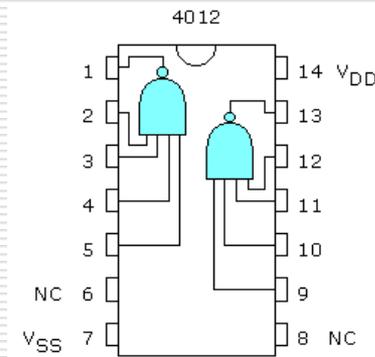
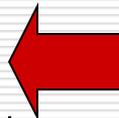
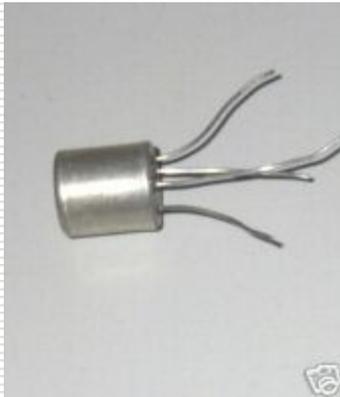
Unité Arithmétique et logique



Registres



TRANSISTORS

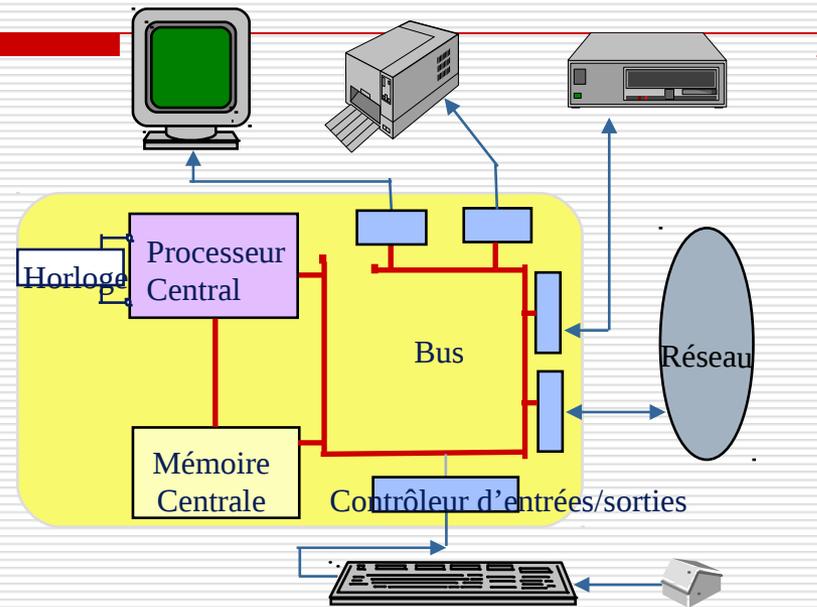


CIRCUITS LOGIQUES OU INTEGRES (PORTES)

Les fonctions et les composants

Les fonctions de l'ordinateur sont de permettre à des utilisateurs (via des logiciels informatiques) de :

- Effectuer du calcul;
- Stocker des données;
- Communiquer.

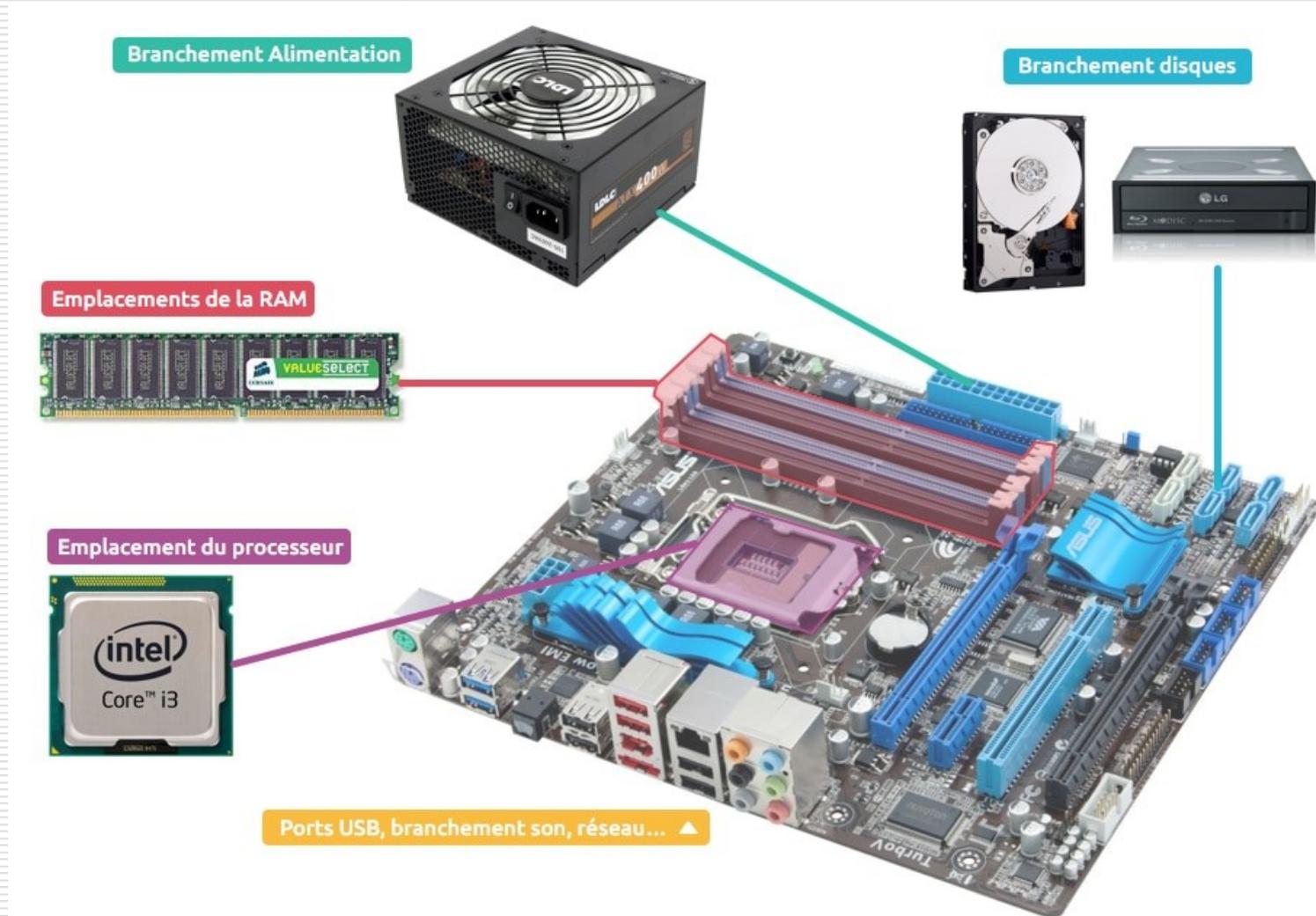


Pour cela, l'ordinateur est doté d'un ensemble de composants physiques

- Des éléments permettant la communication entre l'ordinateur et l'être humain : ce sont les **périphériques**.
- Un élément permettant d'exécuter les instructions d'un programme : c'est le **processeur** (CPU).
- Des éléments permettant de stocker les données : ce sont les **mémoires** de l'ordinateur.
- Des éléments permettant aux différents composants (périphériques, processeur, mémoire) de l'ordinateur de communiquer : ce sont les **bus** de l'ordinateur

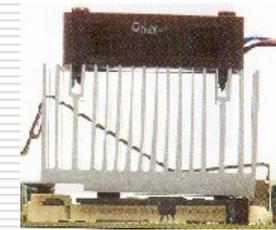
La carte mère de l'ordinateur

La **carte mère** est le socle permettant la connexion de l'ensemble des éléments essentiels



Le processeur

- Le **processeur** (**CPU**, pour *Central Processing Unit*) est le cerveau de l'ordinateur. Il permet de manipuler, des données et des instructions codées sous forme binaires.
- Le processeur est un circuit électronique cadencé au rythme d'une horloge interne qui envoie des impulsions, appelées « **top** ». La **fréquence d'horloge**, correspond nombre d'impulsions par seconde. Elle s'exprime en Hertz (Hz).



Il est composé de millions de transistors placés dans un boîtier, qu'on appelle **circuit intégré** ou **puce**

Il est surmonté d'un refroidisseur.

Exemple : Ordinateur à 200 MHz

→ l'horloge envoie 200 000 000 de battements par seconde.

Les 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** (HDMI, USB, ...), le plus souvent par l'intermédiaire d'un **connecteur**. L'interface d'entrées-sorties est pilotée par un **driver (pilote d'entrées-sorties)**



- On distingue habituellement les catégories de périphériques suivantes :



- **périphériques de sortie** : ils permettent à l'ordinateur d'émettre des informations vers l'extérieur (écran, imprimante, ...)



- **périphériques de stockage** : ils stockent des informations de manière permanente (disque dur, clé USB, ...)



- **périphériques d'entrée** : ils envoient des informations à l'ordinateur (souris, clavier, ...)

- **périphériques d'entrée-sortie** : ils sont à la fois d'entrée et de sortie (le disque dur, ...)

Les mémoires de l'ordinateur

Les mémoires de l'ordinateur

- Une « **mémoire** » est un composant électronique capable de stocker temporairement des informations.

- Une mémoire est caractérisée par :
 - Sa **capacité**, représentant le volume global d'informations
 - Son **temps d'accès**, correspondant à l'intervalle de temps entre la demande de lecture/écriture et la disponibilité de la donnée.

- L'ordinateur contient différents niveaux de mémoire, organisés selon une **hiérarchie mémoire**.

L'unité de la mémoire : le bit ou l'octet

- La donnée de base manipulée par la machine physique est le **bit** (*Binary Digit*) qui ne peut prendre que deux valeurs : 0 et 1
- Ces 0 et 1 correspondent aux deux niveaux de voltage (0-1 et 2-5 volts) admis pour les signaux électriques issus des transistors qui constituent les circuits physiques de la machine.

Nombre de combinaisons représentables, en binaire, par rapport au nombre de bits :

1 bit	0, 1	2^1 états
2 bits	00, 01, 10, 11	2^2 états
3 bits	000,001,010,011,100,101,110, 111	2^3 états
n bits		2^n états

L'unité de la mémoire : le bit ou l'octet

Les Kilo, Mega et Giga octets

1 octet = 8 bits (byte)	Avant 1998	Après 1998
Kilooctet (Ko)	2^{10} octets = 1024 octets	1000 octets
Mégaoctet (Mo)	2^{20} octets = 1024 Koctets	1000 Koctets
Gigaoctet (Go)	2^{30} octets = 1024 Moctets	1000 Moctets

Dans le cadre de NFA003, nous considérerons les valeurs d'avant 1998

Les catégories de mémoires

- Mémoire volatile : le contenu de la mémoire n'existe que si il y a une alimentation électrique (typiquement les mémoires caches et mémoire centrale)
- Mémoire permanente, de masse : mémoire de grande capacité dont le contenu demeure même sans alimentation électrique (typiquement le disque dur, clé USB, carte mémoire, CD, DVD)

La hiérarchie des mémoires de l'ordinateur

Mémoires vives : mémoires **volatiles**

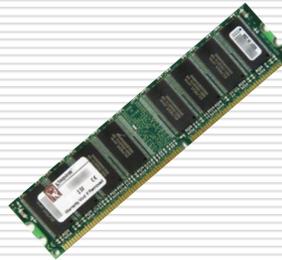
Mémoires de masse :
mémoires **permanentes**



REGISTRES
N bits (32, 64)
1 nanoseconde



Mémoires Caches
Kilo octets
5 nanosecondes



Mémoires Centrales
Giga octets
10 nanosecondes



Mémoires de masse
Tera octet
5 millisecondes

Au plus près du cpu

Capacité, vitesse

Au plus loin du cpu

Les mémoires vives ou RAM (*Random Access Memory*)

- C'est une mémoire accessible en lecture et écriture
- C'est une mémoire volatile interne
- Elle compose la mémoire centrale et les caches
- Deux structures :
 - La DRAM - Dynamic RAM (tps d'accès : 60 nano secondes)
 - La SRAM - Static RAM (tps d'accès 5 nano secondes)

Les types de mémoire : mémoires vives

DRAM (*Dynamic Random Access Memory*)



- ◆ Elle compose la mémoire centrale
- ◆ Barrette DIMM (*Dual Inline Memory Module*)
- ◆ Temps d'accès : 60 nano secondes
- ◆ Peu coûteuse

Fonctionnement : 1 cellule mémoire mémorise un bit et est constituée par un transistor et un condensateur

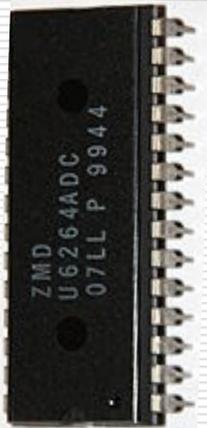
le condensateur se décharge dans le temps. Il convient de recharger chaque cellule périodiquement (1000 fois / s) : le rafraichissement de la mémoire.

Les types de mémoire : mémoires vives

SRAM (*Static Random Access Memory*)

- ◆ Elle compose les mémoires caches (cache CPU et MMU)
- ◆ Temps d'accès : 10 nano secondes
- ◆ Plus coûteuse et plus encombrante que la DRAM

Fonctionnement : 1 cellule mémoire mémorise un bit et est constituée par 4 à 6 transistors (*circuit de type bascule*)



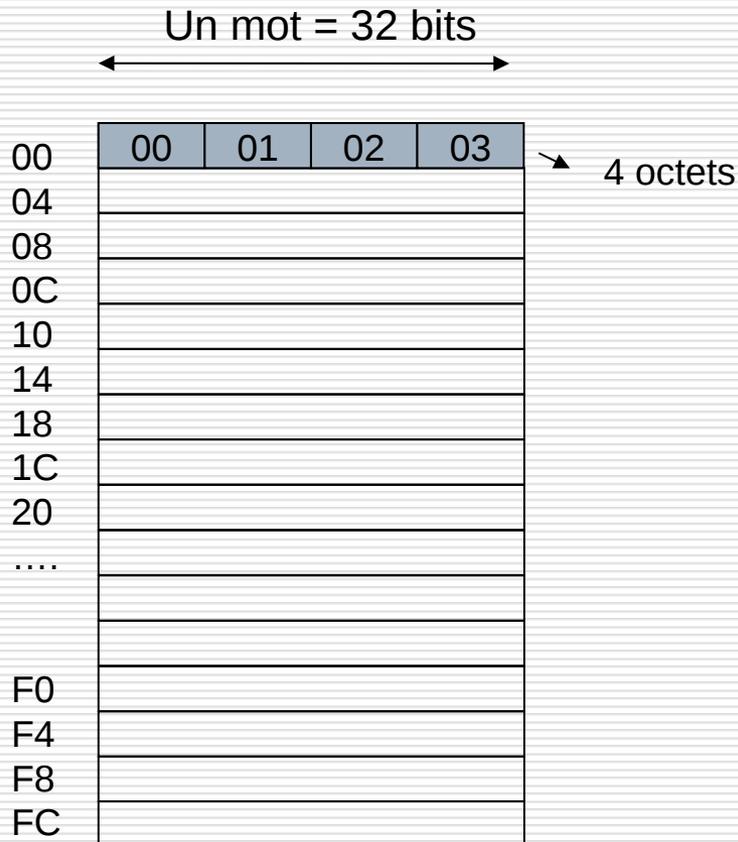
Mémoires mortes : ROM (*Read Only Memory*)

- Mémoire accessible en **lecture** (150 ns)
- Mémoire **non volatile interne**.
- ***Une fois l'information enregistrée, celle-ci ne peut pas (ou difficilement) être modifiée.***

Mémoires flash : compromis entre les deux types de mémoire :

- Mémoire accessible en **lecture et écriture**
- Mémoire **non volatile**
- Temps d'accès plus important que la RAM

La mémoire centrale

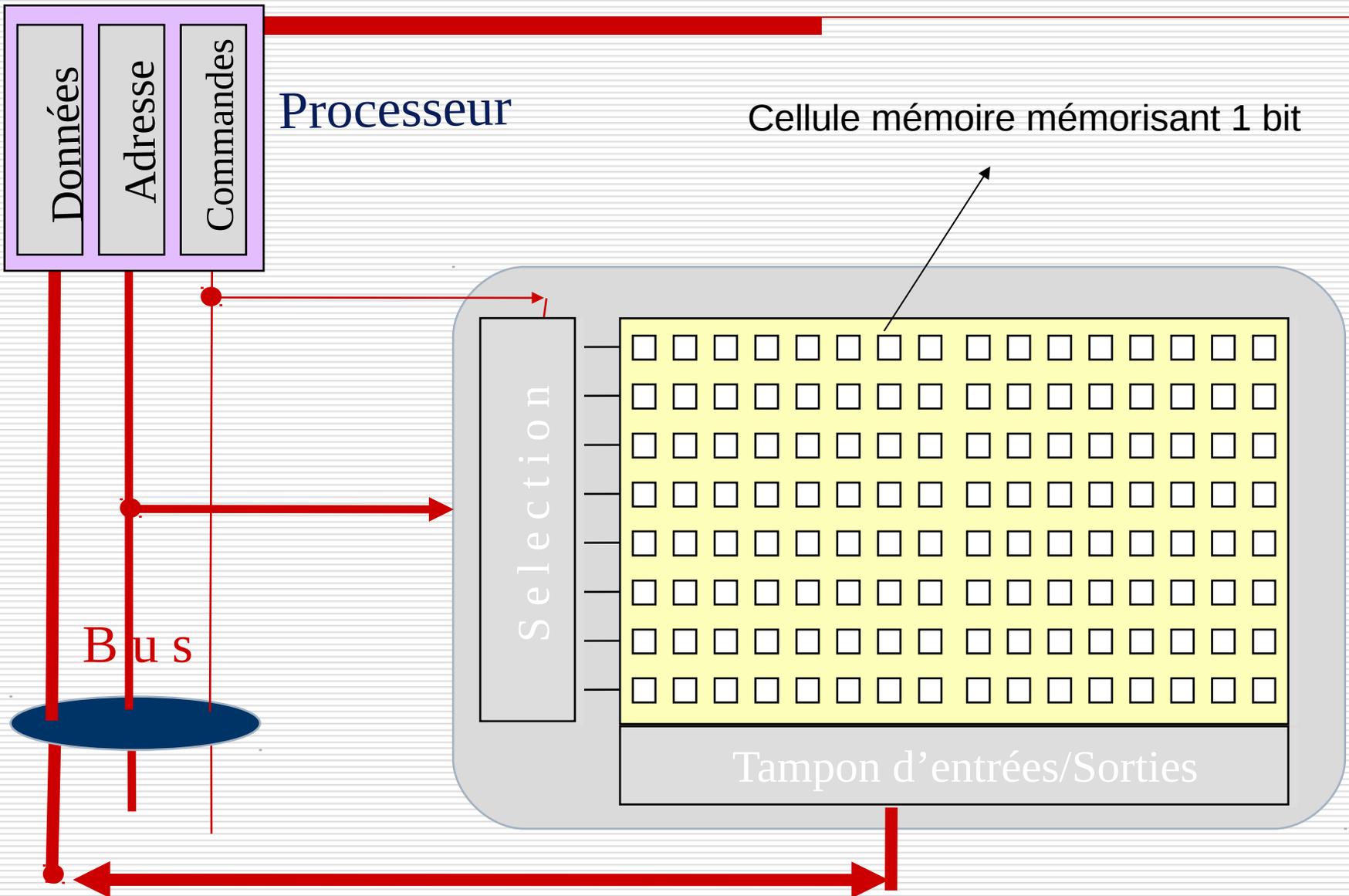


- La mémoire centrale est constituée par un ensemble de **mots** mémoire (32 bits).
- Elle contient les instructions et données des programmes à exécuter
- Un mot est constitué par un ensemble d'octets; chaque octet est repéré de manière unique par une **adresse**.
- Elle s'interface avec le cpu via le bus mémoire-cpu.

Adresse 8 bits

Capacité mémoire : 2^8 octets 2^6 mots

Adressage de la mémoire centrale



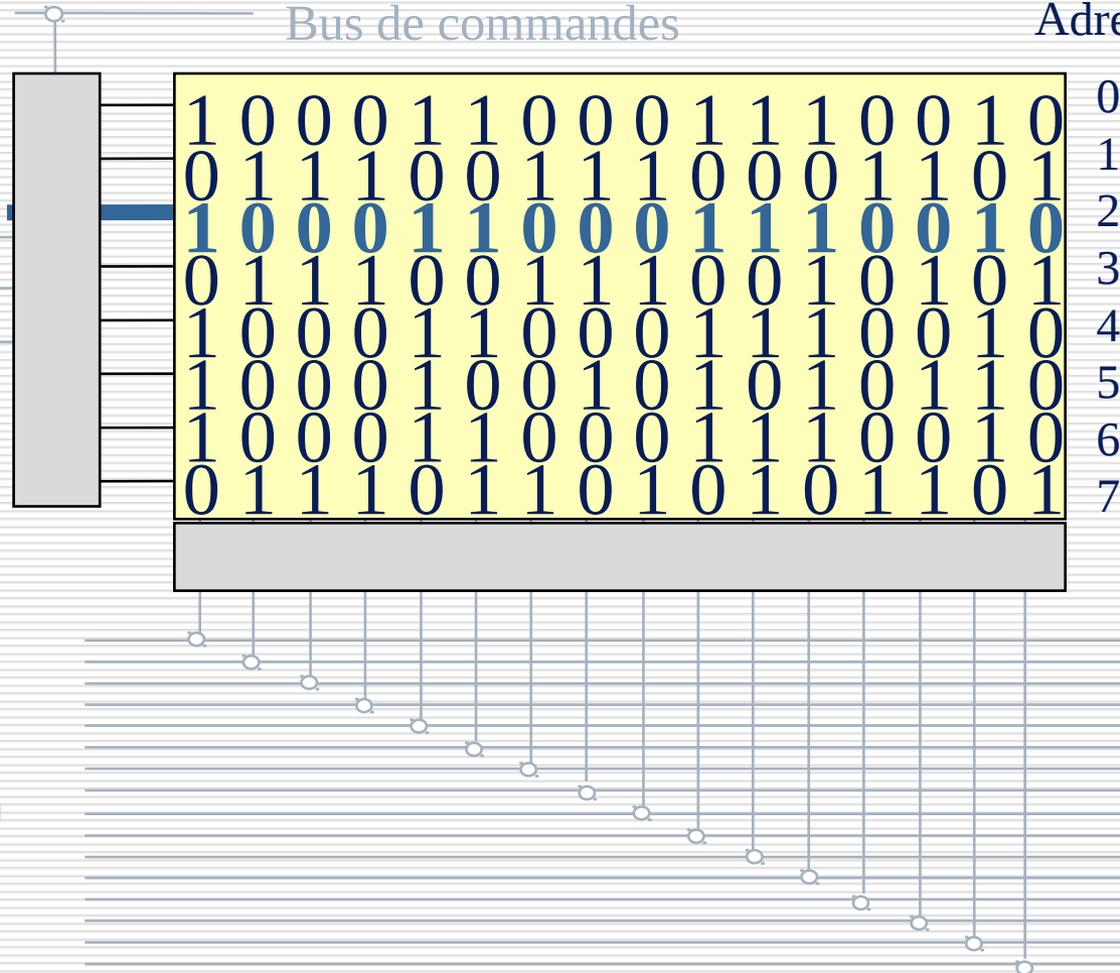
Mémoire centrale : écriture

Bus adresse

0 1 0

Bus de commandes

Adresses mémoire



1- Sélection

2- Bus(données)

3- Ecriture

Bus données

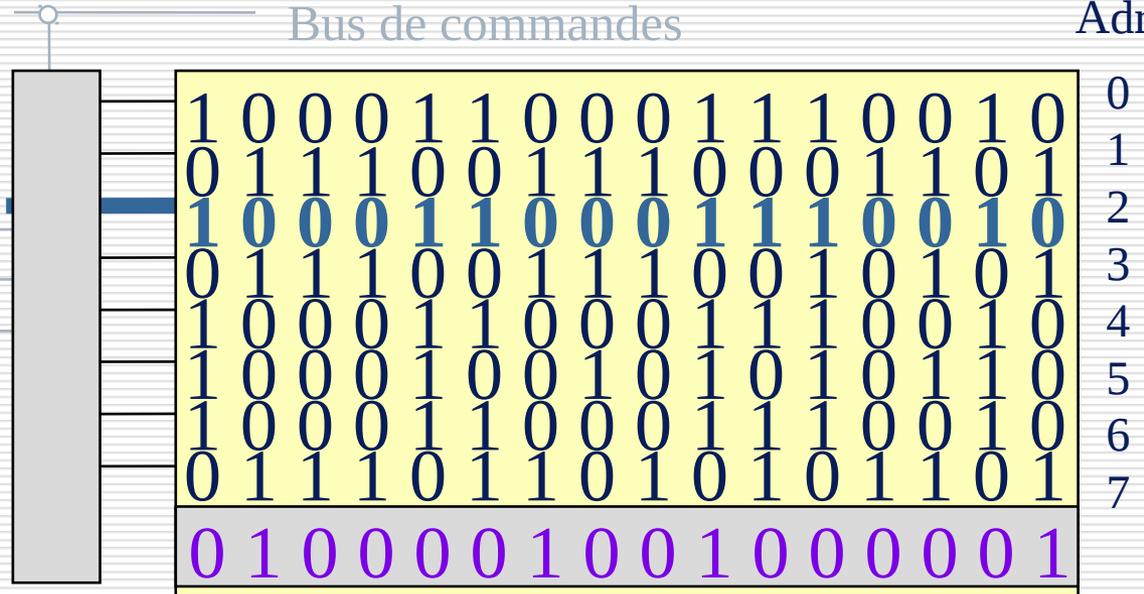
Mémoire centrale : écriture

Bus adresse

0 1 0

Bus de commandes

Adresses mémoire



0
1
2
3
4
5
6
7

- 1- Sélection
- 2- Bus (données)
- 3- Ecriture

Bus données

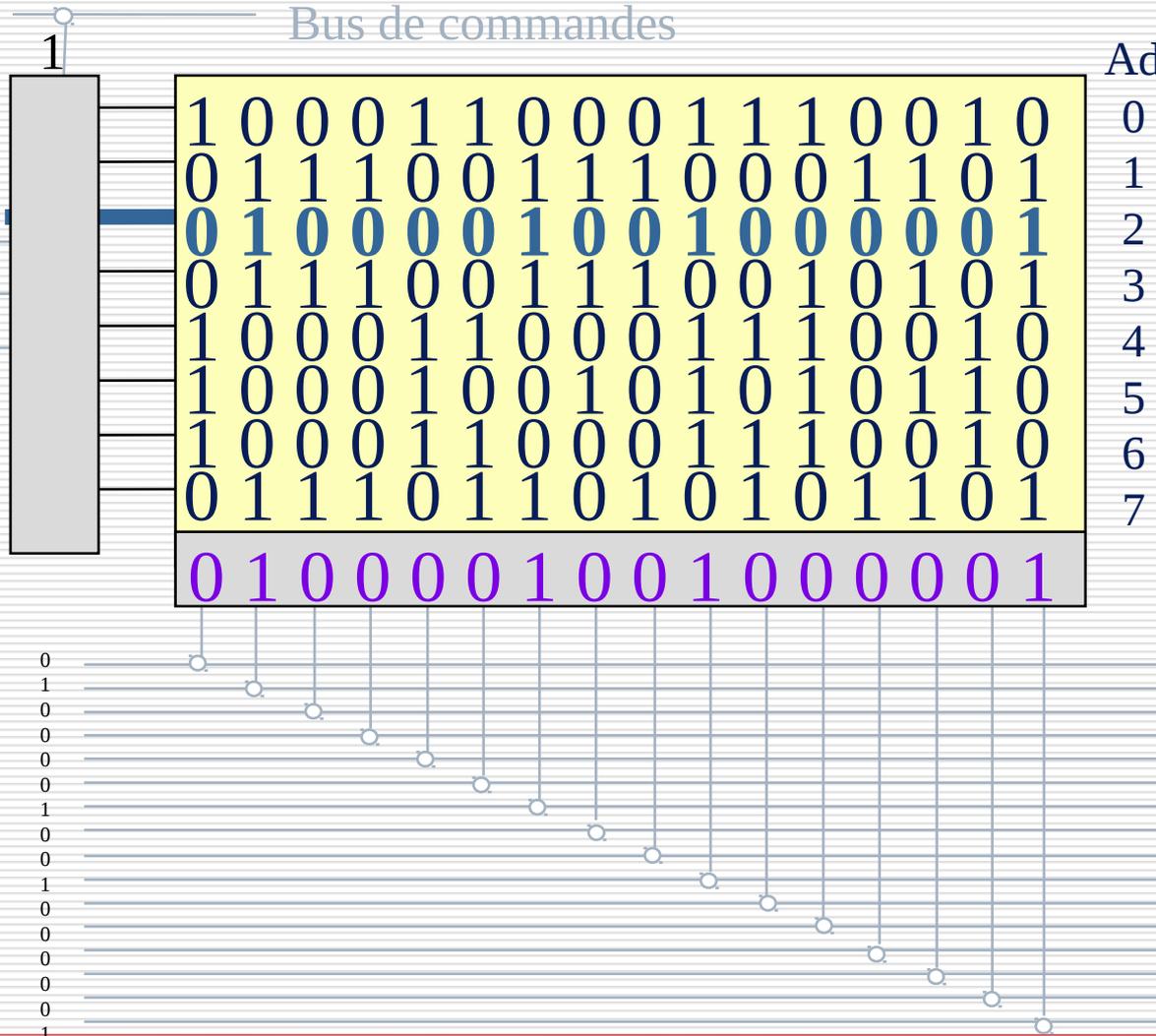
Mémoire centrale : écriture

Bus adresse

0 1 0

Bus de commandes

Adresses mémoire



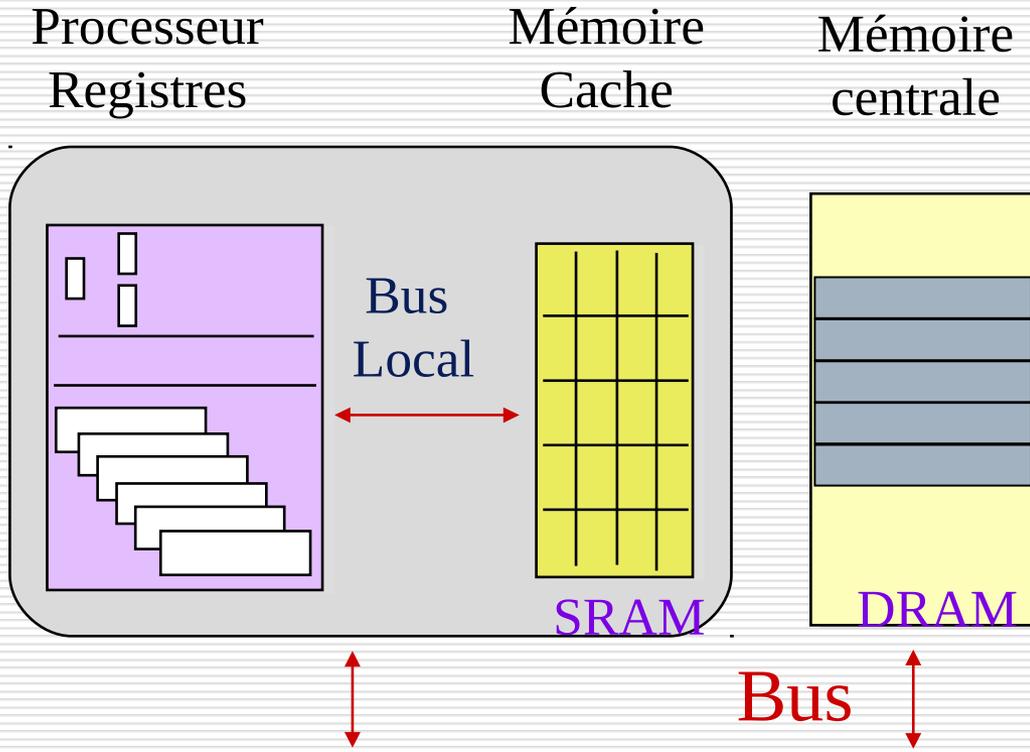
- 1- Sélection
- 2-BUS(données)
- 3- Ecriture

La hiérarchie des mémoires : exemple le cache

vitesse → 6-35 ns 70 - 120 ns

cout ← Le plus élevé Le moins élevé

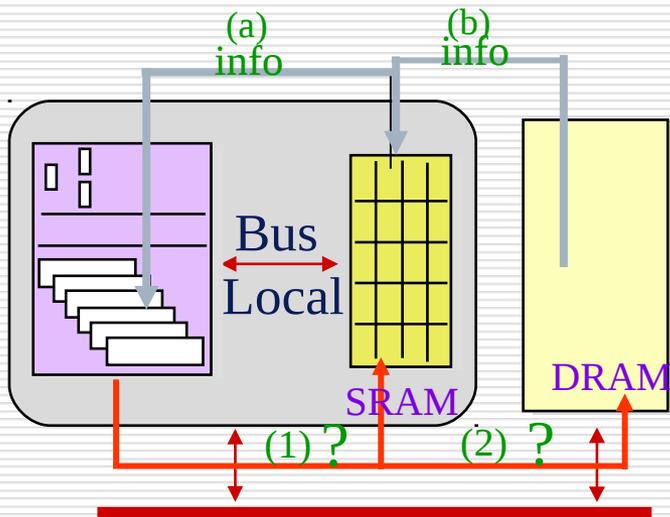
capacité ← La plus petite La plus grande



La mémoire cache est une mémoire intermédiaire placée entre le processeur et la mémoire centrale

dont le temps d'accès est de 4 à 20 fois inférieur à celui de la mémoire centrale.

Elle comporte un nombre fini d'entrées (n mots mémoire)

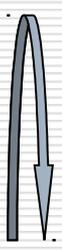


- (1) L'info cherchée est-elle dans le cache ?
OUI / **Succès (a)** : ramener l'info dans le CPU
NON / **Défaut (2)** : chercher l'info dans la MC
- (2) L'info est-elle en MC ?
OUI / **Succès (b)** :
ramener l'info dans le cache
puis dans le CPU (a)
NON / Défaut

Localité temporelle : si une donnée d'adresse A est accédée à un temps t, la probabilité qu'elle soit de nouveau accédée aux temps t+1, t+2 est très forte.

→ La donnée est remontée dans le cache pour minimiser les temps d'accès suivants

I1	load Im R1 5
I2	loop : add Im R2 3
I3	add Im R1 -1
I4	JMPZ Fin
I5	JMP Loop
I6	fin : store D R2 10



Localité temporelle

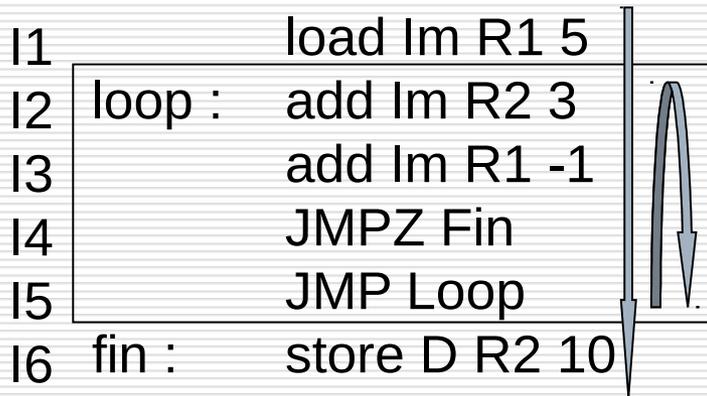
Premier accès aux instructions

I2, I3, I4, I5 → en MC

Les 4 accès suivants s'effectuent à partir du cache

Localité spatiale : si une donnée d'adresse A est accédée à un temps t, la probabilité que les données d'adresses voisines soient accédées aux temps t+1, t+2 est très forte.

→ La donnée d'adresse A et **également les données d'adresse voisines** sont remontées dans le cache pour minimiser les temps d'accès suivants



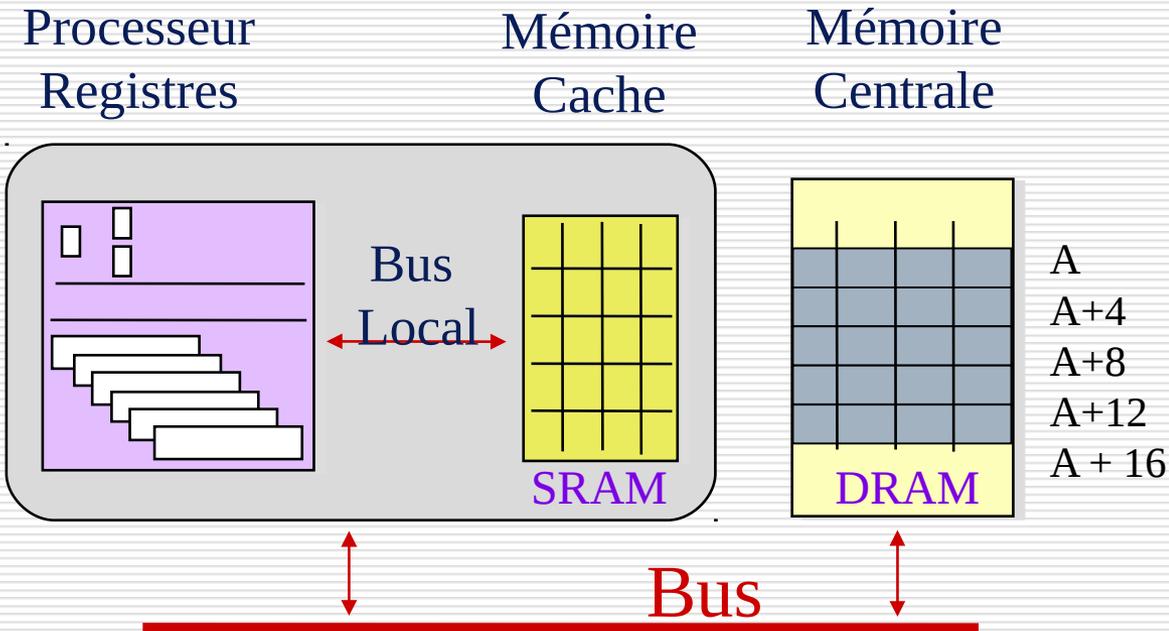
Localité spatiale

Premier accès à l'instruction I1

→ en MC

Les accès à I2, I3, I4, I5, I6 s'effectuent à partir du cache

Mémoire cache : le cache en lecture



Si (A) présent Alors Charger processeur avec (A)

Sinon

Charger cache avec (A) et ses voisines

Charger processeur avec (A)

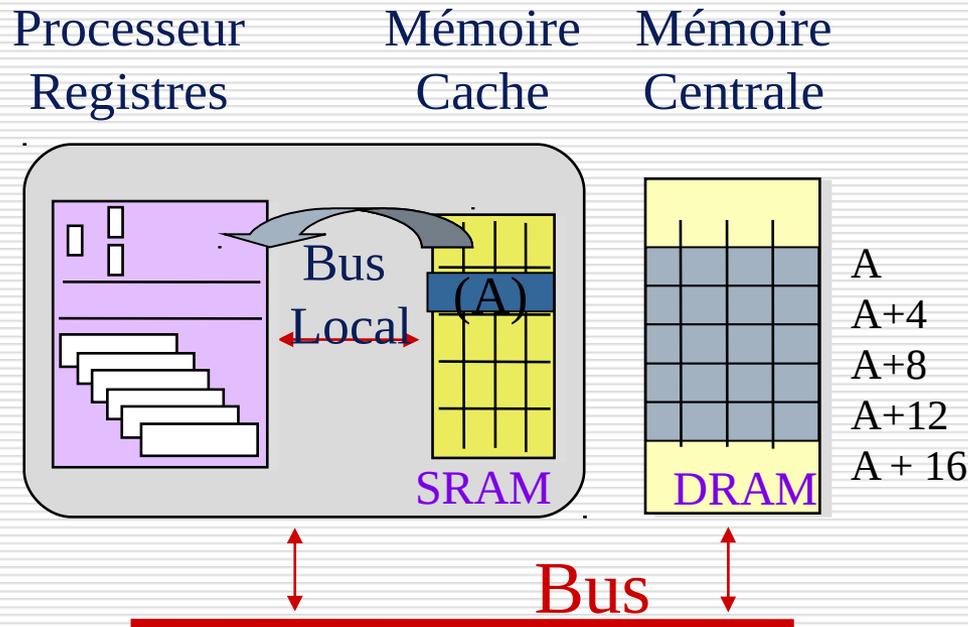
FinSi

FinSi

Lecture

Load D R1 A

Mémoire cache : le cache en lecture



Si (A) présent Alors Charger processeur avec (A)

Sinon

Charger cache avec (A) et ses voisines

Charger processeur avec (A)

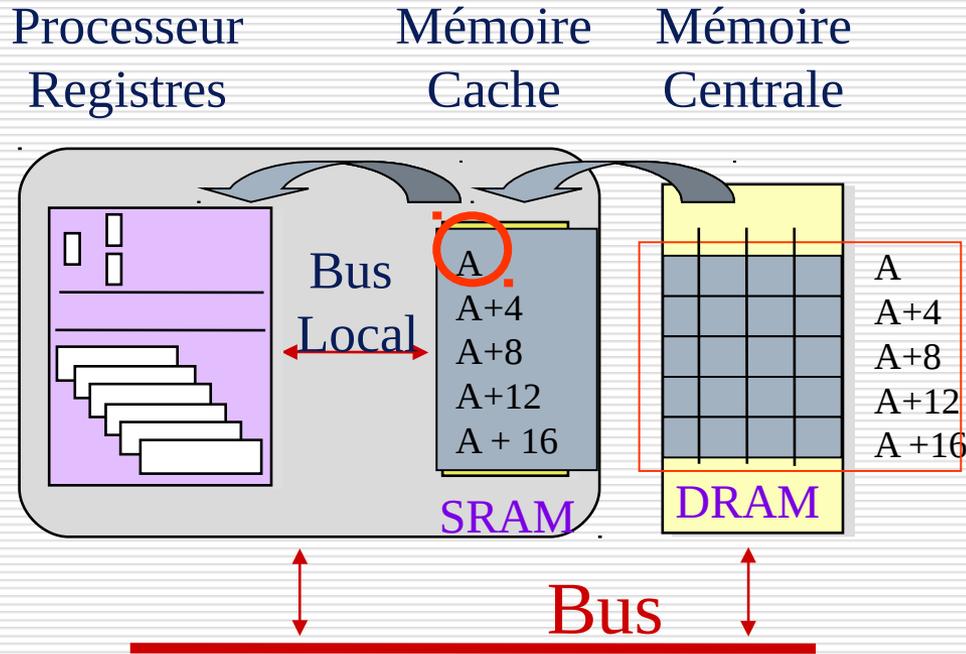
FinSi

FinSi

Lecture

Load D R1 A

Mémoire cache : le cache en lecture



Si (A) présent Alors Charger processeur avec (A)

Sinon

Charger cache avec (A) et ses voisines

Charger processeur avec (A)

FinSi

FinSi

Lecture
Load D R1 A

Principes d'exécution des instructions

La pile de la mémoire centrale



- Une zone particulière de la mémoire est gérée comme une structure de **pile**
- Elle est repérée par le registre **RSP** du processeur
- La politique de gestion est de type **LIFO (Last In First Out)**

Adresse
8 bits

Capacité mémoire : 2^8 octets
 2^6 mots

Fonctionnement d'une pile : politique LIFO

- ❑ La pile est une zone de la mémoire centrale constituée d'un ensemble de mots.
- ❑ Seul le mot au sommet de la pile est accessible. Son adresse est contenue dans le registre du processeur RSP
- ❑ Deux opérations sont seulement autorisées :
 - Ajouter un mot au sommet de la pile (PUSH)
 - Oter le mot au sommet de la pile (POP)

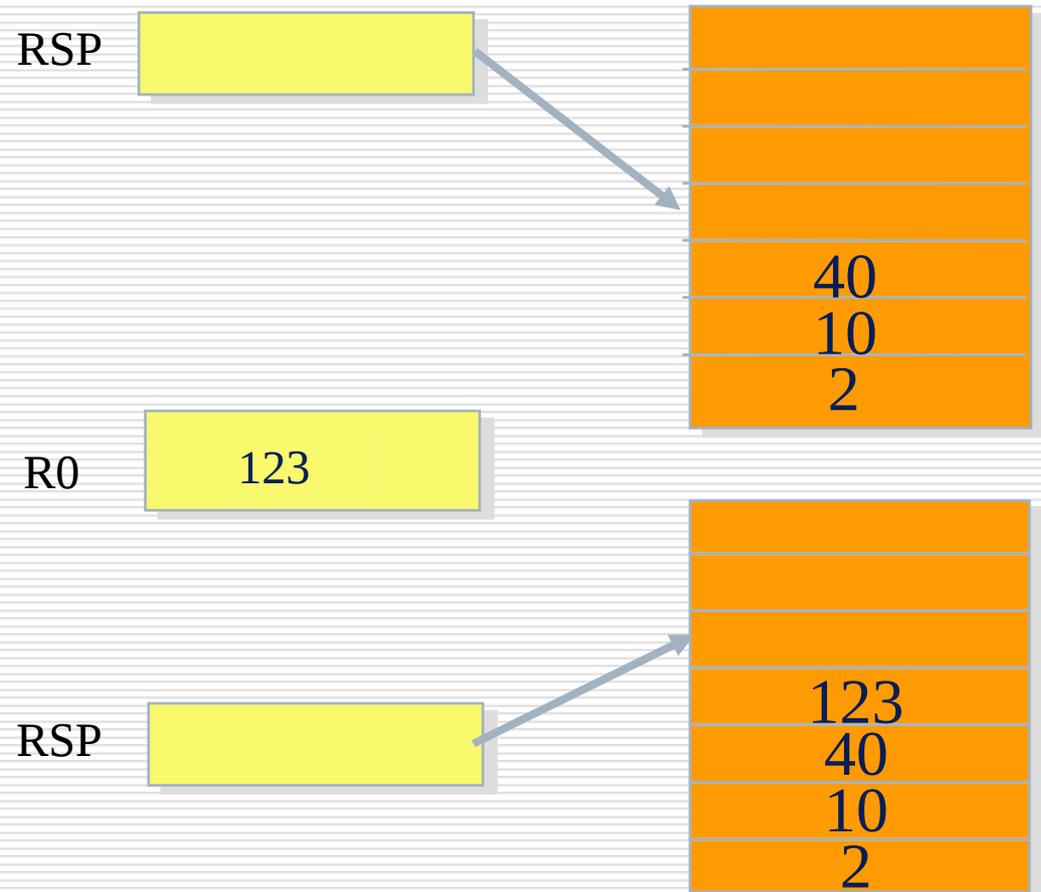


La pile : opération PUSH

- Deux opérations sont seulement autorisées :
 - Ajouter un mot au sommet de la pile (**PUSH**)

Ajouter un mot au sommet de la pile : **Push R0**

- (1) Le contenu du registre R0 est écrit dans le mot dont l'adresse est (RSP).
- (2) RSP est incrémenté

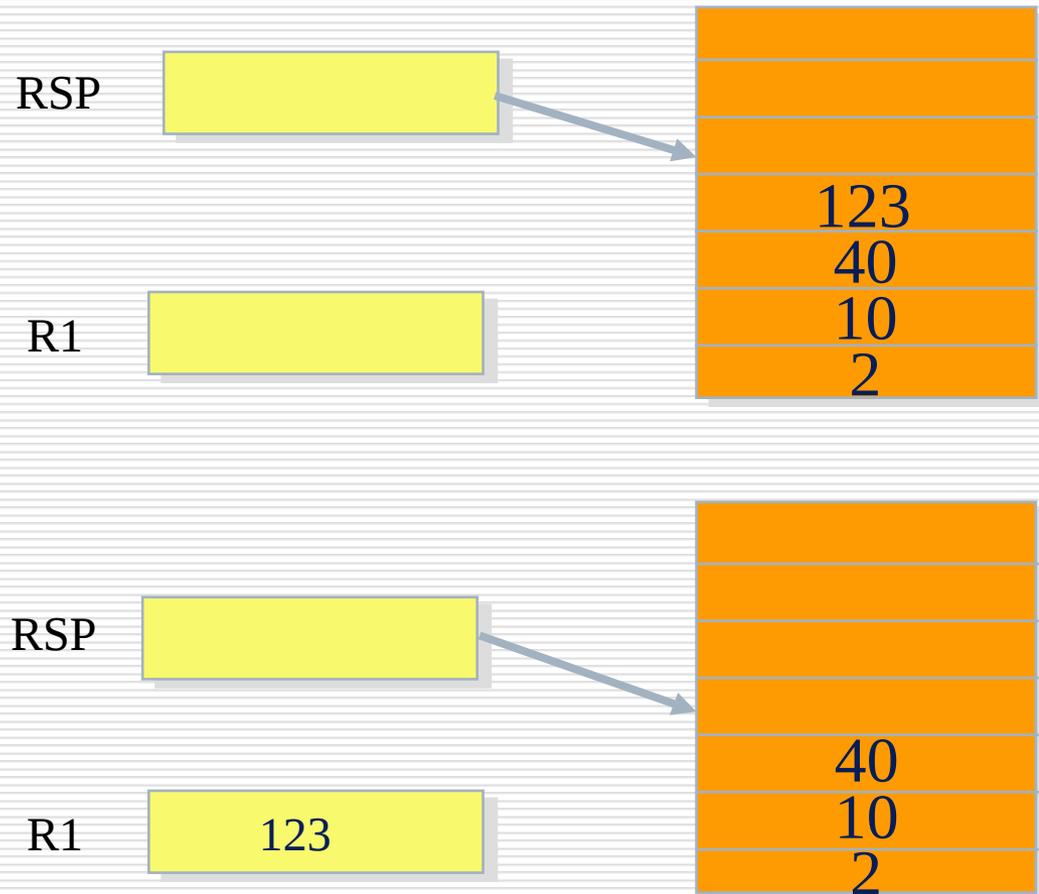


La pile : opération POP

- Deux opérations sont seulement autorisées :
 - Enlever le mot du sommet de la pile (**POP**)

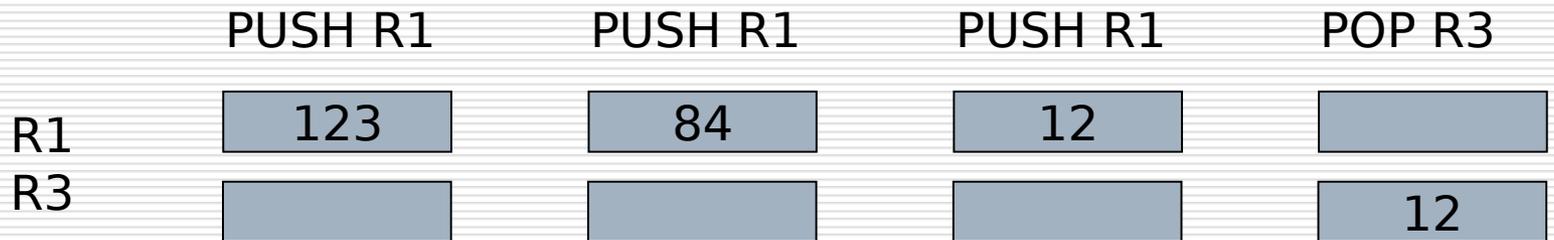
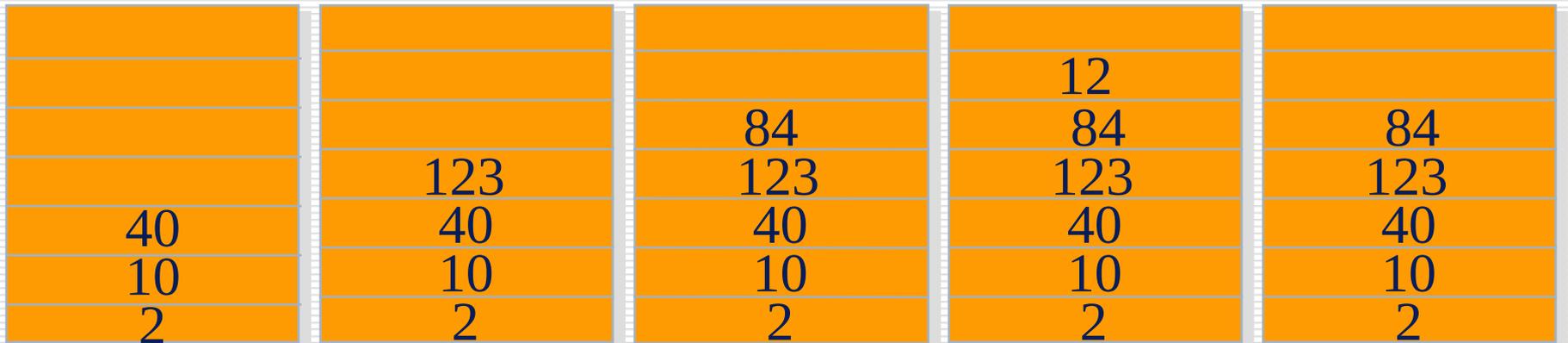
Enlever le mot du sommet de la pile : **POP R1**

(1) RSP est décrémenté.
(2) Le contenu du mot dont l'adresse est (RSP) est mis dans le registre R1.

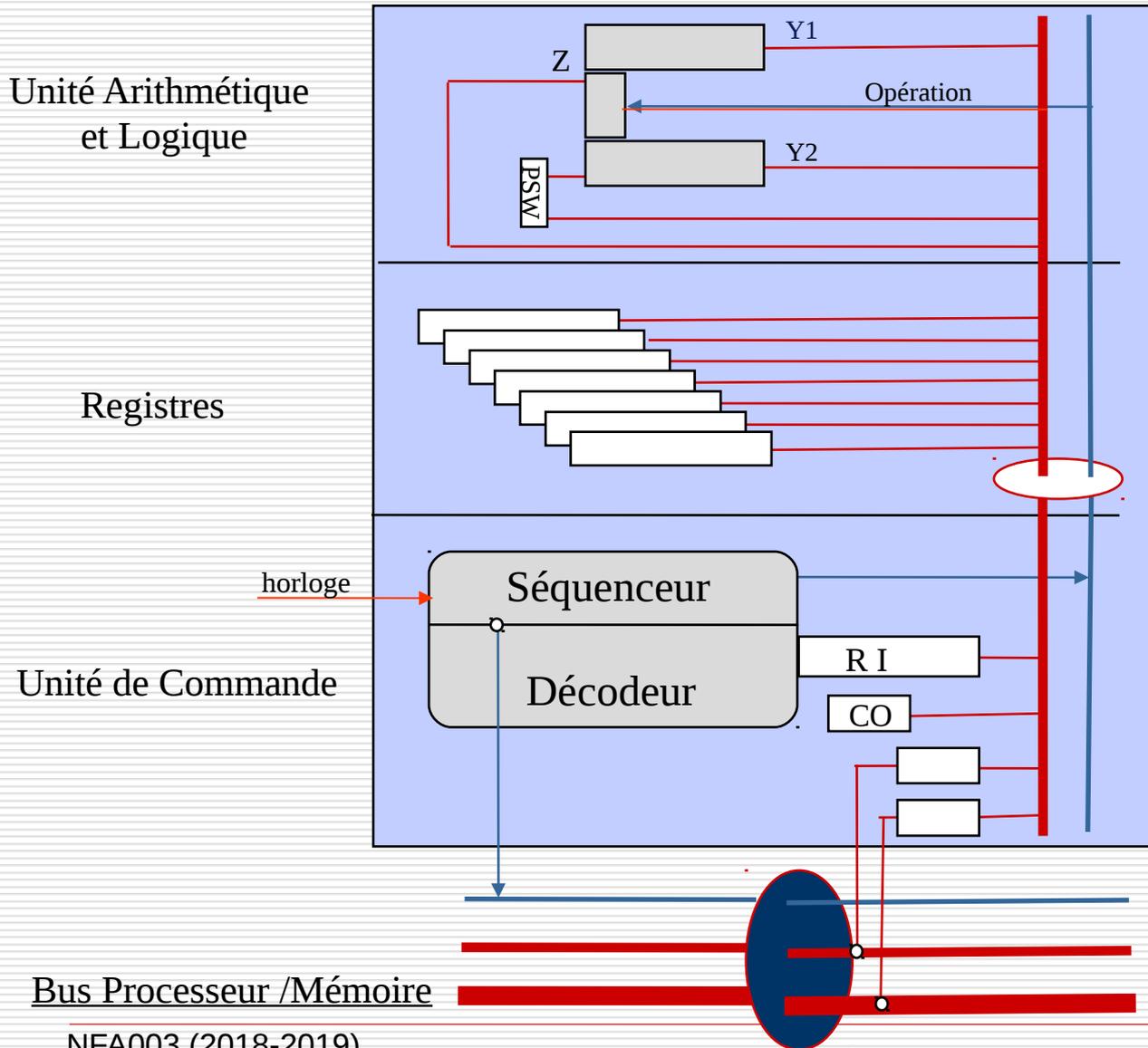


Fonctionnement d'une pile : Exemple

Politique LIFO (Last In, First Out) : le dernier élément ajouté est le premier retiré.



Processeur (Unité Centrale)



Le registre est l'entité de mémorisation manipulée par le processeur :

CO (compteur ordinal) : contient l'adresse de la prochaine instruction à exécuter

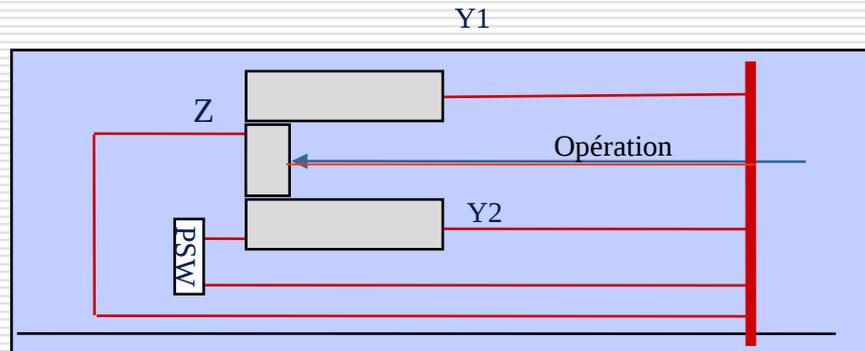
RI (registre instruction): contient l'instruction en cours d'exécution

Bus Processeur /Mémoire

Commandes Lecture/Ecriture
Adresses
Données

Processeur (Unité Centrale)

Unité Arithmétique et Logique



L'unité Arithmétique et Logique (UAL) 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
- d'un registre d'état PSW qui contient des indicateurs positionnés par le résultat Z des opérations effectuées :

S	O	C	Z	
---	---	---	---	--

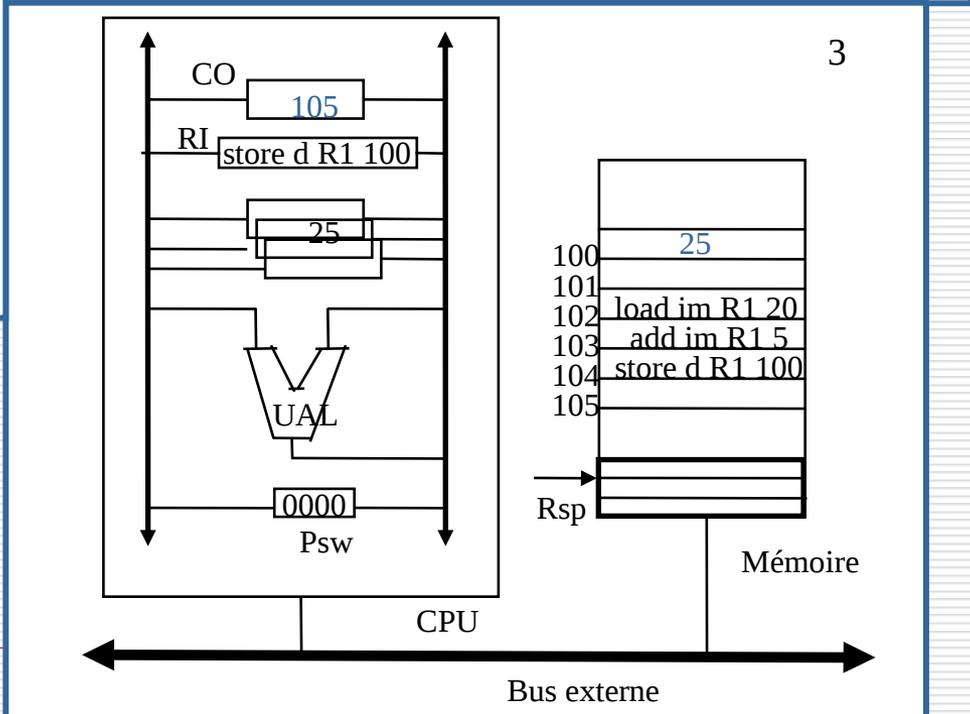
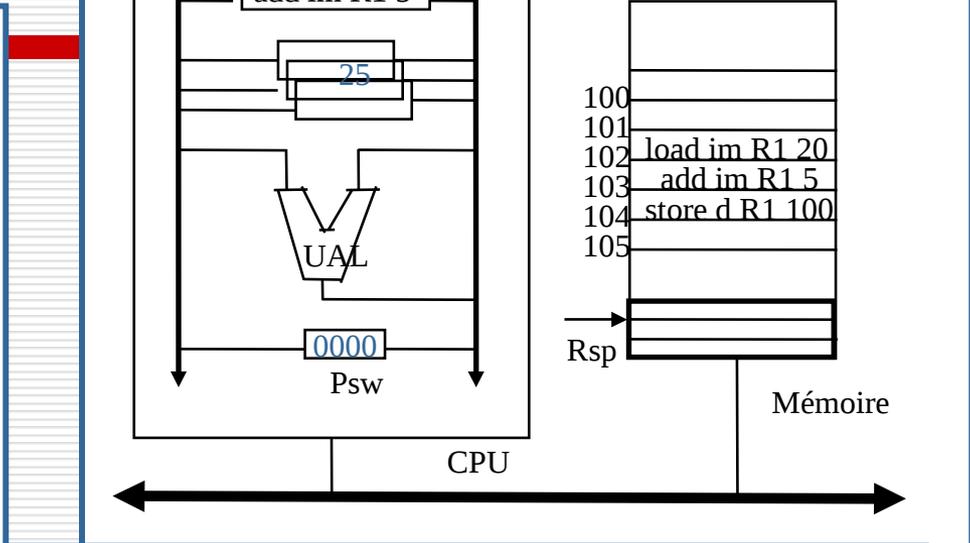
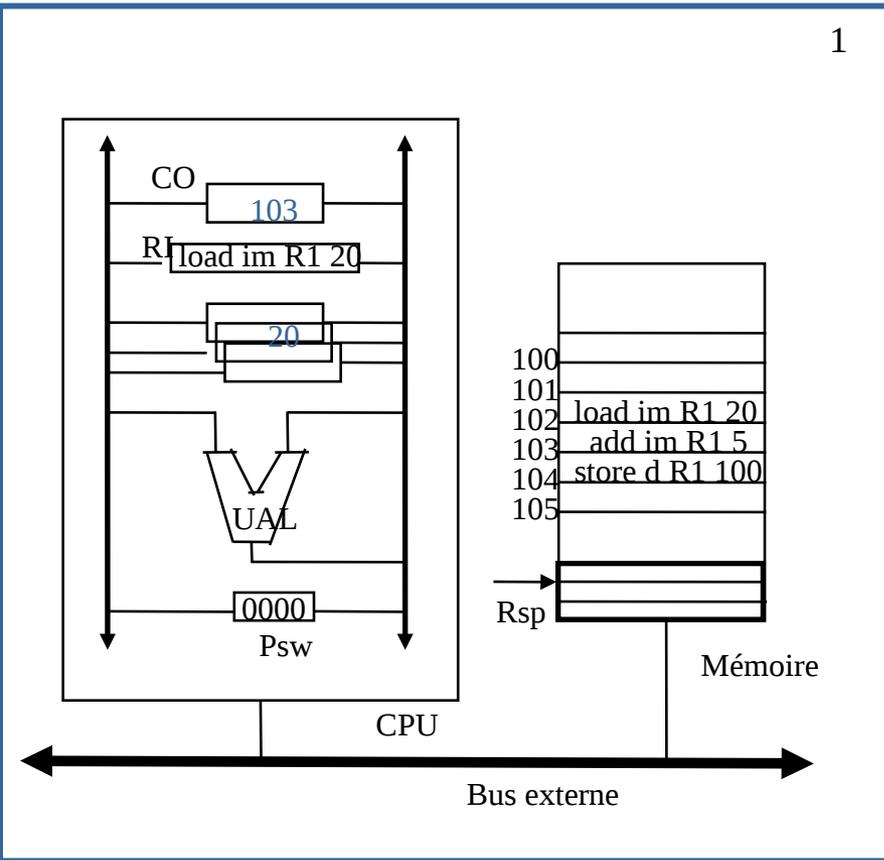
- O : positionné à 1 si Overflow, 0 sinon
- Z : positionné à 1 si résultat opération nul, 0 sinon
- C : positionné à 1 si carry, 0 sinon
- S : positionné à 0 si résultat opération positif, 1 sinon

Exécution d'un programme : Contexte

- ❑ 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 :



- ❑ **FETCH** : l'instruction est lue en mémoire centrale et copiée dans le registre RI du processeur
- ❑ **DECODAGE** : l'instruction est reconnue par l'unité de décodage
- ❑ **EXECUTION** : l'opération correspondant à l'instruction est réalisée



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

Les interruptions matérielles

L'exécution d'un programme s'effectue instruction après instruction.

Une **interruption** est un mécanisme permettant de stopper l'exécution du programme en cours afin d'aller exécuter une tâche jugée plus prioritaire.

Une **interruption** permet de signaler un événement survenu sur la machine et d'exécuter un traitement spécifique (programme de service ou **routine d'interruption**) lié à cet événement.

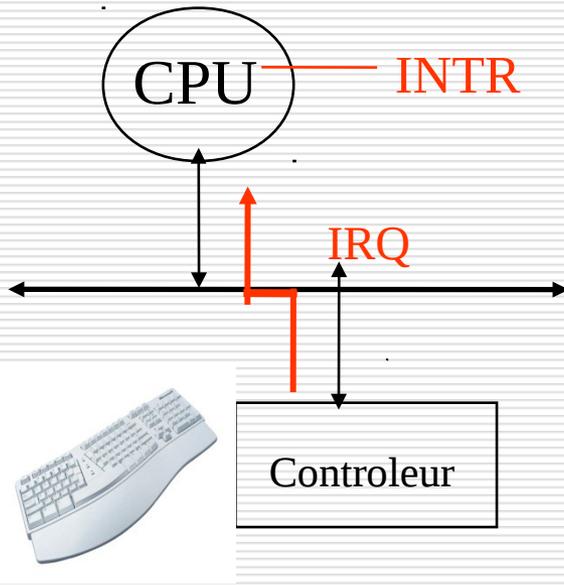
Elle évite au processeur de **scruter** les périphériques

Notion d'interruptions

On distingue principalement deux types d'événements :

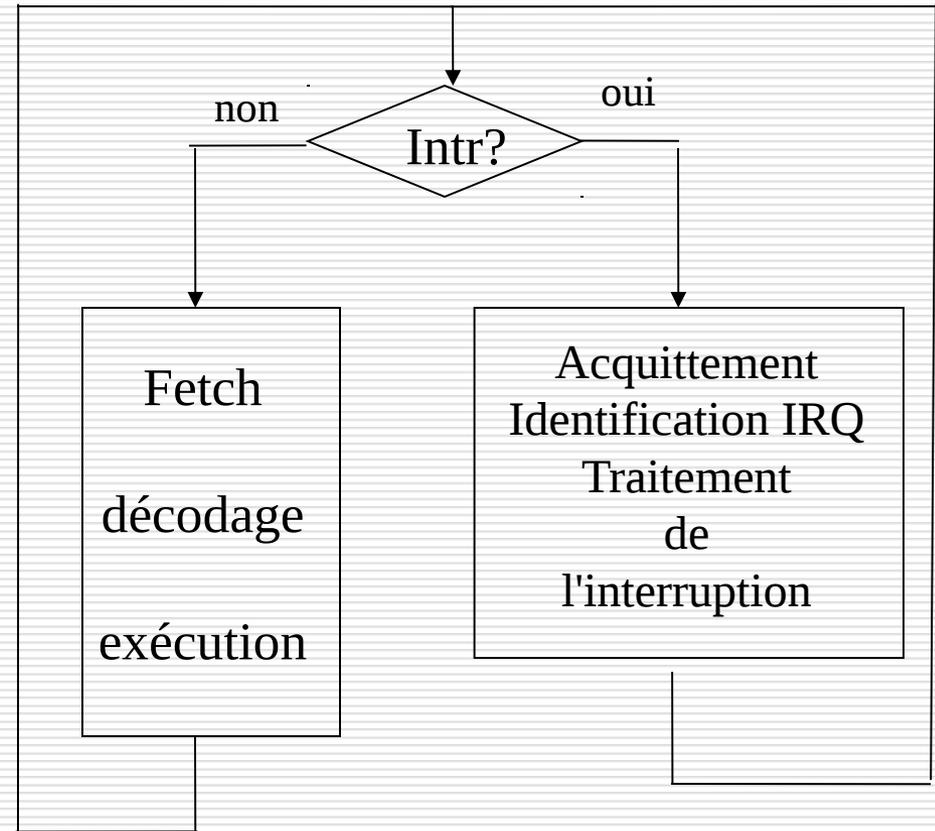
- **Les interruptions externes ou matérielles** sont émises par les périphériques du processeur (fin d'écriture disques, plus de papier imprimante...). Ce sont les **IRQs**.
- **Les interruptions internes ou logicielles** sont émises par le processeur lui-même lorsqu'il rencontre une erreur dans l'exécution du programme (division par zéro, accès mémoire illégal). Ce sont les **trappes**.

Mécanisme des interruptions

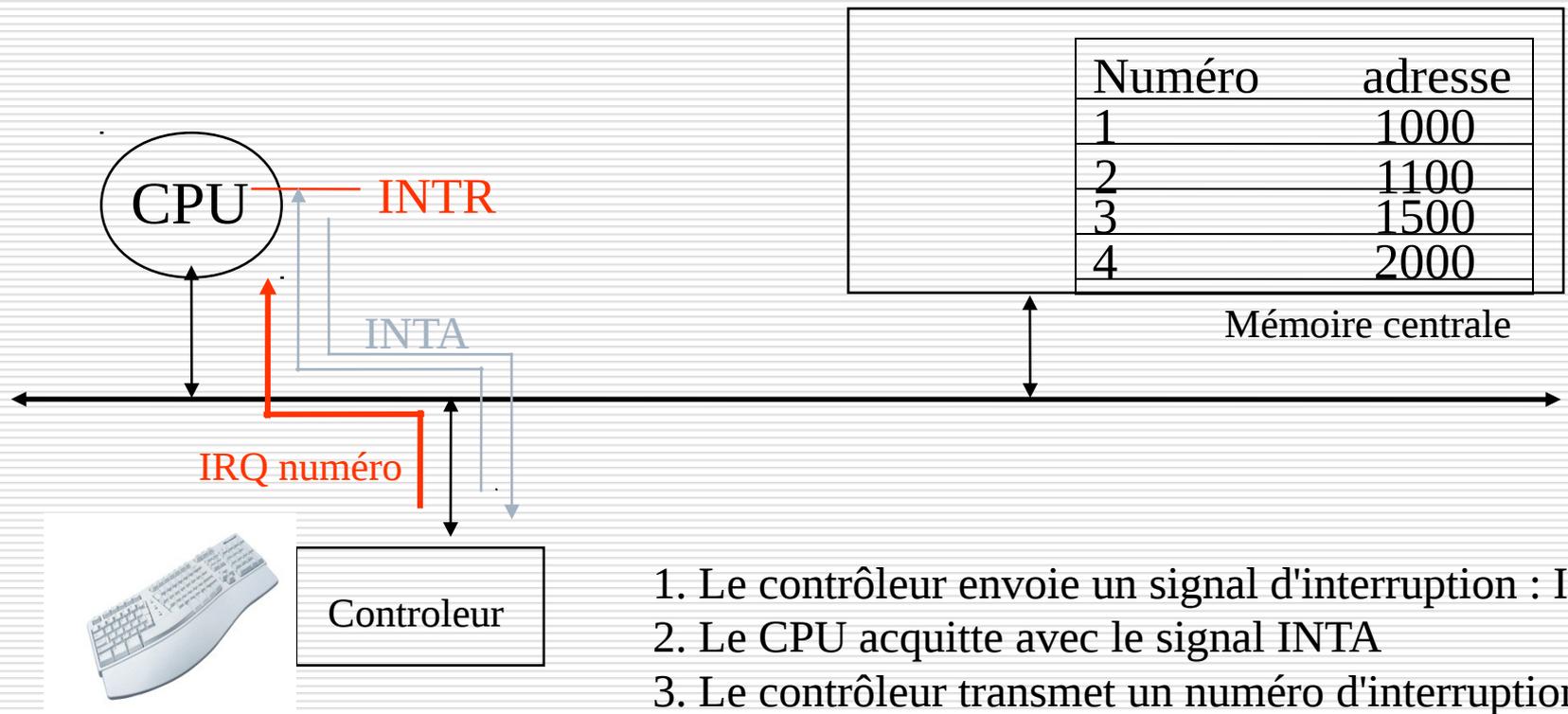


Un périphérique signale un événement au processeur en émettant une interruption matérielle.

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

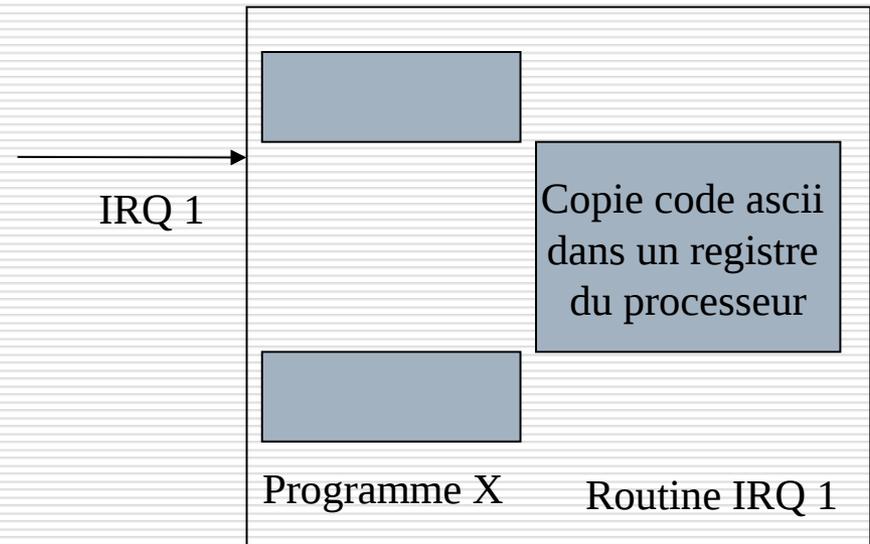
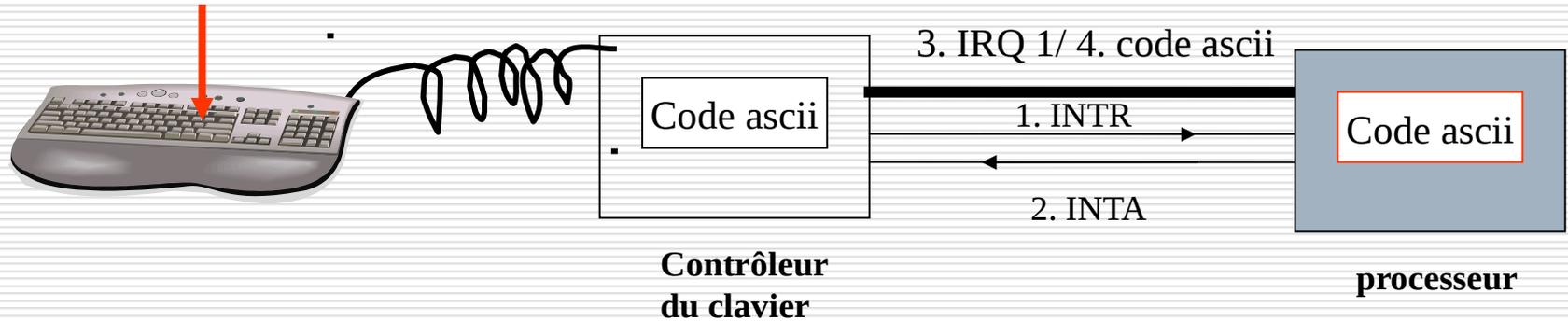


Mécanisme des interruptions



Ce numéro indexe une table en mémoire centrale le **vecteur d'interruption**, qui pour chaque numéro d'interruption donne l'adresse de la routine associée en mémoire centrale.

Exemple : IRQ 1 (donnée du clavier)



Exécutions du CPU