

# 14. Architecture de von Neumann

---



## Principes de fonctionnement des ordinateurs

Jonas Lätt

Centre Universitaire d'Informatique





# Contenu du cours

## Partie I: Introduction

1. Introduction

2. Histoire de l'informatique

3. Information digitale et codage de l'information

4. Codage des nombres entiers naturels

5. Codage des nombres entiers relatifs

6. Codage des nombres réels

7. Codage de contenu média

8. Portes logiques

9. Circuits logiques combinatoires et algèbre de Boole

10. Réalisation d'un circuit combinatoire

11. Circuits combinatoires importants

12. Principes de logique séquentielle

13. Réalisation de la bascule DFF

**14. Architecture de von Neumann**

15. Réalisation des composants

16. Code machine et langage assembleur

17. Architecture d'un processeur

18. Performance et micro-architecture

19. Du processeur au système

## Partie II: Codage de l'information

## Partie III: Circuits logiques

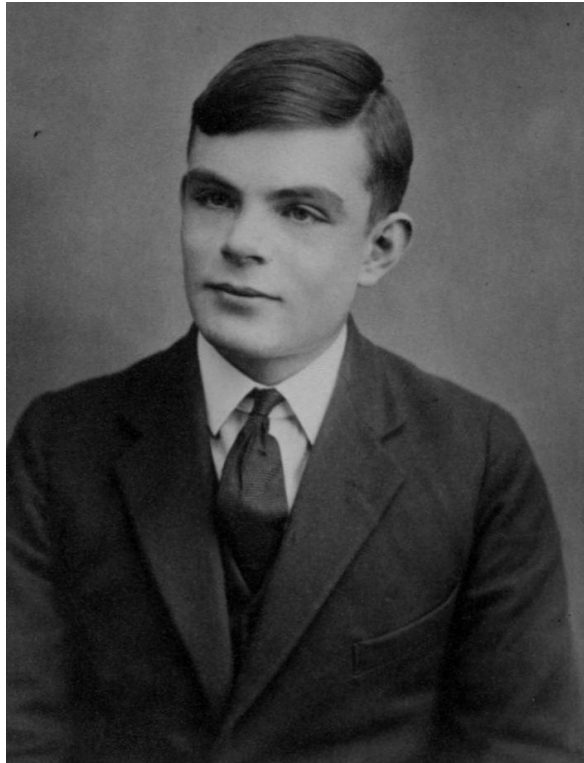
## Partie IV: Architecture des ordinateurs

# Universalité

---



Théorie: Machine  
Universelle de Turing



*Alan Turing*

Pratique: Architecture de  
von Neumann

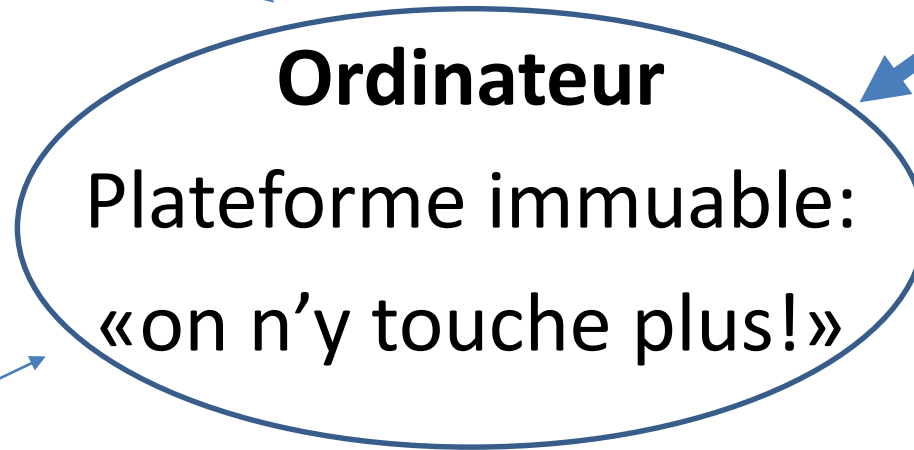


*John von Neumann*

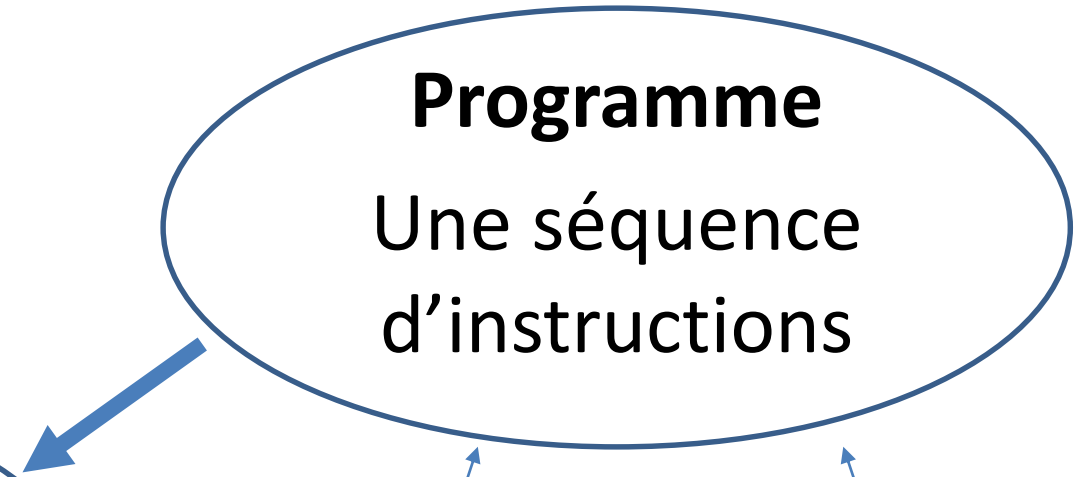
# Idée sous-jacente: architecture à programme stocké



Capable d'effectuer «n'importe quelle tâche», de la bureautique au calcul scientifique.



Capable d'exécuter, de manière répétée, des instructions tirées d'un **jeu d'instructions**.



**Stocké** dans une **mémoire de travail** de l'ordinateur

Origine de l'idée du programme stocké: diverses personnes dans les années 1930.

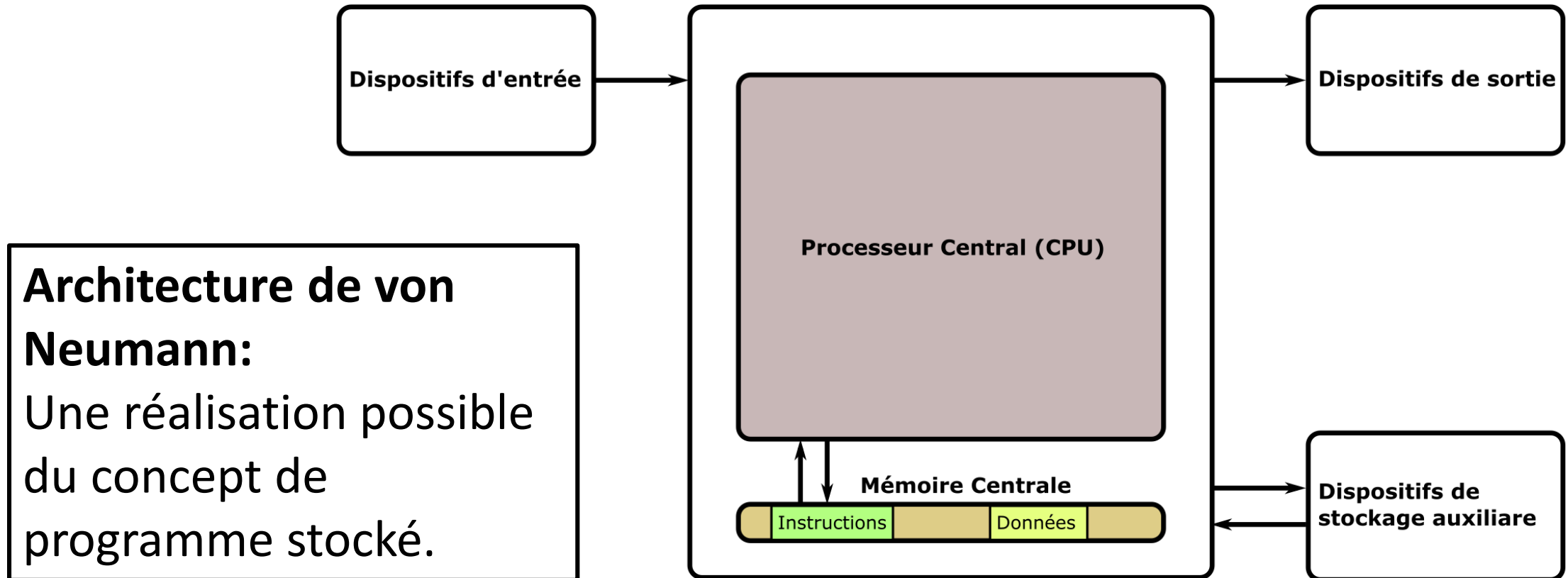
# 1950: L'ordinateur Edvac

---



- Successeur de Eniac, projet supervisé par Eckert et Mauchly.
- Commandité par l'armée américaine dans le cadre du projet Manhattan.
- Contributeur: le mathématicien John von Neumann, auteur d'une publication sur l'architecture de Eniac.
- *L'architecture de von Neumann* est, dans les grandes lignes, toujours utilisée dans les ordinateurs modernes.

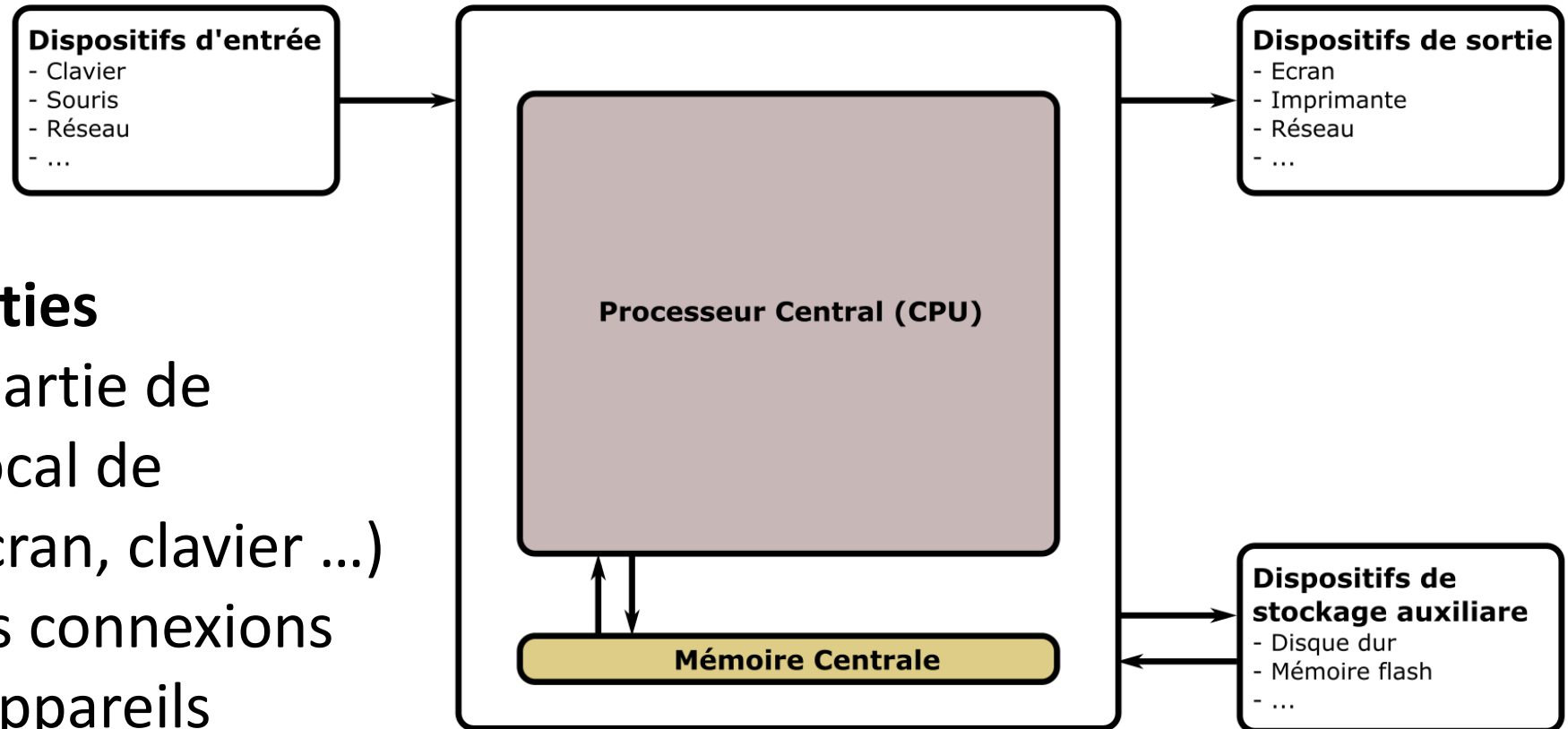
# Architecture de von Neumann





# Éléments de l'architecture de von Neumann

# Entrées / Sorties



## Les entrées / sorties

- Peuvent faire partie de l'écosystème local de l'ordinateur (écran, clavier ...)
- Permettent des connexions avec d'autres appareils (Bluetooth, WiFi, ...)

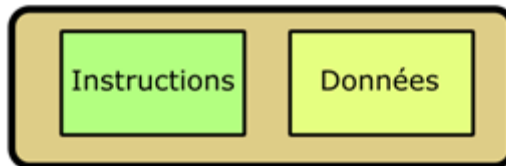


# Deux types de mémoire

---



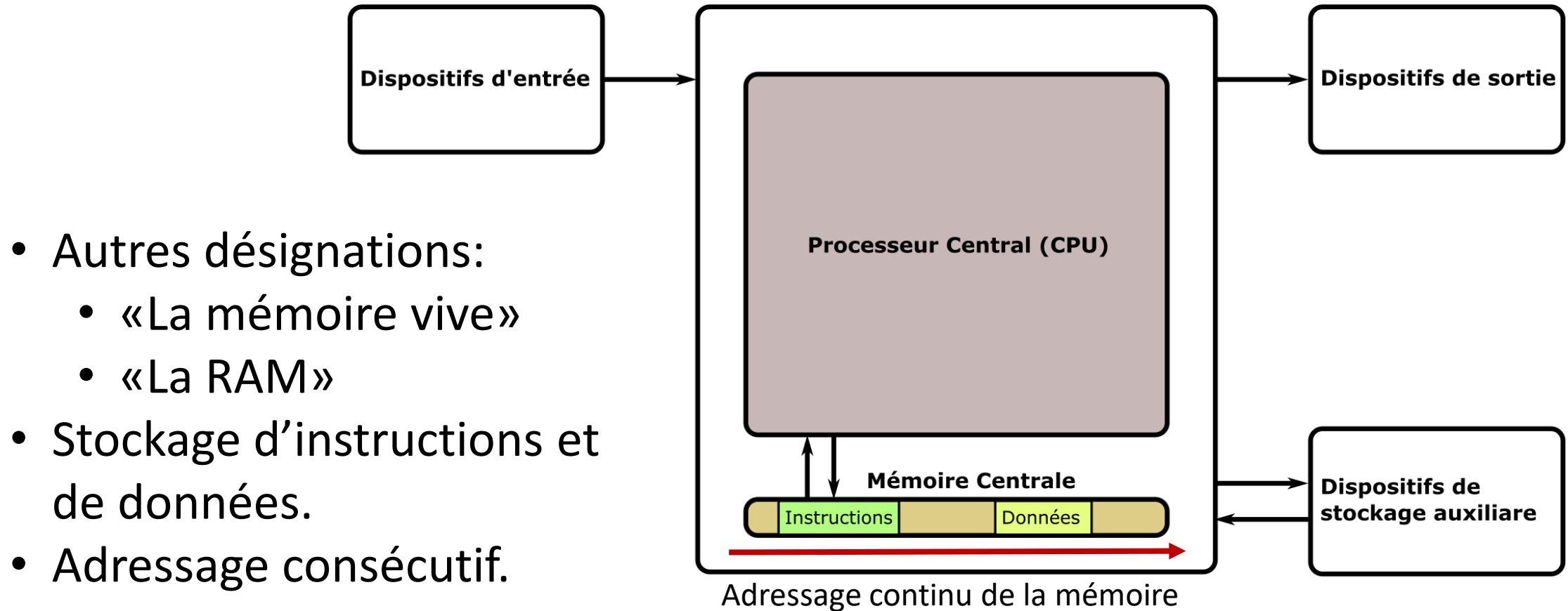
## Mémoire Centrale



## Dispositifs de stockage auxiliaire

- Disque dur
- Mémoire flash
- ...

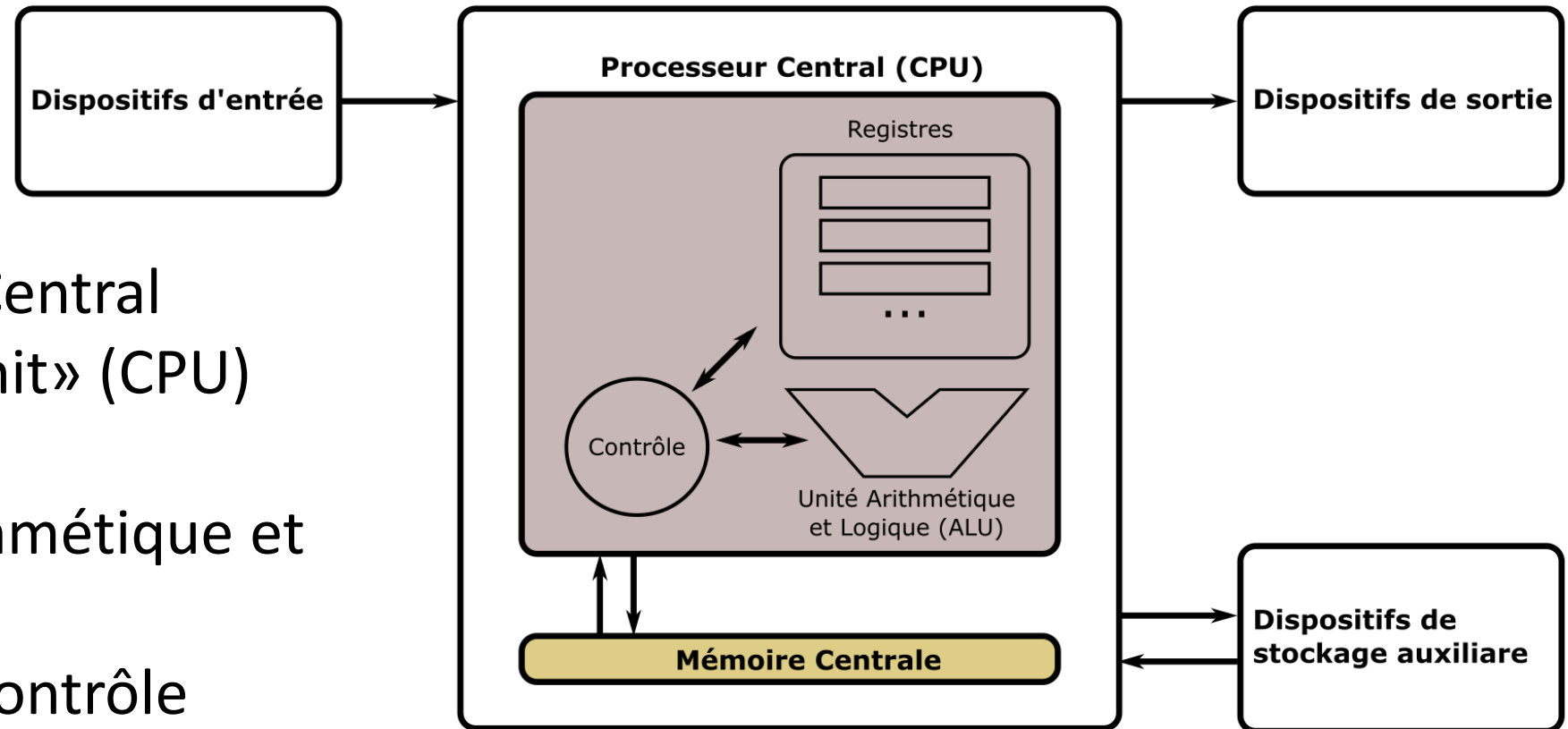
# Mémoire Centrale



# Le Processeur Central (CPU)



- En anglais: «Central Processing Unit» (CPU)
- Composants:
  - Unité arithmétique et logique
  - Unité de contrôle
  - Registres

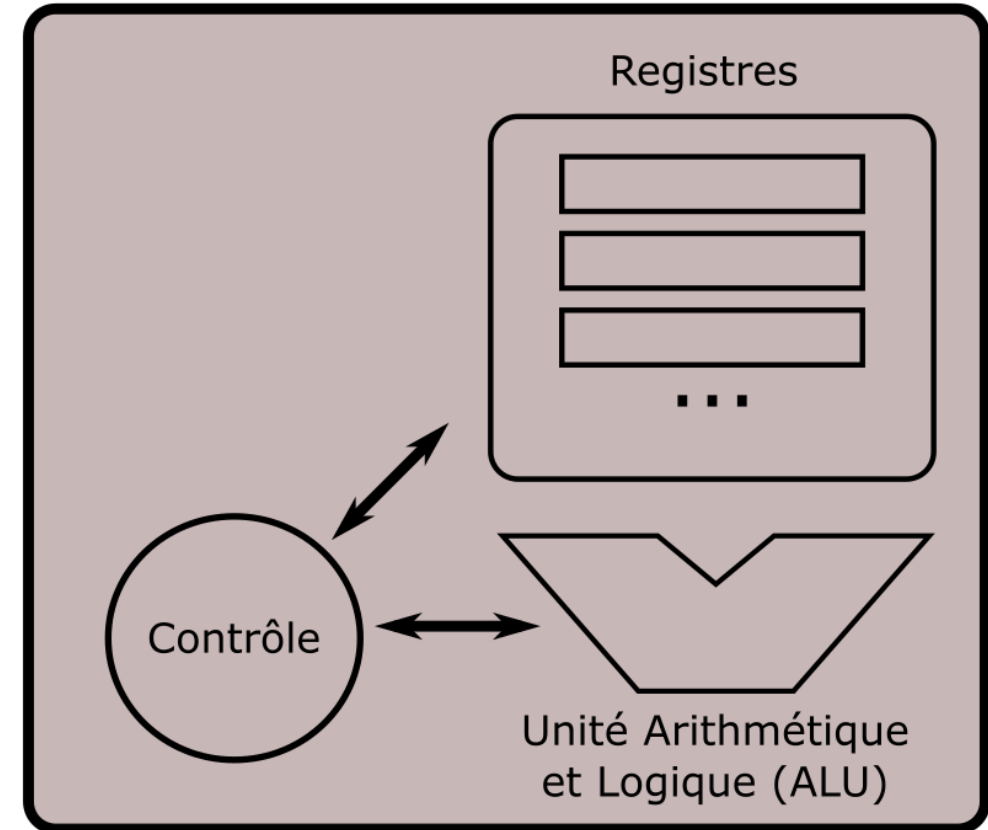


# Le CPU – Registres

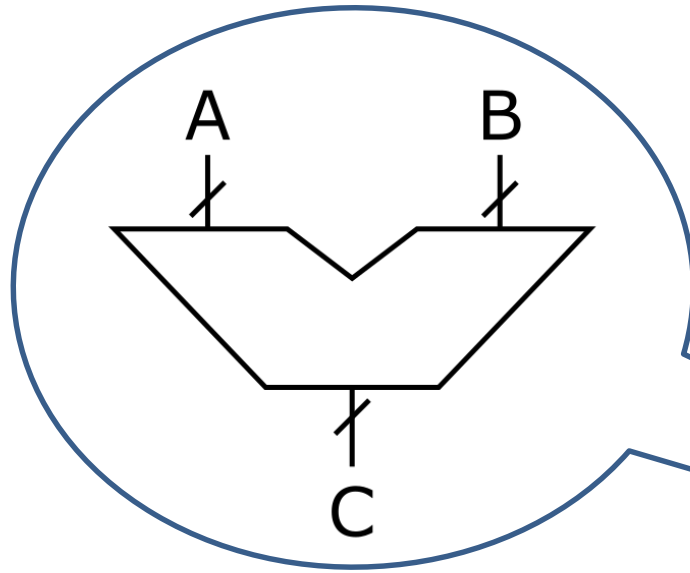


- Petites unités de **mémoire très rapides**.
- Permettent de mémoriser des **valeurs temporaires** en cours d'utilisation.
- Exemple: une valeur récupérée de la mémoire centrale.
- Exemple: un résultat intermédiaire de plusieurs opérations arithmétiques.

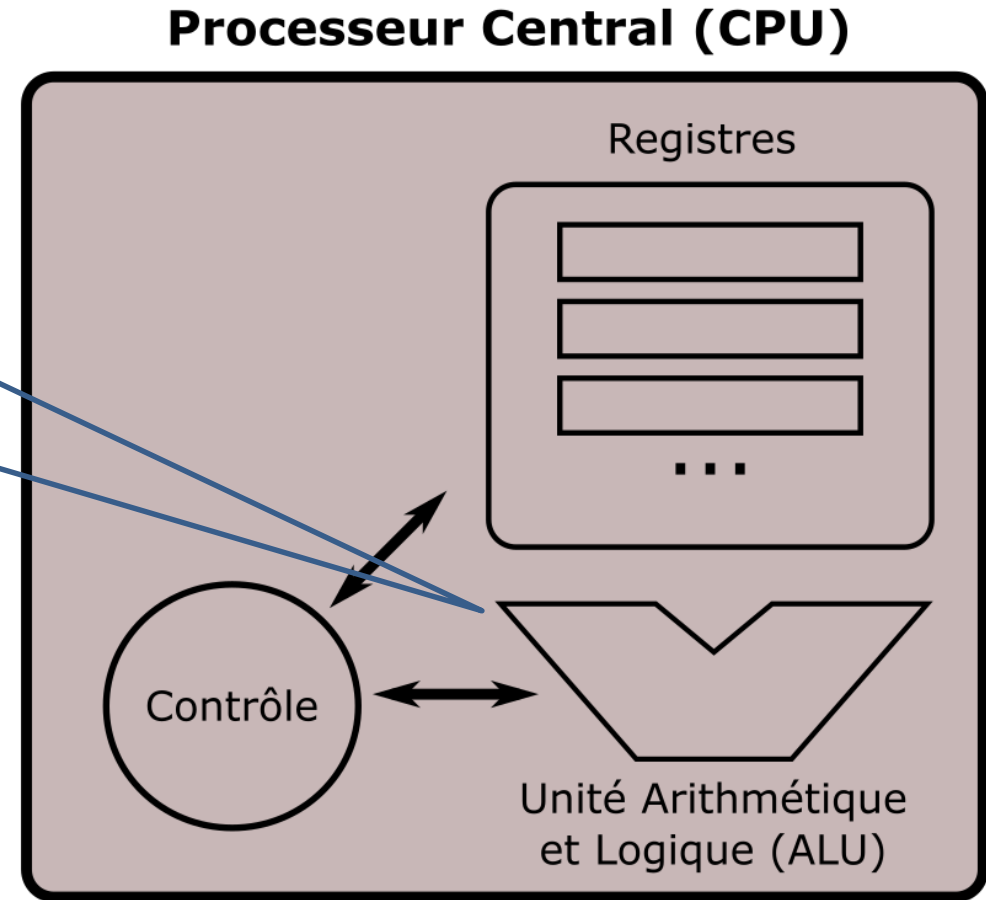
## Processeur Central (CPU)



# Le CPU – Unité arithmétique et logique (ALU)



- En anglais:  
«Arithmetic Logic Unit» (ALU)
- **Entrée:** Deux mots provenant de registres
- **Sortie:** Un mot, résultat d'une opération arithmétique (addition, ...) ou logique (opérateur AND, ...).

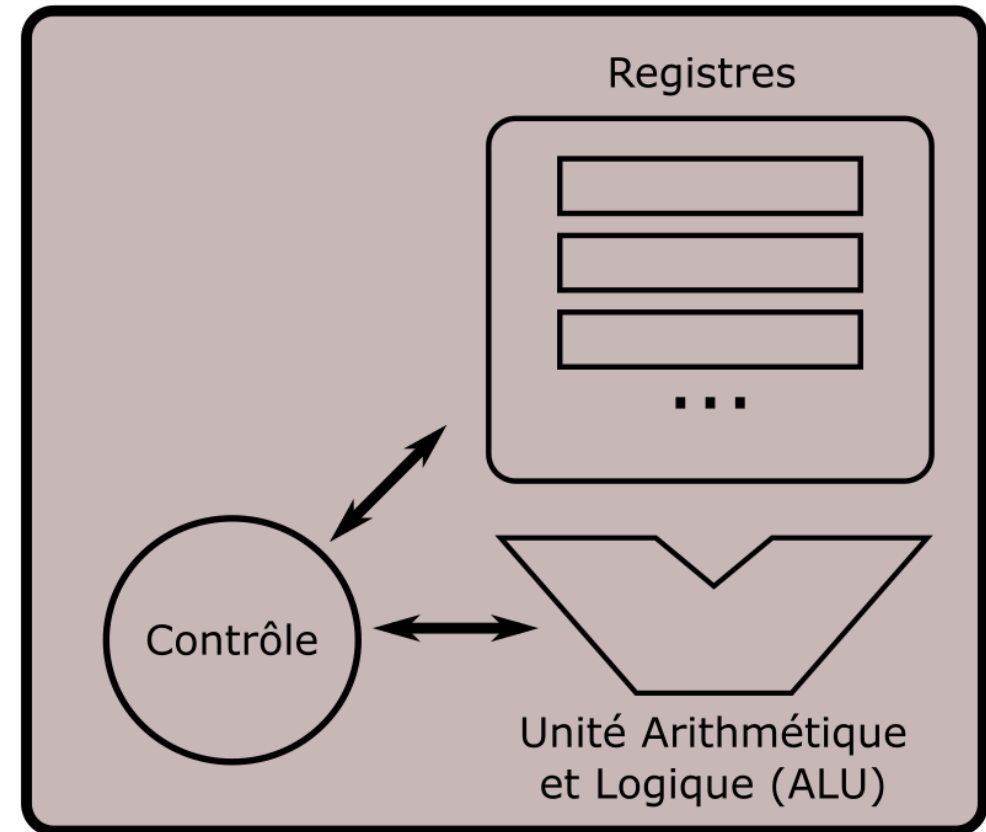


# Le CPU – Unité de Contrôle



- Commande et contrôle l'opération du CPU.
- Transmet des signaux de contrôle aux différents dispositifs du CPU.
- Responsable du bon déroulement des cycles FETCH-DECODE-EXECUTE.

## Processeur Central (CPU)



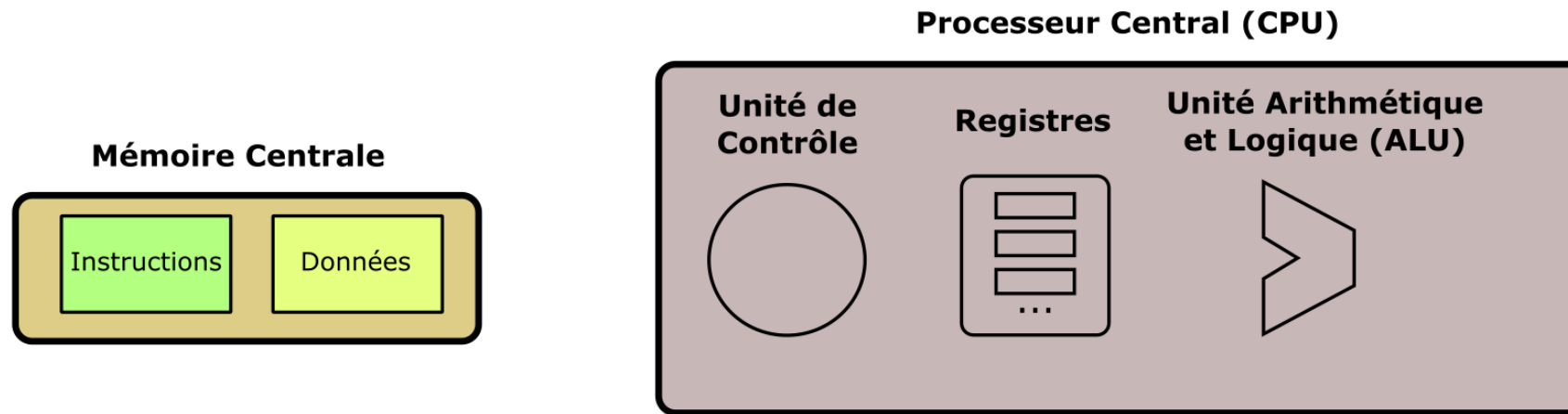
# 17. Architecture de von Neumann

---



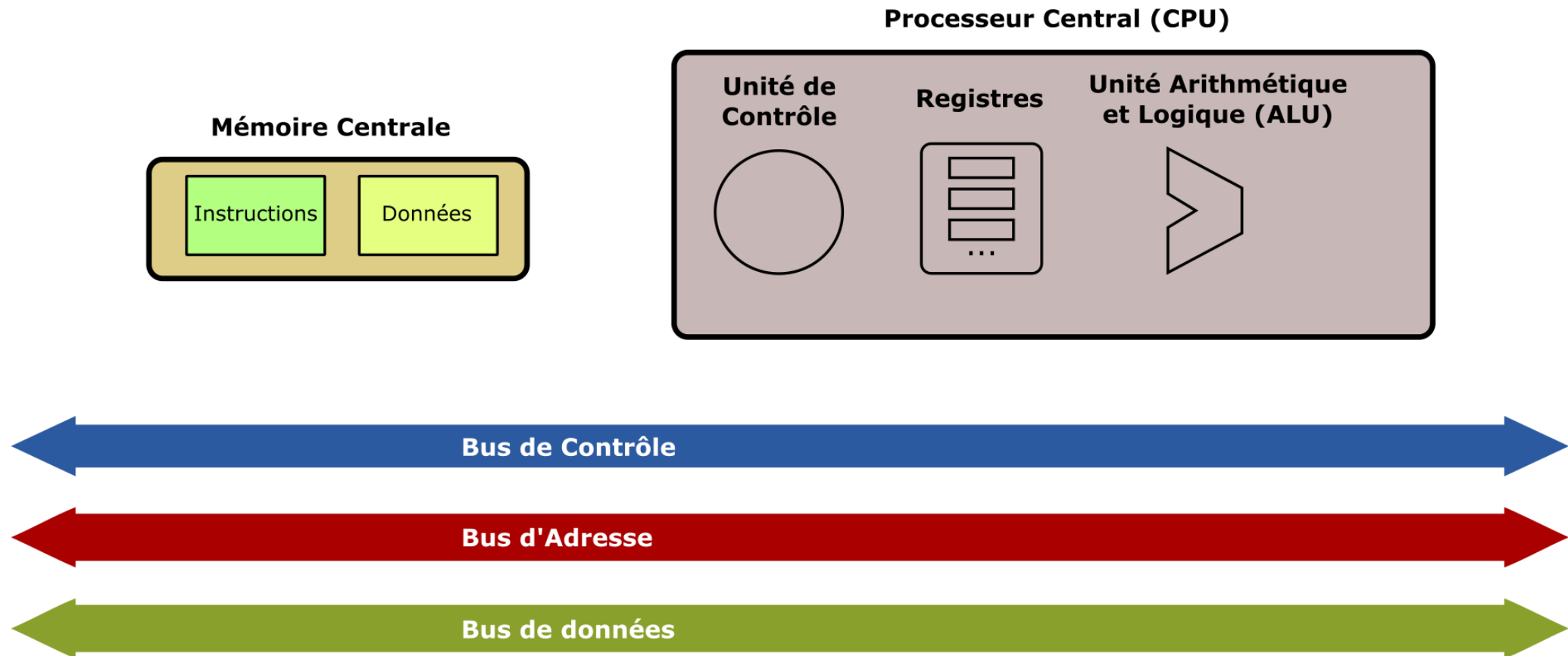
## Les flux d'information

# Éléments de l'architecture et communication



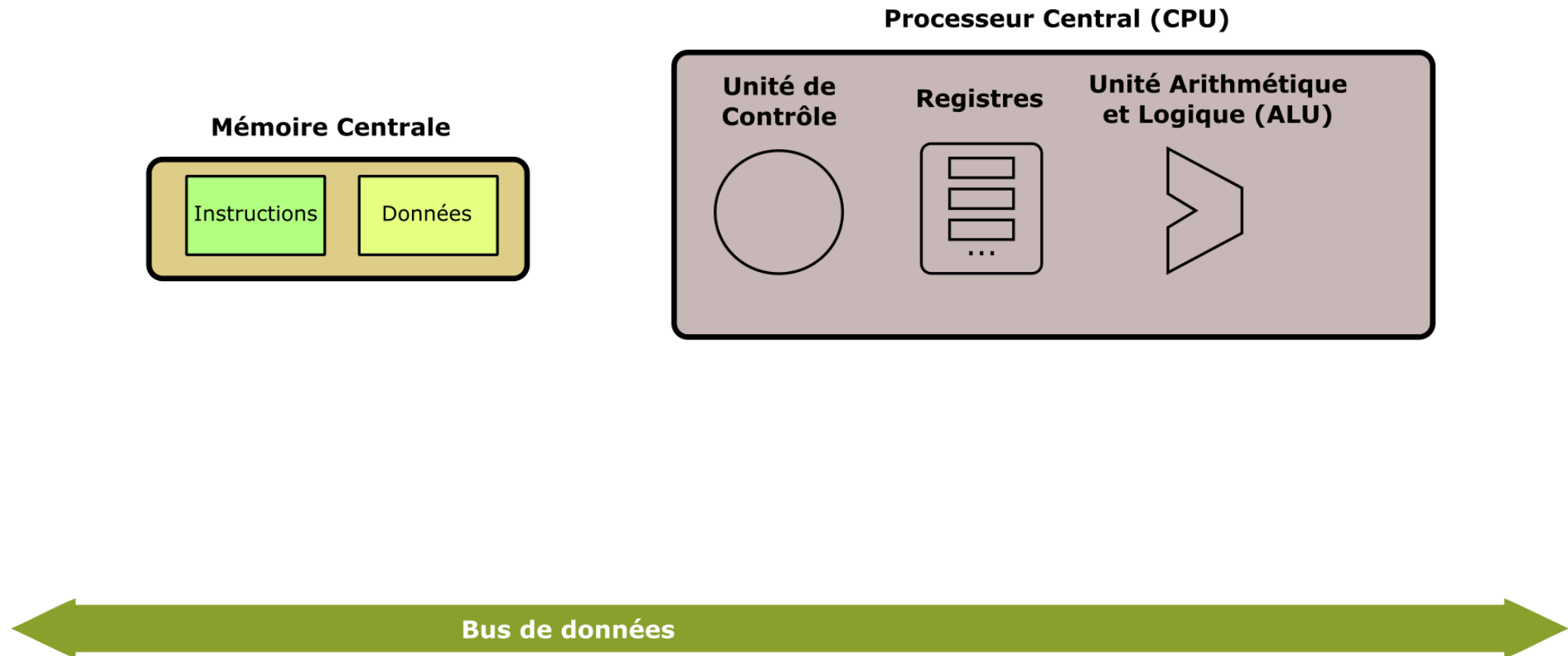


# Les systèmes de communication (Bus)

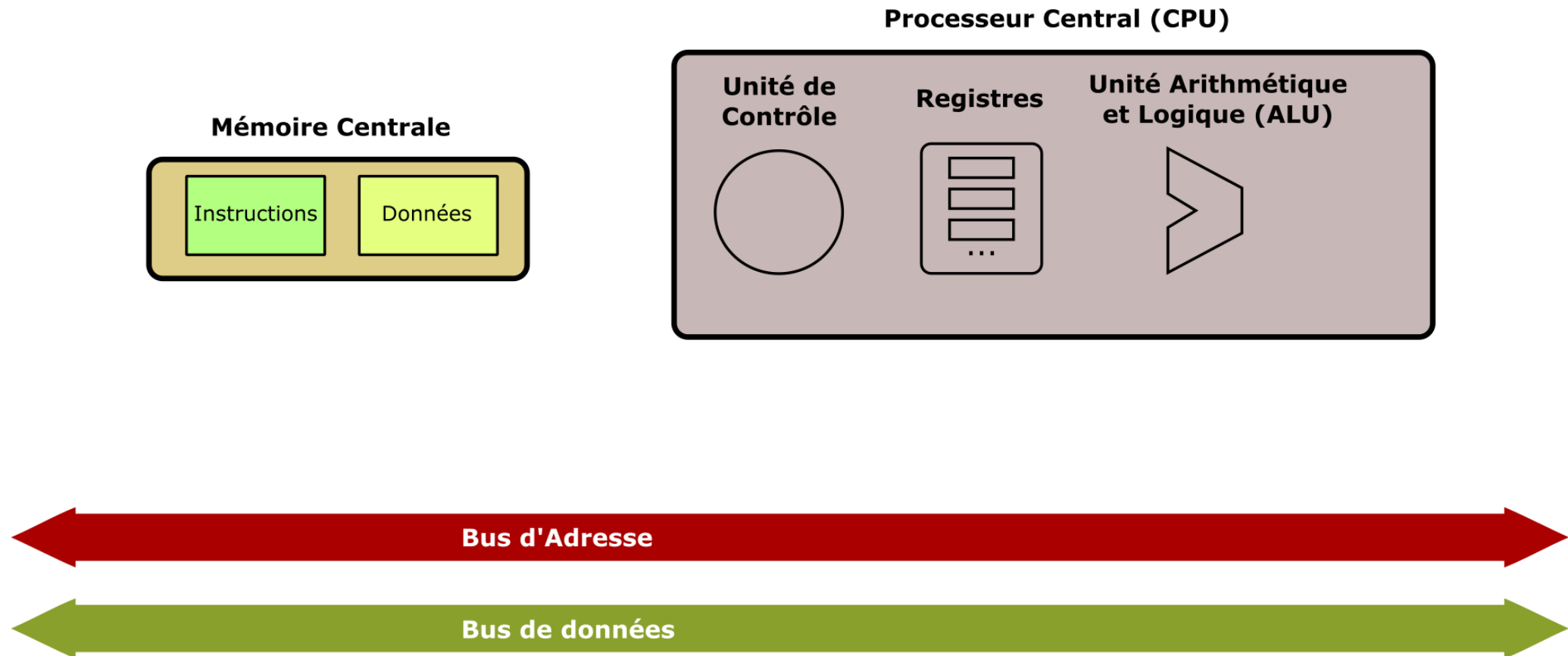




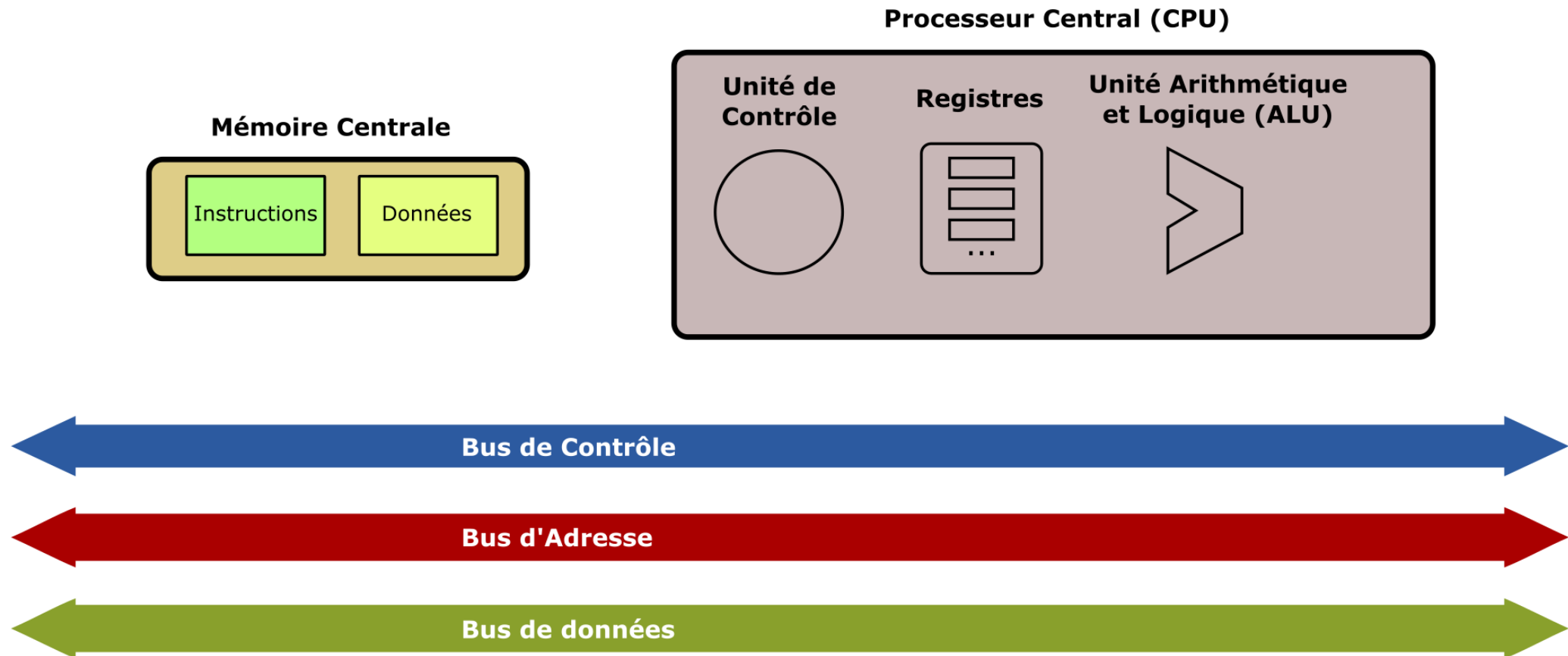
# Le Bus de données



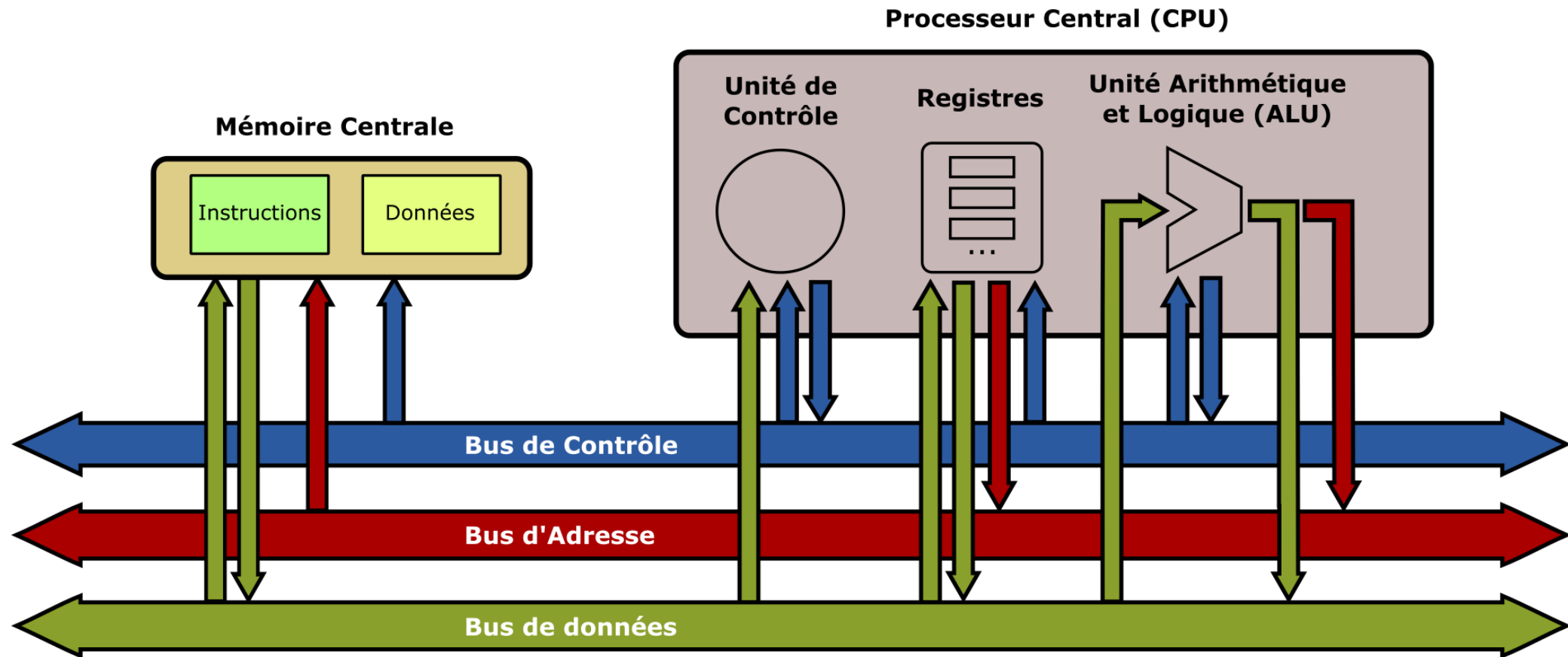
# Le Bus d'adresse



# Le Bus de contrôle



# Les systèmes de communication (bus)



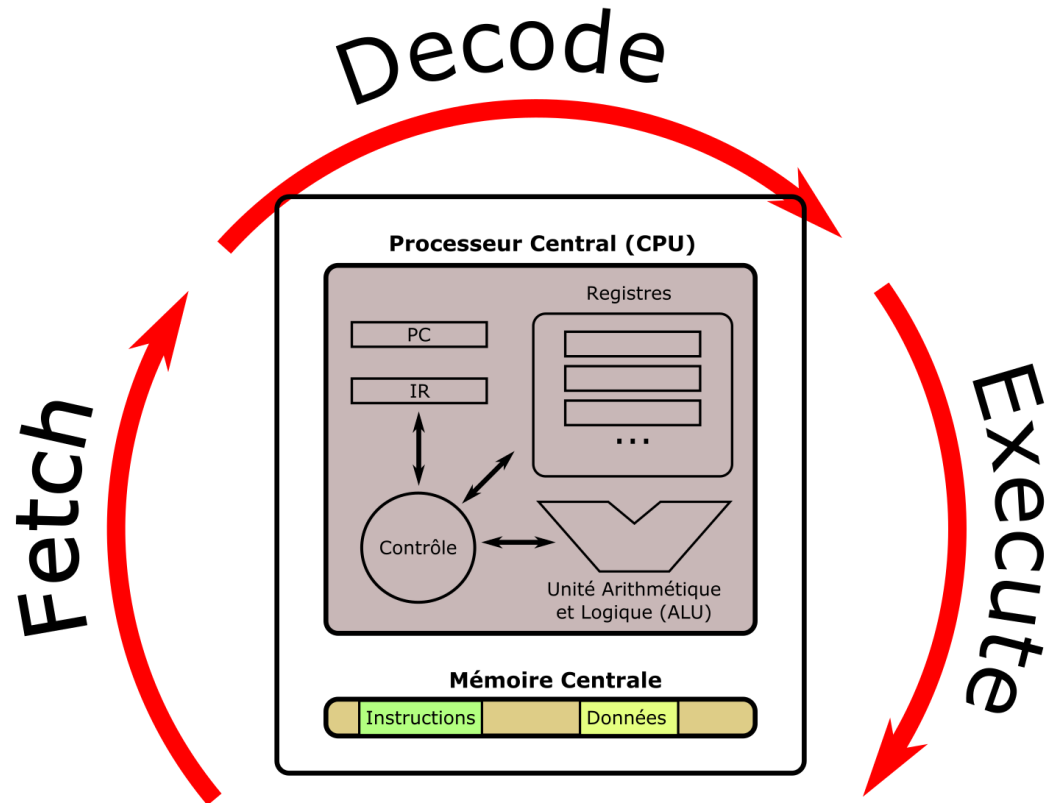
# 17. Architecture de von Neumann

---



## Les cycles Fetch-Decode-Execute

# Les cycles Fetch-Decode-Execute



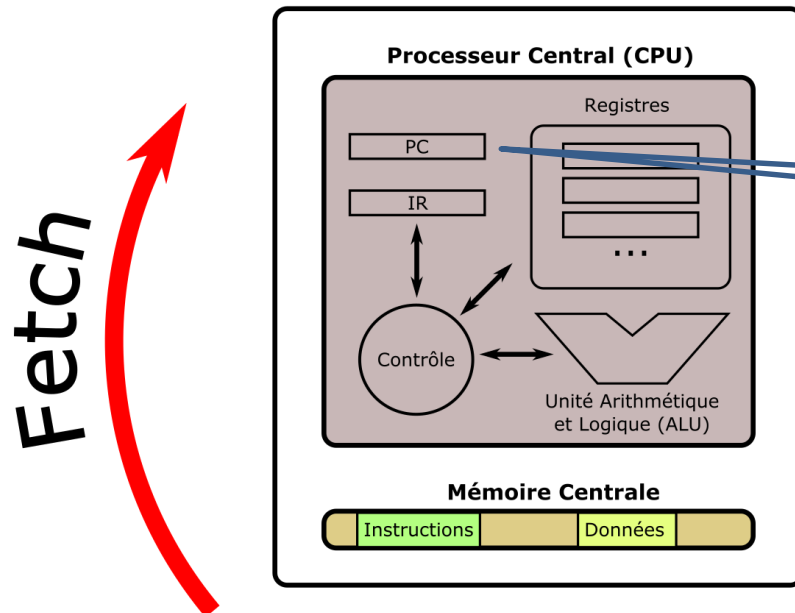
- Circuit **synchrone**, rythmé par une **horloge**.
- Exécution **séquentielle** et **cyclique**.
- Cycle en 3 étapes
  1. Fetch
  2. Decode
  3. Execute
- Cycle orchestré par l'**unité de contrôle**.

# Les cycles Fetch-Decode-Execute: FETCH



FETCH: le processeur lit la prochaine instruction dans la mémoire centrale.

Le registre **PC**: «Program Counter»  
(**Compteur de programme**)  
Adresse de l'a prochaine instruction.



Exécution normale:  
PC est incrémenté à chaque cycle.

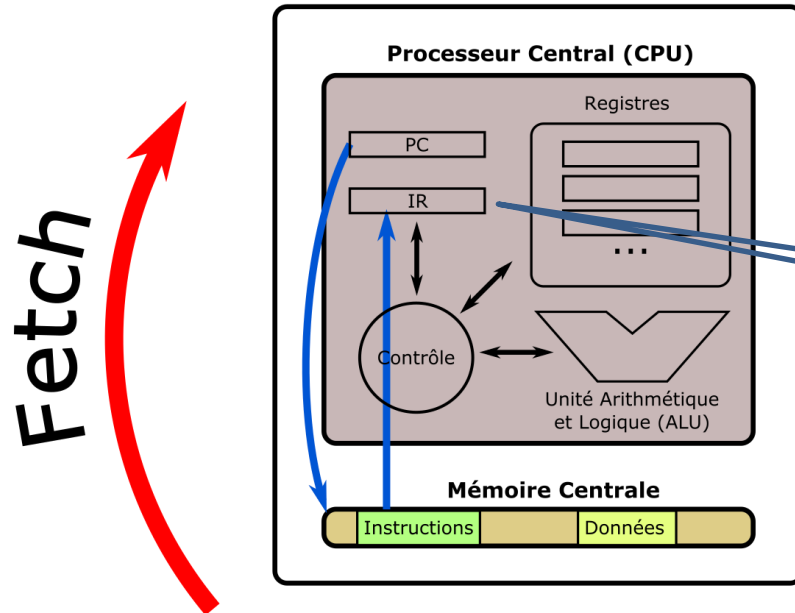
«Saut», p.ex. branchement:  
Chargement d'une nouvelle valeur dans le PC.



# Les cycles Fetch-Decode-Execute: FETCH



FETCH: le processeur lit la prochaine instruction dans la mémoire centrale.



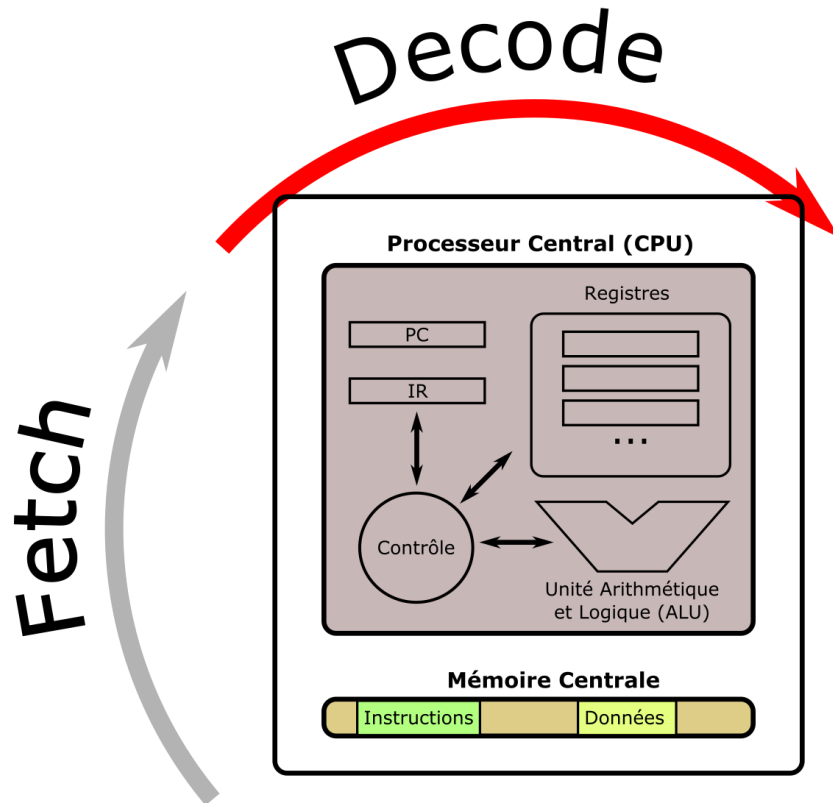
Le registre **IR**: «Instruction Register»  
**(Registre d'instruction)**  
L'instruction lue en mémoire y est déposée.

IR

# Les cycles Fetch-Decode-Execute: DECODE



DECODE: Le CPU **interprète** l'instruction dans le registre IR.



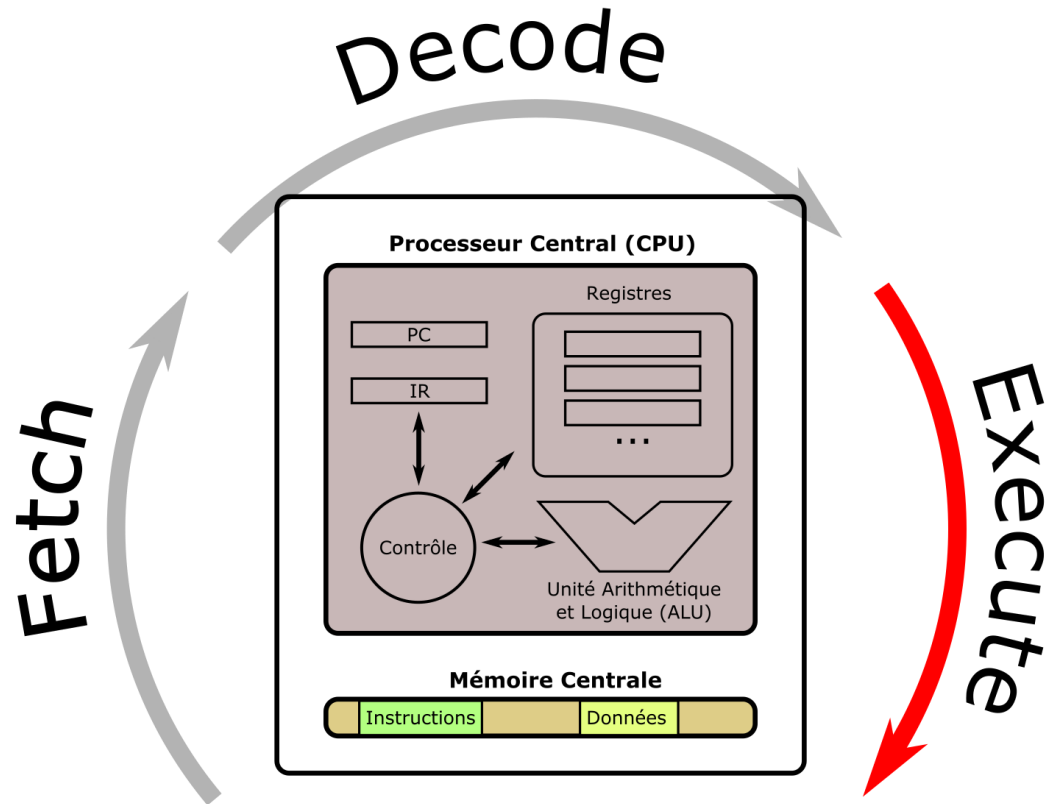
Quelle est la tâche demandée? Exemples:

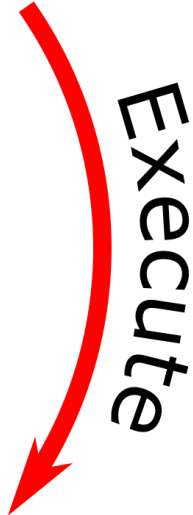
- Mettre à jour le PC (Saut inconditionnel).
- Charger une valeur dans un registre.
- Effectuer un calcul.

L'unité de contrôle envoie les signaux de contrôle aux autres dispositifs:

- La mémoire centrale
- L'unité arithmétique et logique
- Les registres

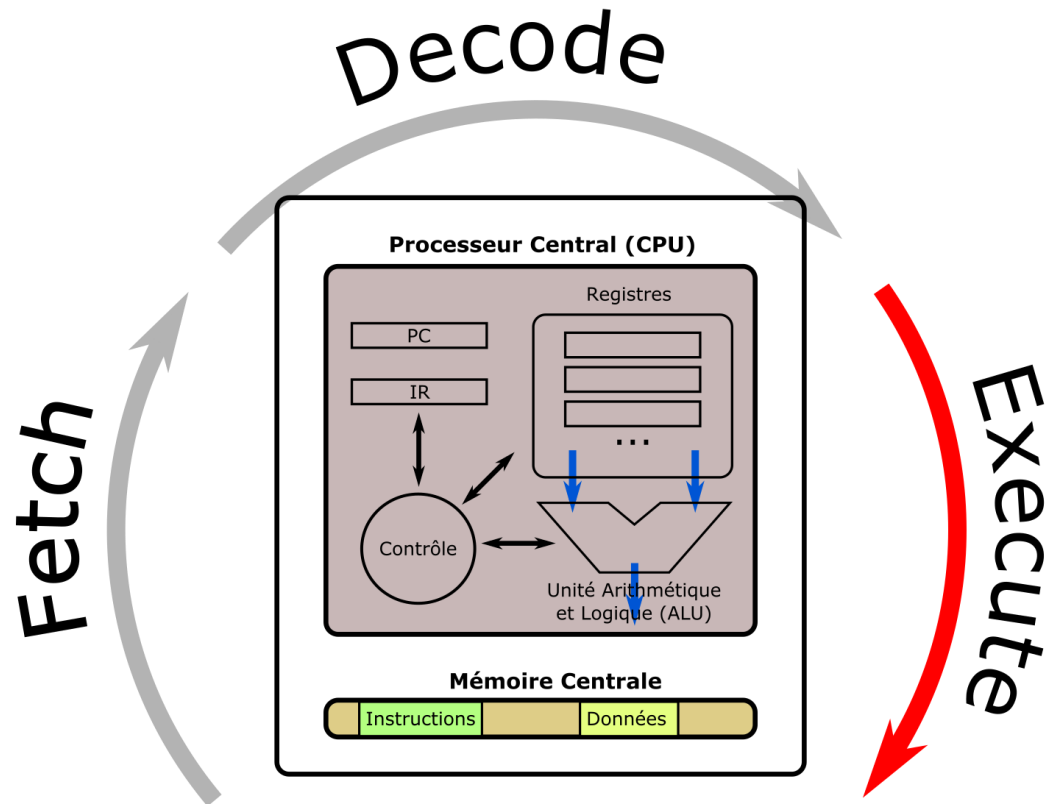
# Les cycles Fetch-Decode-Execute: EXECUTE





- Execute

# Les cycles Fetch-Decode-Execute: EXECUTE



Exemple: effectuer un calcul.

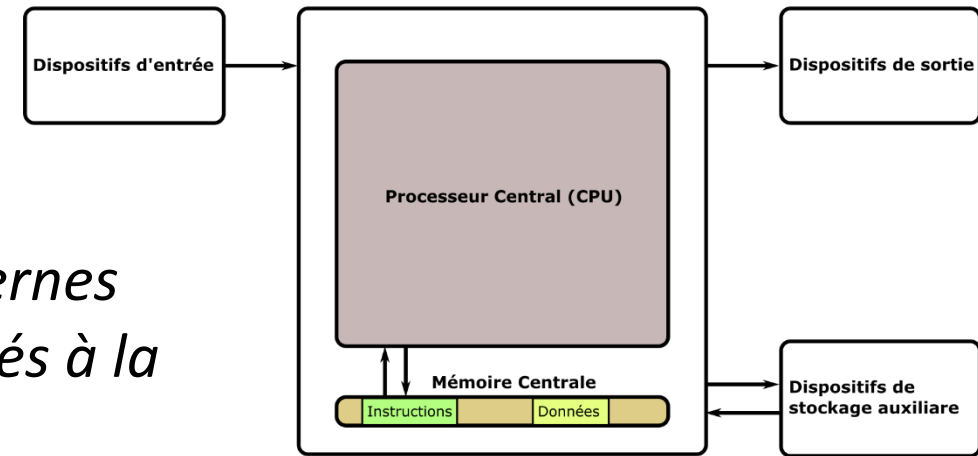
- Addition / soustraction.
- Opération Booléenne bit-par-bit.
- Comparaison ( $A > B$ ?)
- ...
- Le calcul est effectué par l'ALU.

# Commentaires sur l'architecture de von Neumann



**Question:** Un processeur moderne est-il vraiment l'implémentation exacte de l'architecture von Neumann?

**Réponse:** *Non, la plupart des processeurs modernes sont **multi-cœurs**. Plusieurs CPUs sont connectés à la mémoire centrale.*



**Question:** L'architecture de von Neumann est-elle la seule implémentation du concept de machine à programme stocké qu'on ait réalisée?

**Réponse:** *Non, certains ordinateurs ont été construits selon **l'architecture de Harvard**. Dans ce cas on a deux mémoires distinctes, une pour les instructions et une pour les données.*

# Exercice instantané

---



Quel type d'information récupère-t-on dans la mémoire lors d'une étape *Fetch*?

1. La prochaine instruction à exécuter.
2. Des données (par exemple un nombre entier à additionner).
3. Une instruction ou une donnée, ça dépend.

# A venir: l'architecture RISC-V

---



- Nous avons présenté l'architecture de von Neumann d'une manière générale, qui est en gros valable pour n'importe quelle architecture moderne.
- Mais, en entrant dans les détails, on voit qu'il existe beaucoup d'architectures d'ordinateurs modernes, avec des différences importantes.
- On va choisir une architecture particulière: l'**architecture RISC-V**, et présenter ses détails.



