

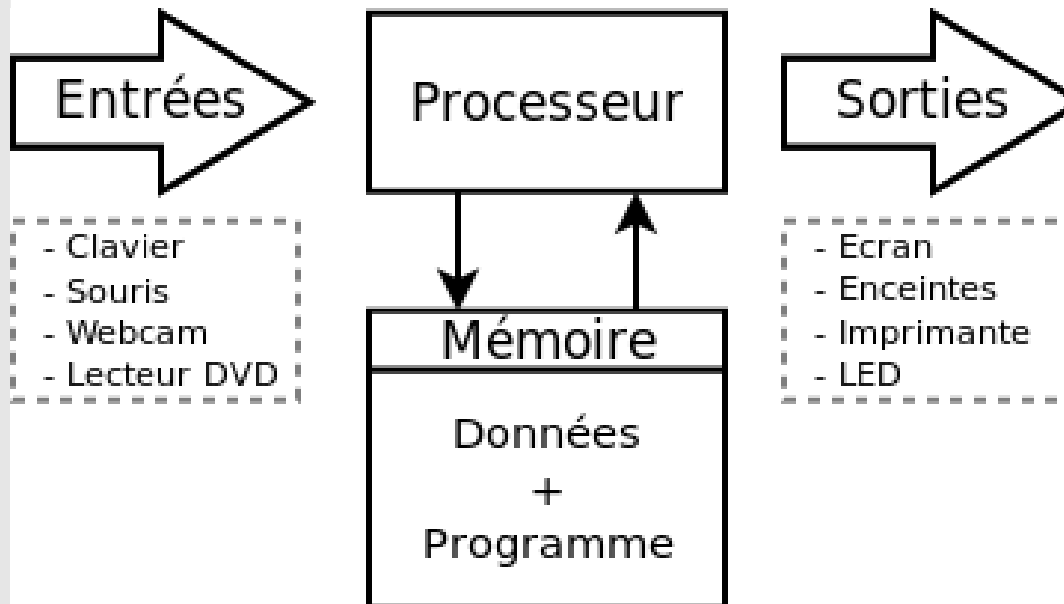
---

# Introduction à l'architecture des ordinateurs



# Le modèle de Von Neumann

L'ordinateur est une machine à calculer à mémoire, doté d'un programmeur (C. Babbage)



**Programme** : suite d'actions élémentaires choisies parmi plusieurs actions possibles.

**Le programmeur est chargé d'orchestrer la suite des opérations (instructions).**

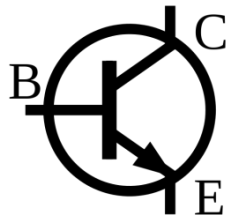


---

# Composants matériel d'un ordinateur :

## Processeur et Mémoire

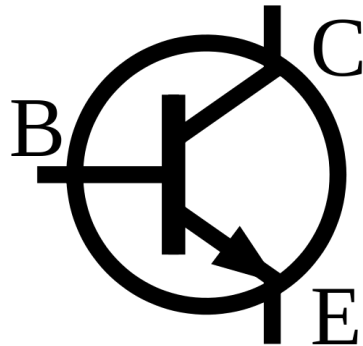
Comment construire un processeur et une mémoire à partir de composants électroniques



# Du transistor au processeur

---

- **Le transistor** : composant électronique semi-conducteur
  - Peut laisser passer le courant ou pas
  - Similaire à un interrupteur
  - Deux états physique possible (bloquant ou passant) représenté par deux **états logique** (0 ou 1)


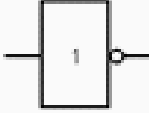
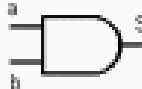
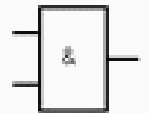
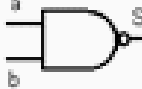
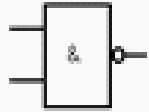

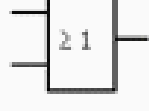



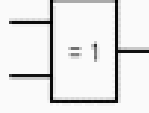

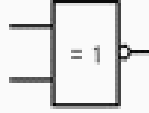


Le courant passe entre C et E  
si et seulement si B est alimenté



# Du transistor au processeur

- **Les portes logiques :**
  - Brique de base de tout système **numérique**
  - Leurs entrées et sorties sont appelées des **bits**
  - Un **bit** peut prendre deux valeurs : **0 ou 1**
  - Implémentent des fonctions logiques simples : ET, OU, NON ...
  - On peut les réaliser à partir de transistors

| FONCTION     | SYMBOLES  |   | TABLES DE VERITE |   |   |
|--------------|---|---|------------------|---|---|
|              | International   | Français  | a                | b | S |
| NON          |    |    | a                | S |   |
|              |   |   | 0                | 1 |   |
| ET           |    |    | a                | b | S |
|              |   |   | 0                | 0 | 0 |
| NAND         |    |    | a                | b | S |
|              |   |   | 0                | 0 | 1 |
| OU           |    |    | a                | b | S |
|              |   |   | 0                | 0 | 0 |
| NOR          |   |   | a                | b | S |
|              |   |   | 0                | 0 | 1 |
| OU Exclusif  |  |  | a                | b | S |
|              |   |   | 0                | 0 | 0 |
| NOR Exclusif |  |  | a                | b | S |
|              |   |   | 0                | 0 | 1 |

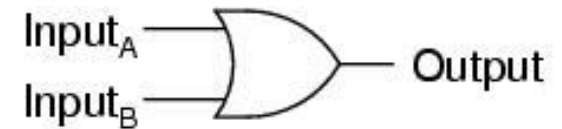


# Du transistor au processeur

- Exemple de la porte OU (OR):
  - Deux **entrées** A et B
  - Une **sortie** (Output)
  - La valeur de la sortie dépend de celle des entrées

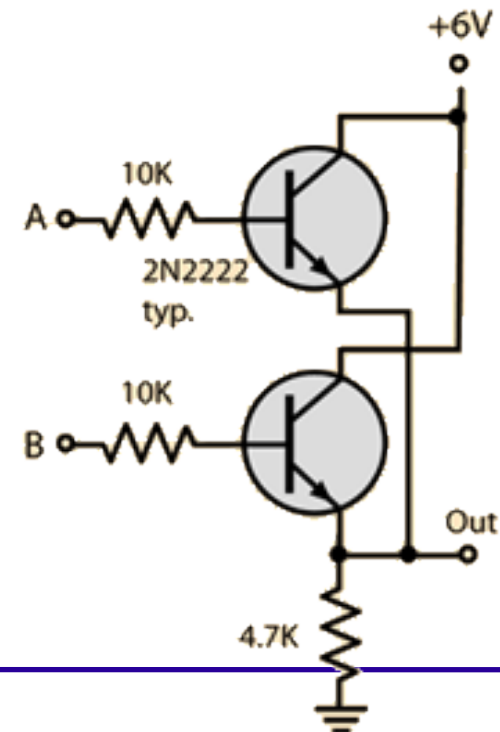
A et B : les deux bits d'entrée  
Output : le bit de sortie

2-input OR gate



| A | B | Output |
|---|---|--------|
| 0 | 0 | 0      |
| 0 | 1 | 1      |
| 1 | 0 | 1      |
| 1 | 1 | 1      |

- La porte de OR peut être réalisée à partir de deux transistors :



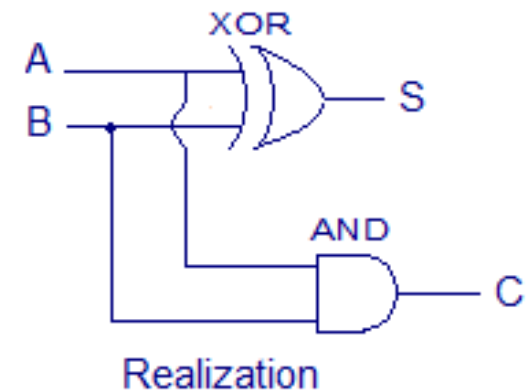
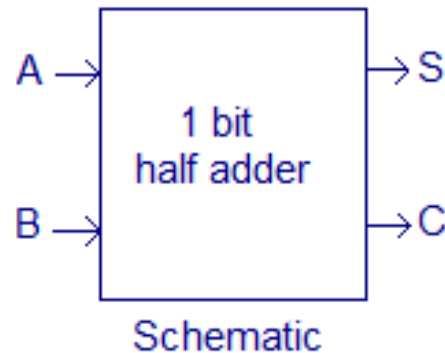
# Du transistor au processeur

- **Opérateurs arithmétiques**

- Implémentent des fonctions arithmétiques de bases : addition, soustraction, multiplication, division, comparaison ( $<$ ,  $>$ ,  $=$ ) ....
- On peut les réaliser avec des portes logiques
- Exemple de l'**additionneur 1 bits** :

| Inputs |   | Outputs |   |
|--------|---|---------|---|
| A      | B | S       | C |
| 0      | 0 | 0       | 0 |
| 1      | 0 | 1       | 0 |
| 0      | 1 | 1       | 0 |
| 1      | 1 | 0       | 1 |

Truth table



Ex:  $1 + 0 = 1$   
(pas de retenue)

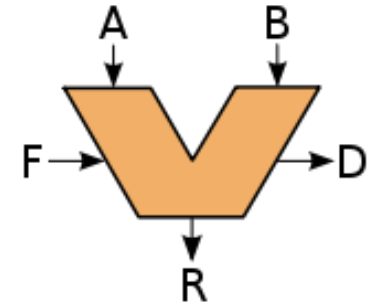
A et B : les deux bits d'entrée  
S : le résultats  
C : la retenue éventuelle



# Du transistor au processeur

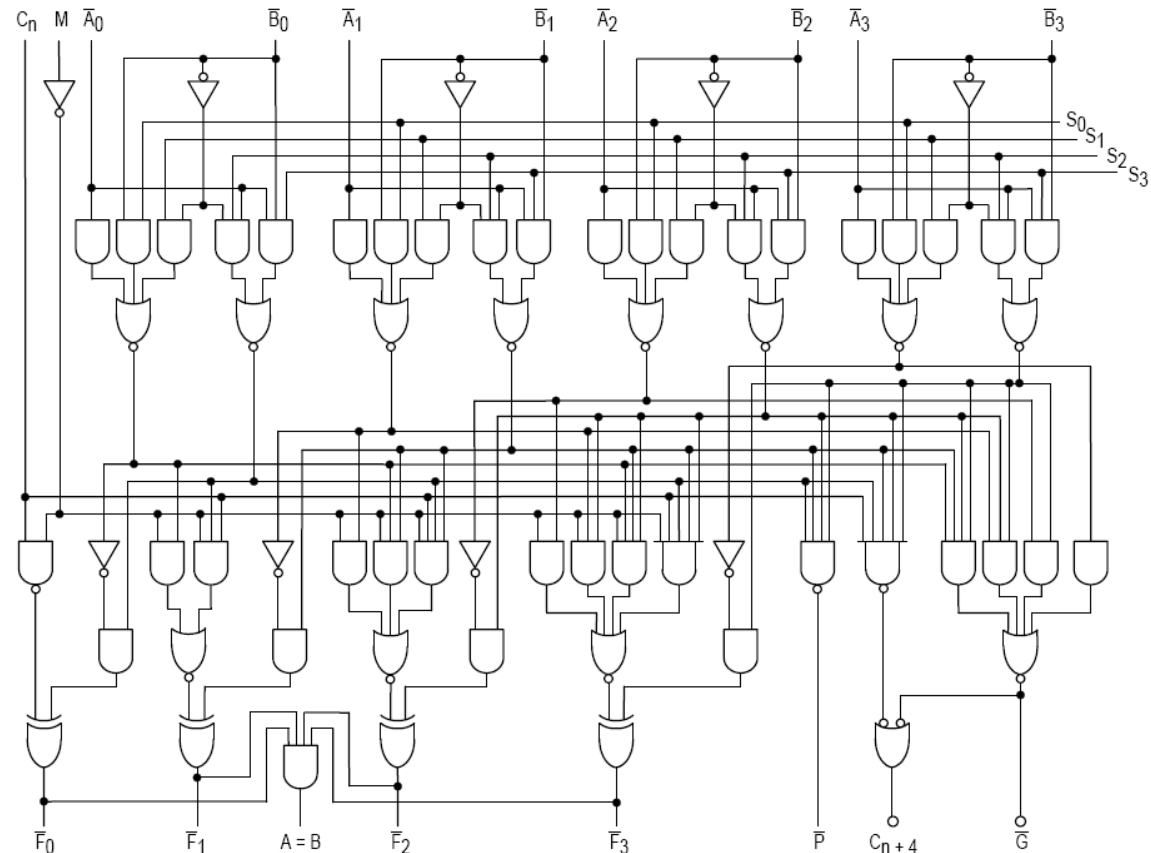
- **UAL : Unité Arithmétique et Logique**

- Effectuent des **opérations** arithmétiques et logiques
- Circuit complexe composé d'un grand nombre de **portes logiques de base**



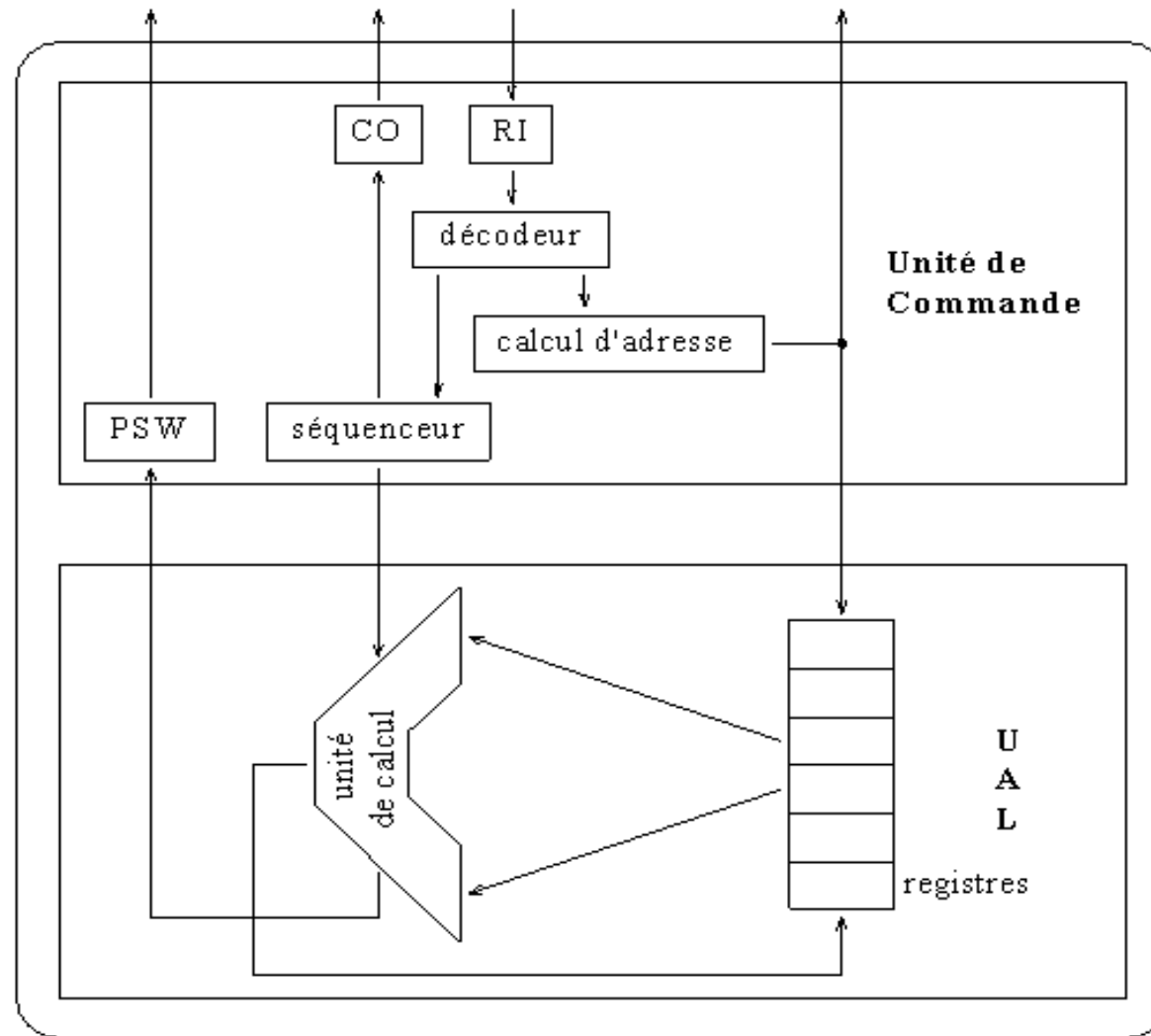
Exemple de l'UAL 4 bits 74181

- Tailles de opérantes : 4 bits
- Nombre de fonctions : 32

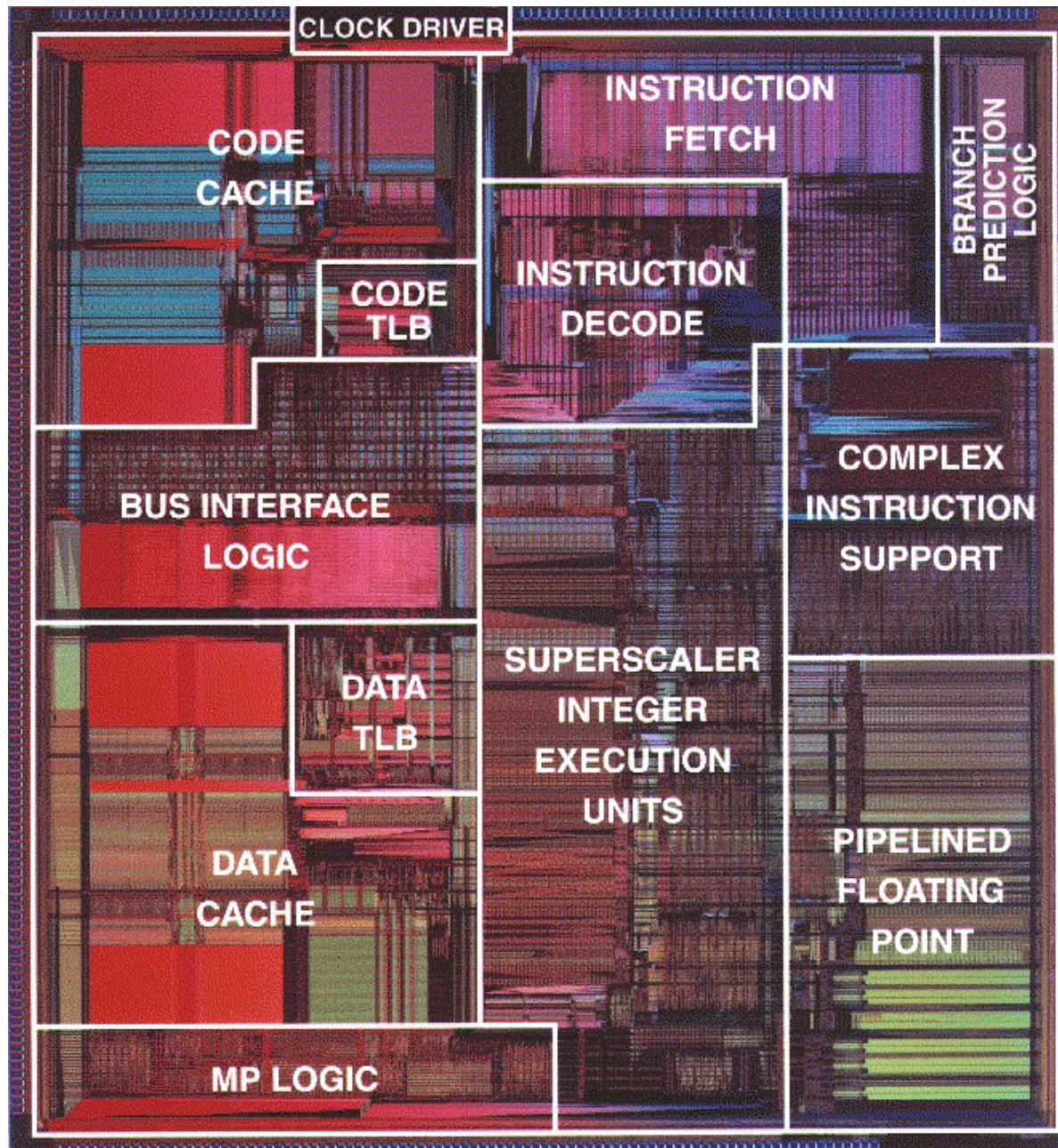




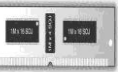
## L'unité centrale du processeur



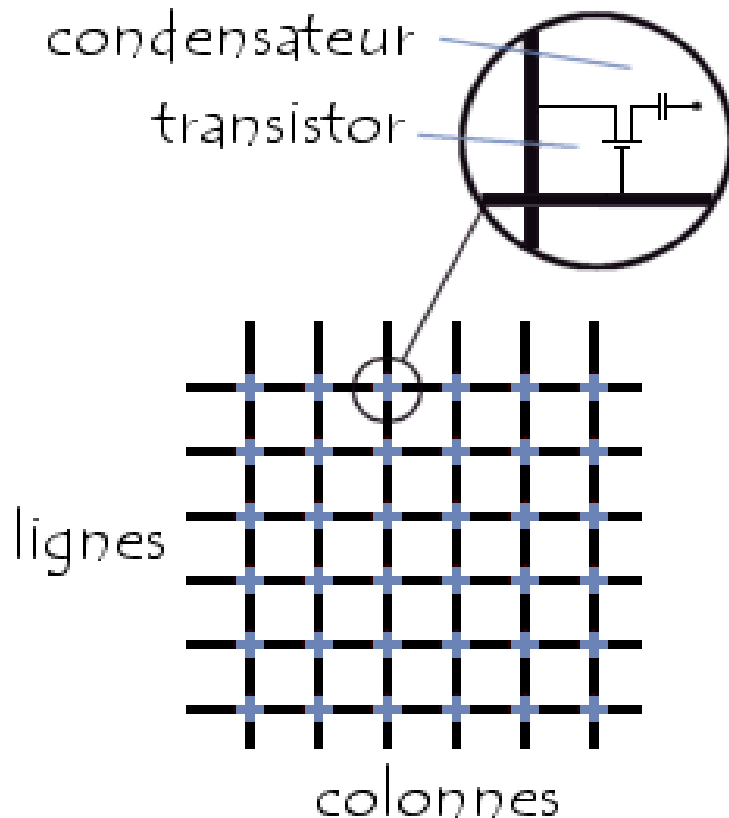
# Détails d'un processeur (Intel Pentium - 1993)



- 3 100 000 transistors
- Opérandes de 32 bits
- 200 000 000 opérations par seconde



# La mémoire centrale



L'information binaire est stockée sous la forme d'une **charge** dans un **condensateur**. Cette charge est lue/écrite par un transistor à effet de champ (MOS).

Les cases sont rangées en **matrices**. Chaque point mémoire est repéré par son indice de **ligne** et son indice de **colonne**.

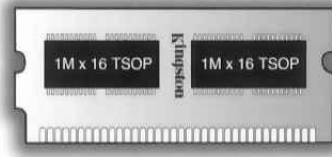
Un emplacement mémoire est repéré par son **adresse**.

Nécessité de **raffraîchir** la mémoire à intervalles réguliers (cycle de quelques dizaines de nanosecondes) + temps d'accès de quelques dizaines de nanosecondes.





# Les mémoires vives



## Stocke les informations

Données  
Programmes  
Calculs intermédiaires

## RAM : Random Access Memory Mémoire VIVE

Volatile  
Accessible en lecture/écriture

### Registres

(8 à 32 registres de 8 à 64 bits)

### Mémoire cache

(~1 Mo, Sert de tampon entre la mémoire vive et le processeur)

### Mémoire centrale , RAM

(~512 Mo, Stockage des données et instructions)

Vitesse

1

/ 10

/100

Capacité de  
stockage



# Mémoire non volatile

---

## ● Disques durs (HDD)

**Disques de verre avec un support magnétique.**

**Lecture/écriture très lentes comparées à la mémoire vive**

**Accès par blocs de 512 octets**

**Très grande capacité de stockage**



## ● Mémoires flash

**Mémoire de masse ré-inscriptible basée sur des semi-conducteurs**

**Ex: Clés USB, Solid-State-Drive (SSD), carte SD / microSD, etc ...**

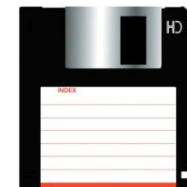
**Très grande capacité de stockage, plus rapide que les HDD, mais aussi plus cher**



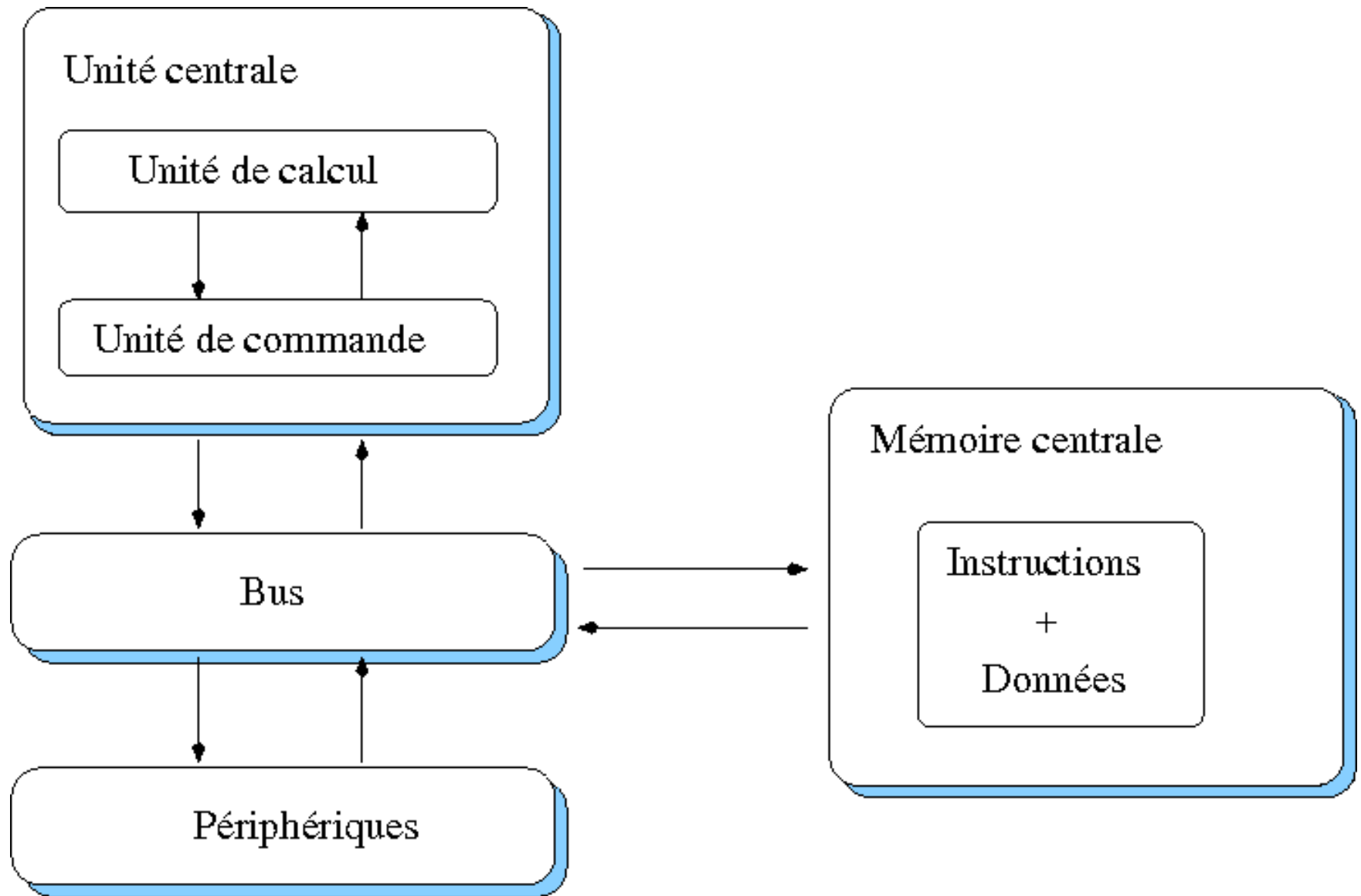
## ● Autres types de mémoires non volatiles

**Supports optiques : CDRom, DVD**

**Supports magnétiques : disquettes, et bandes**



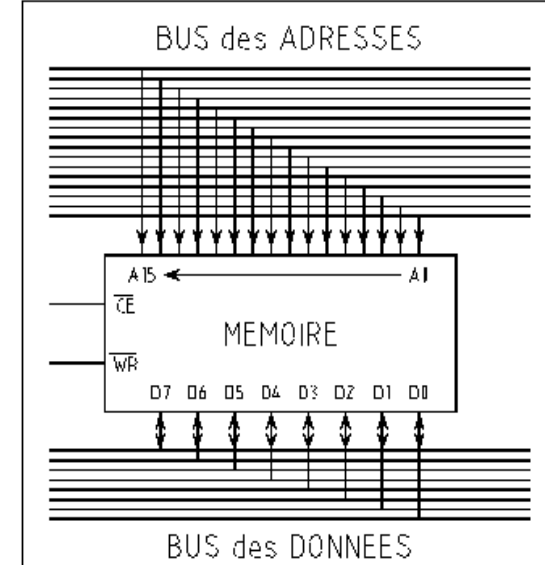
# Principe d'organisation du matériel



# Le bus

## Bus d'adresses ou bus mémoire

transporte les adresses mémoire auxquelles le processeur souhaite accéder pour lire ou écrire une donnée. Il s'agit d'un bus unidirectionnel : du processeur vers la mémoire.



## Bus de données

Véhicule les données et instructions entre le processeur et la (les) mémoire(s). Bidirectionnel

Le transfert d'une donnée (instruction) vers une adresse implique donc les deux bus

## Bus de contrôle ou de commandes

Ordres et signaux de synchronisation en provenance de l'unité de commande et à destination de l'ensemble des composants matériels. Bidirectionnel : il transmet également les signaux de réponse des éléments matériels.



# Les périphériques

---







---

# Fonctionnement de l'unité centrale de traitement et de la mémoire centrale

# Principe d'organisation

---

L'ordinateur est une machine qui traite l'information dans une unité centrale, selon un programme qui est enregistré en mémoire.

Les données fournies en entrée sont traitées par l'unité centrale en fonction du programme et les résultats sont délivrés en sortie.

- L'information est numérisée (valeurs discrètes)
- La mémoire contient **les instructions et les données**
- La mémoire est un ensemble de mots de longueur fixe contenant une information codée en **binaire**.
- On accède aux mots par l'intermédiaire de l'**adresse en mémoire**.



# Les différents niveaux

---

Logiciel

Application  
Stockée en mémoire  
Ecrit par une personne (vous)

Programmable

Matériel

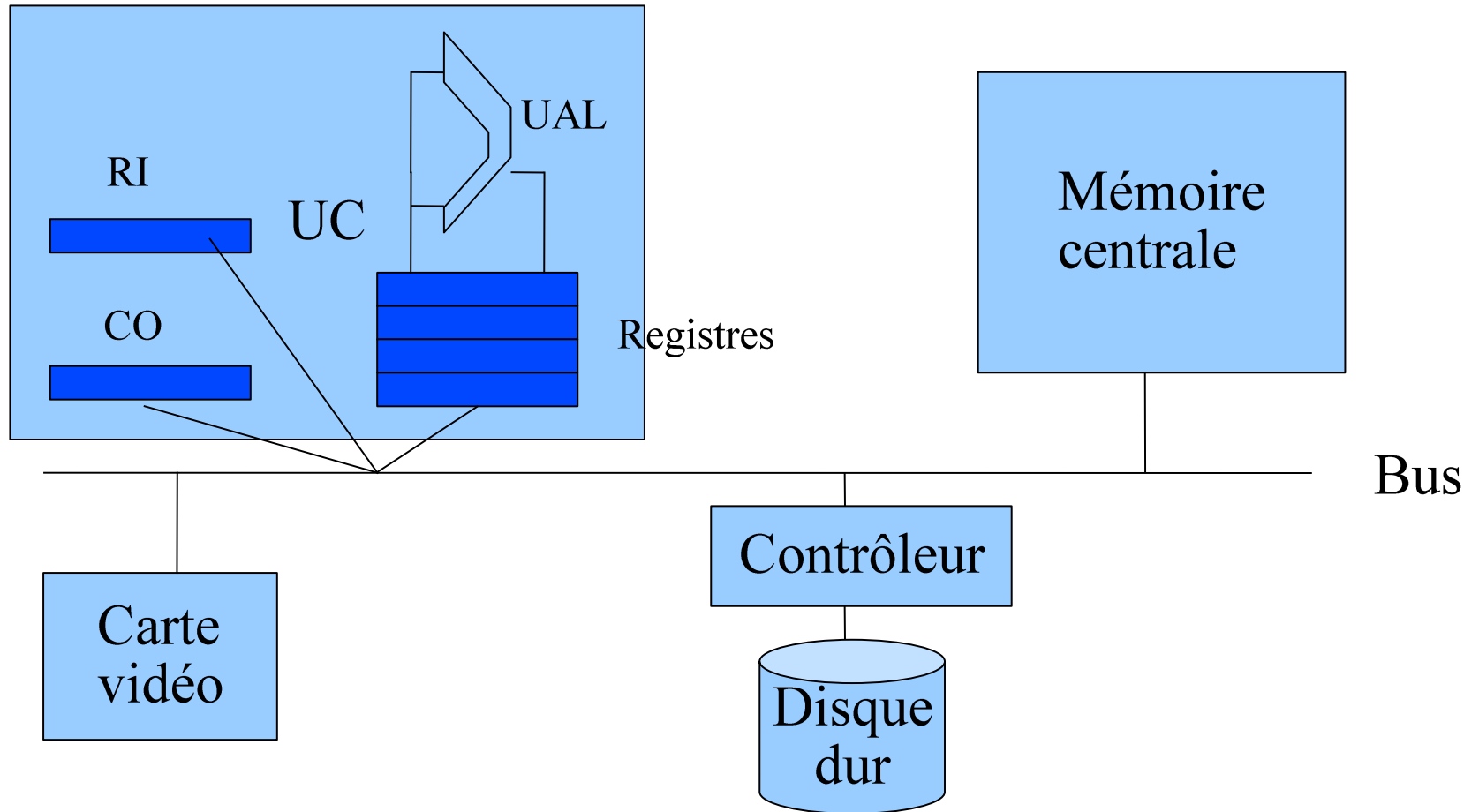
Microarchitecture matérielle  
Circuits intégrés fixes

Non modifiable

- Application = Langage machine : interface de programmation du matériel
- Le langage machine est implanté par divers circuits logiques :
  - Registres (mémoire) + unités de calcul
  - Les instructions sont exécutées en séquence : l'UC conserve l'adresse de la prochaine instruction dans un registre
  - A chaque instruction, l'UC
    - lit l'instruction à l'adresse courante
    - exécute l'instruction
    - incrémente l'adresse de l'instruction courante



# Architecture d'ordinateur



Le **programme** est en mémoire.

L'unité de contrôle va chercher les instructions en mémoire, les décompose et les passe à l'UAL, qui rend un résultat transféré en mémoire.

Les transferts se font par l'intermédiaire de **registres**.

# Un exemple de fonctionnement du processeur

---

Le processeur ne réalise que des opérations simples :

- addition, soustraction, multiplication, division
- opérations booléennes (et, ou, non)
- comparaison de valeurs saut dans la séquence d'instructions

Exemple : Évaluer une expression  
de la forme  
 $ax^2+bx+c$

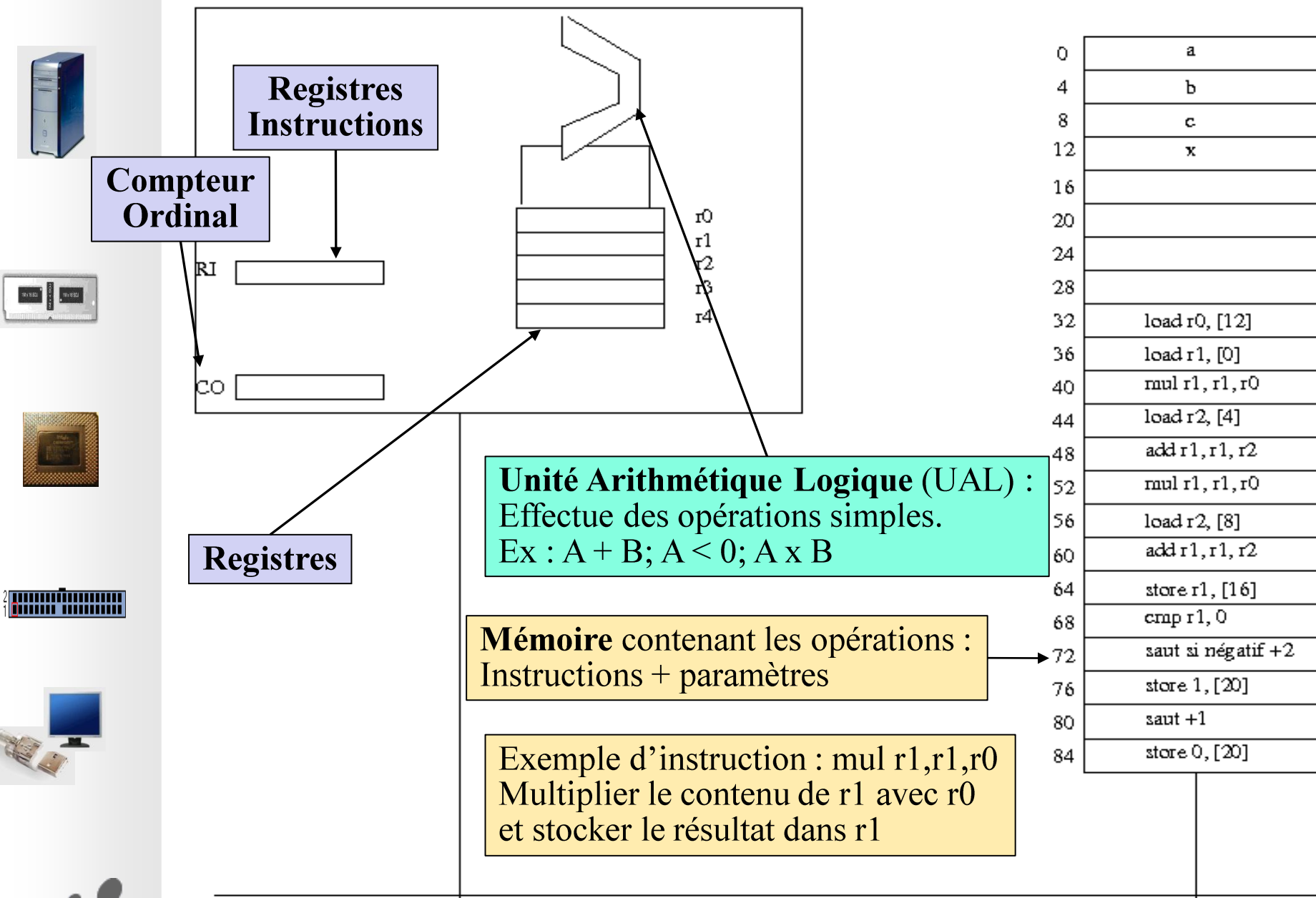
**entrées**: a, b, c, x

**sorties** : Stocker le résultat à l'adresse 16

Si le résultat est positif alors mettre 1 à  
l'adresse 20, sinon mettre 0

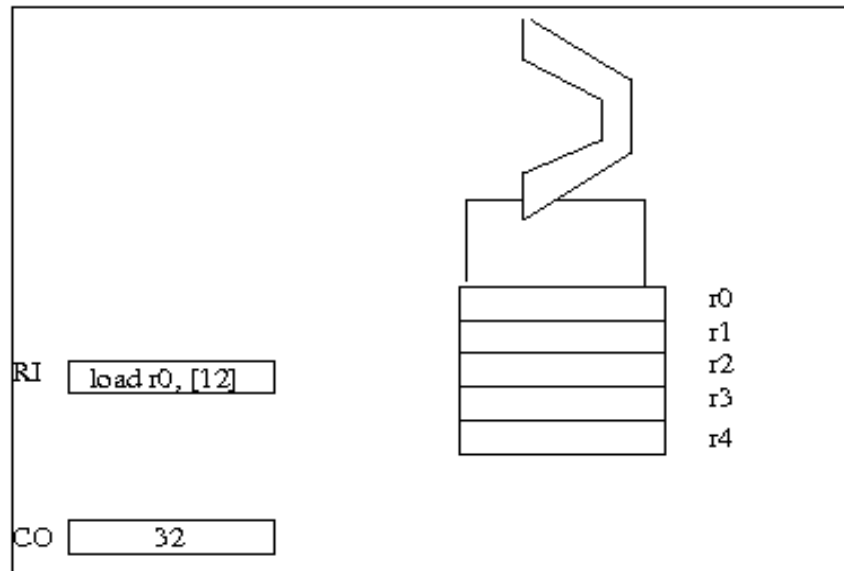


# Un exemple de fonctionnement du processeur



Bus

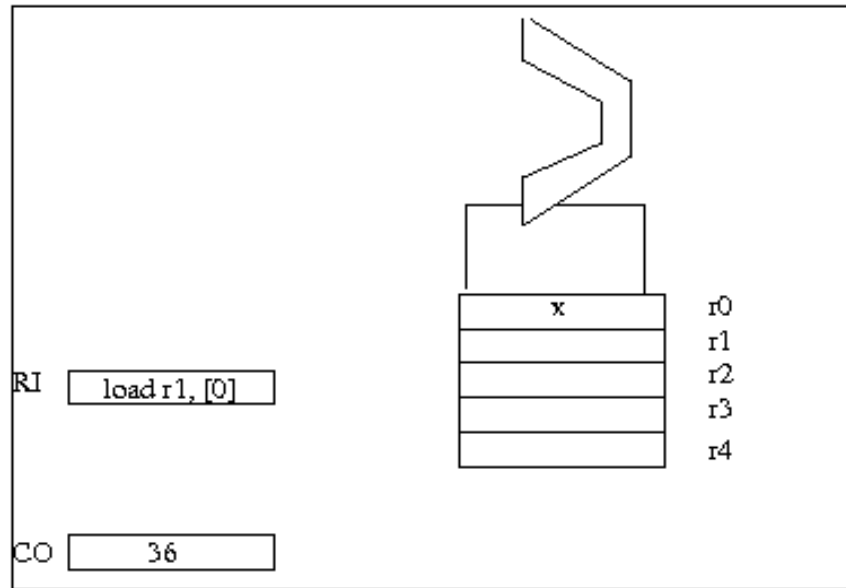
# Un exemple de fonctionnement du processeur



|    |                    |
|----|--------------------|
| 0  | a                  |
| 4  | b                  |
| 8  | c                  |
| 12 | x                  |
| 16 |                    |
| 20 |                    |
| 24 |                    |
| 28 |                    |
| 32 | load r0, [12]      |
| 36 | load r1, [0]       |
| 40 | mul r1, r1, r0     |
| 44 | load r2, [4]       |
| 48 | add r1, r1, r2     |
| 52 | mul r1, r1, r0     |
| 56 | load r2, [8]       |
| 60 | add r1, r1, r2     |
| 64 | store r1, [16]     |
| 68 | cmp r1, 0          |
| 72 | saut si négatif +2 |
| 76 | store 1, [20]      |
| 80 | saut +1            |
| 84 | store 0, [20]      |

Bus

# Un exemple de fonctionnement du processeur



|    |                    |
|----|--------------------|
| 0  | a                  |
| 4  | b                  |
| 8  | c                  |
| 12 | x                  |
| 16 |                    |
| 20 |                    |
| 24 |                    |
| 28 |                    |
| 32 | load r0, [12]      |
| 36 | load r1, [0]       |
| 40 | mul r1, r1, r0     |
| 44 | load r2, [4]       |
| 48 | add r1, r1, r2     |
| 52 | mul r1, r1, r0     |
| 56 | load r2, [8]       |
| 60 | add r1, r1, r2     |
| 64 | store r1, [16]     |
| 68 | cmp r1, 0          |
| 72 | saut si négatif +2 |
| 76 | store 1, [20]      |
| 80 | saut +1            |
| 84 | store 0, [20]      |

Bus



# En résumé

---

- **Un ordinateur est composé d'un grand nombre d'éléments électronique simple**
- **Un ordinateur ne fait que des choses simples mais il les fait très rapidement : calculs, chargement déchargement de données.**
- **Les calculs sont effectués par le processeur à partir des instructions d'un programme**
- **La mémoire vive n'est pas un stockage permanent : les données qui y sont stockées disparaissent si on coupe l'alimentation**
- **Les données transitent entre le processeur et la mémoire en passant par le bus**

