AE 7

Un modèle d'exécution
-
Von Neumann

Introduction

De quoi traite l'Informatique

- Computer Science

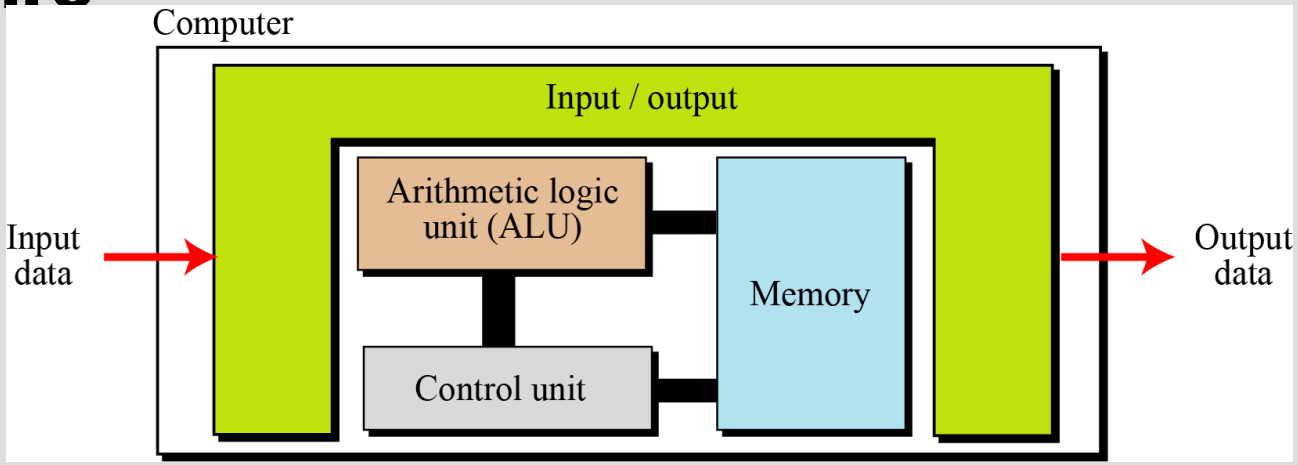
  compute = calculer

- Informatique = Information + Automatique

  traitement automatique de l'information

Modèle de Von Neumann

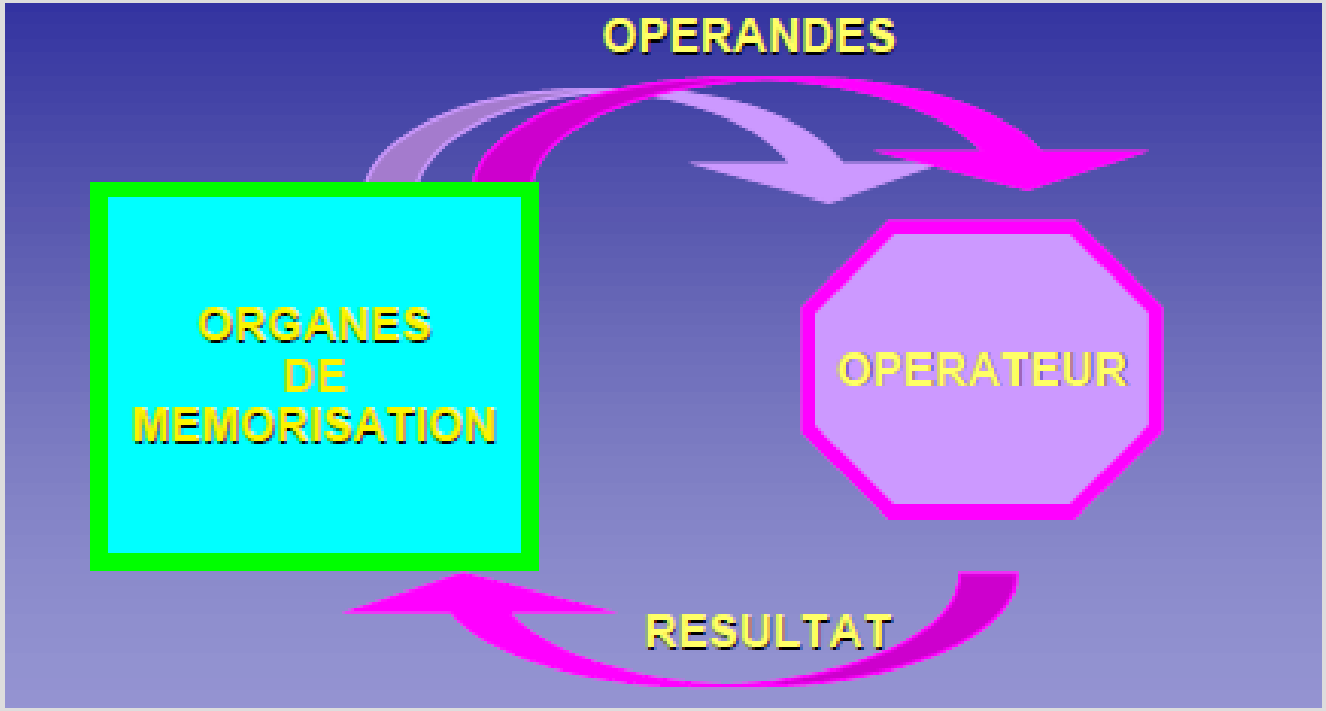
- **John Von Neumann 1943-45**
  - **Lignes essentielles pour construire une machine électronique. (Projet ENIAC)**
  - **Appliqués jusqu'à nos jours**
- **Quatre blocs fonctionnels:**
  - **Le processeur (ALU + Control Unit)**
  - **La mémoire**
  - **Le bus**
  - **Les I/O**



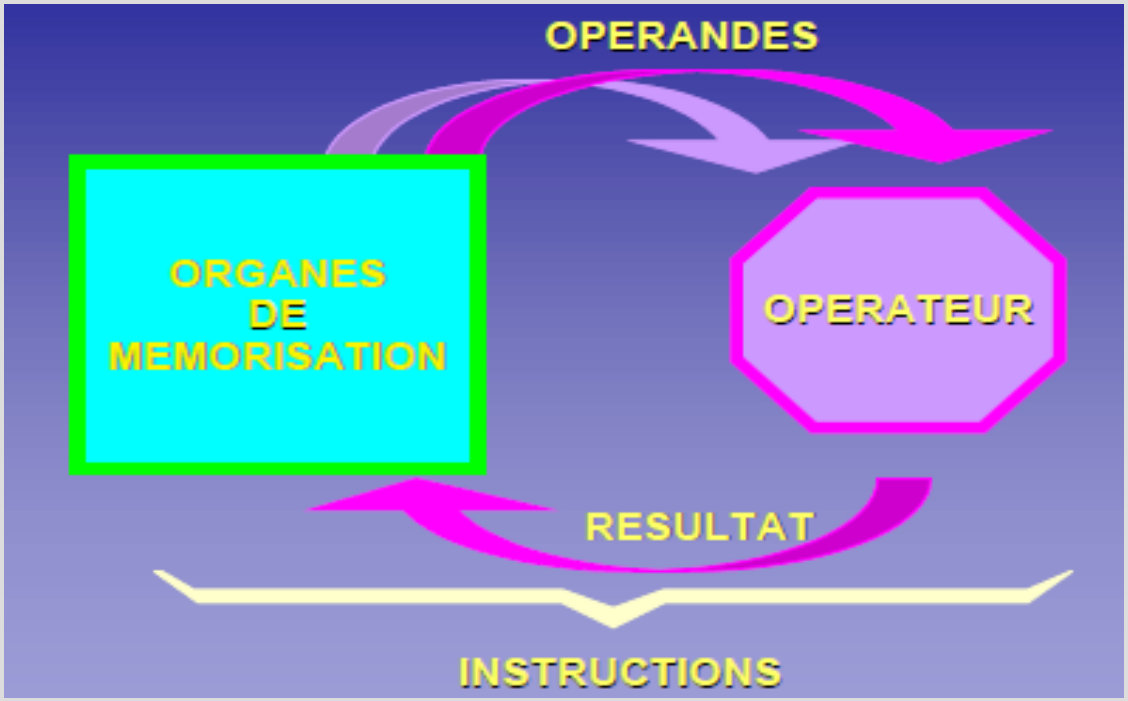
Du câblé au programmé..

- Premiers ordinateurs les pas d’exécution du programme étaient directement câblés dans le circuit
  - Aujourd'hui on parlerait d’accélérateur matériel
- On invente le programme vu comme des données stockées dans la mémoire principale
  - l’architecture de Von Neumann
- La fonction de l’unité de contrôle est de
  - lire le programme de la mémoire
  - décoder les instructions
  - commander leur exécution
- Un changement de programme se fait par une simple réécriture de la mémoire

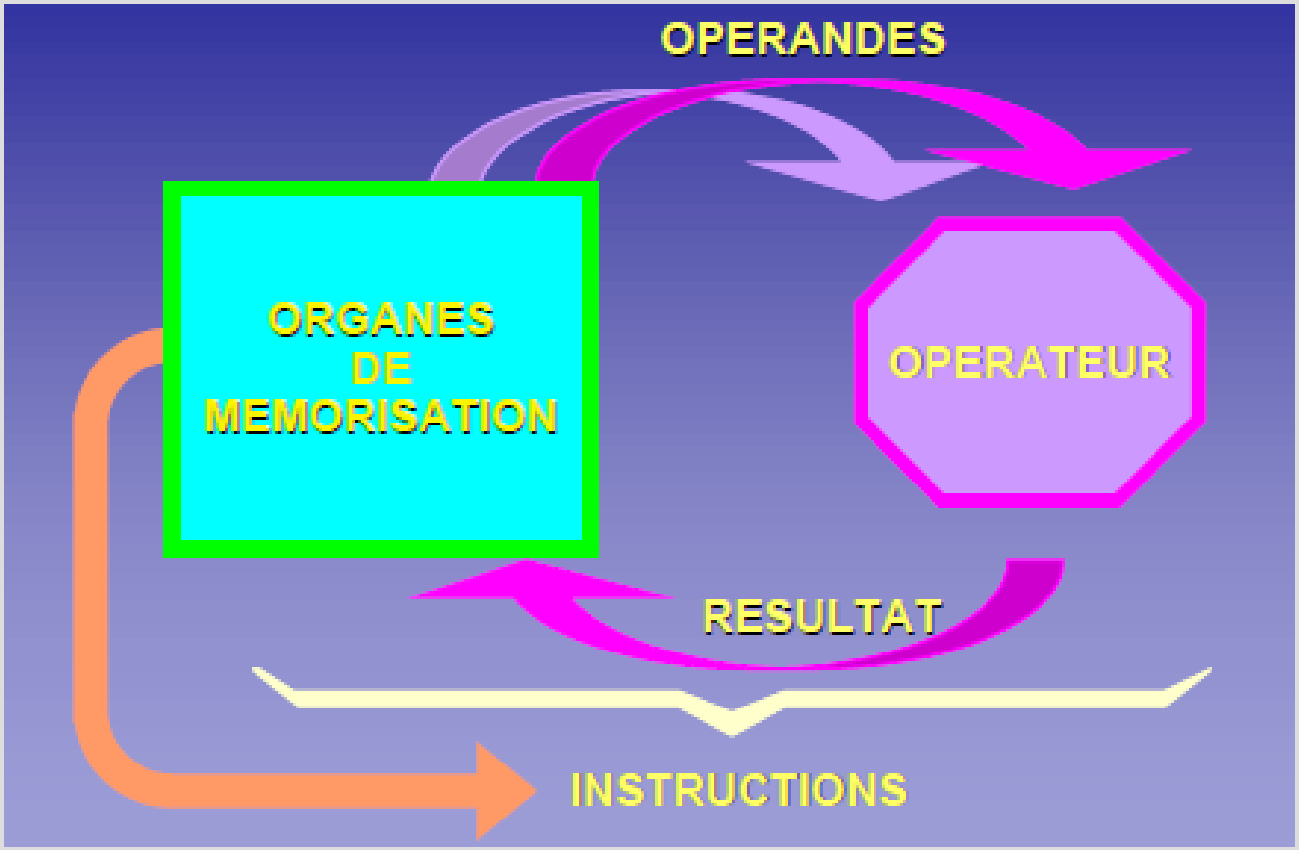
Du calcul sur des données…



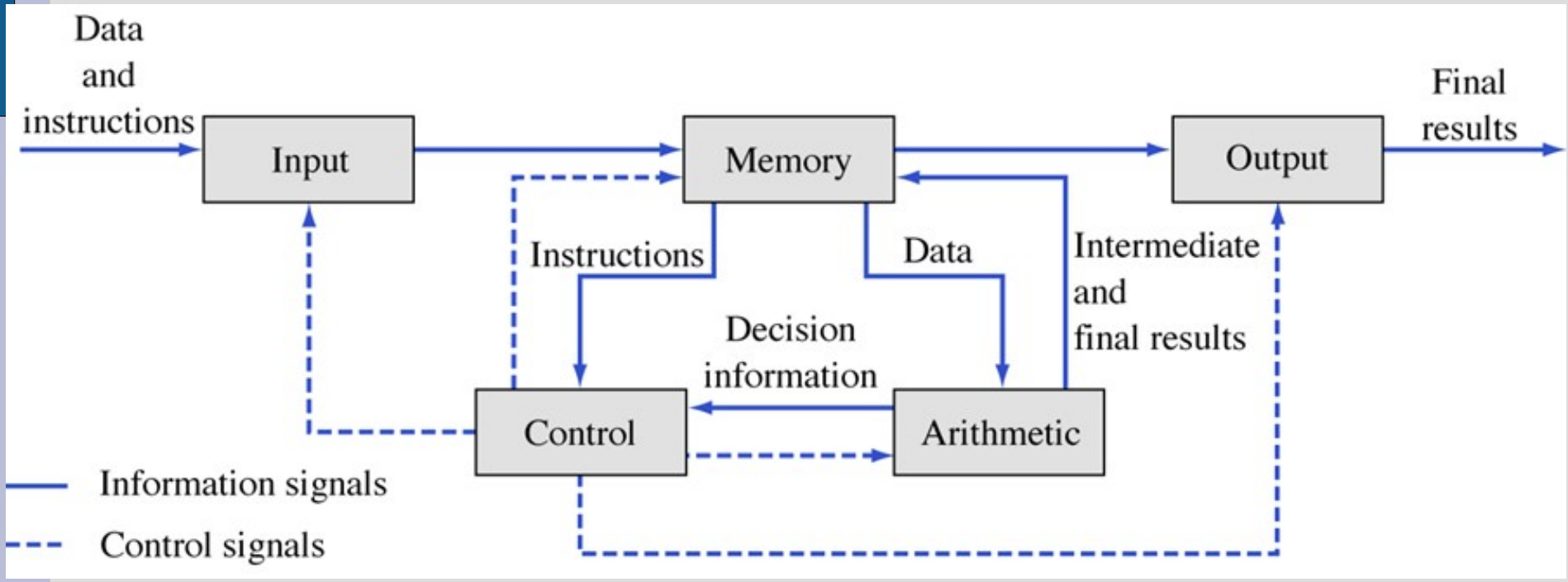
Notion d’instruction



Mémoire banalisée



Quels échanges dans la machine?



Zoom inside

Bascule mémoire

Décodeur

Fonction logique

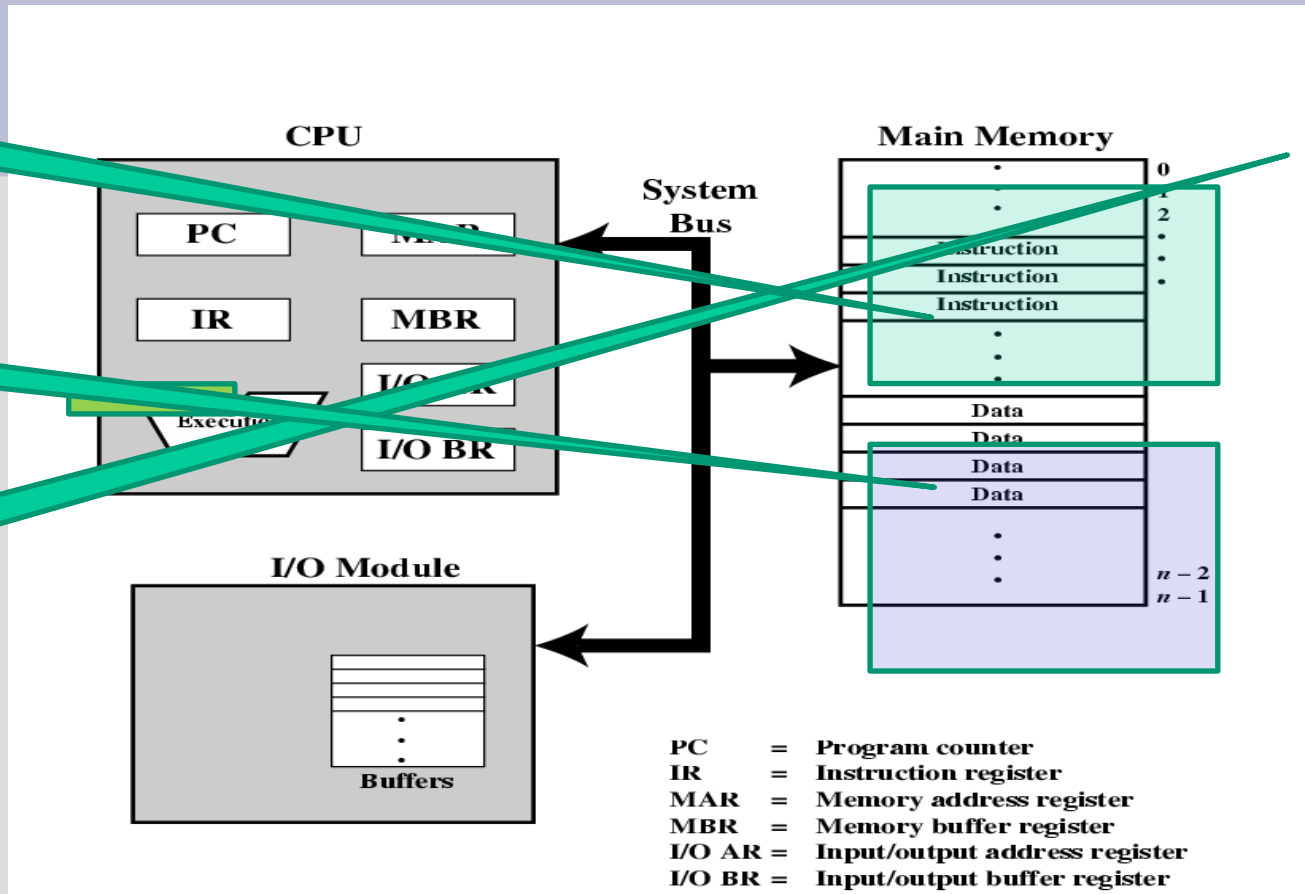
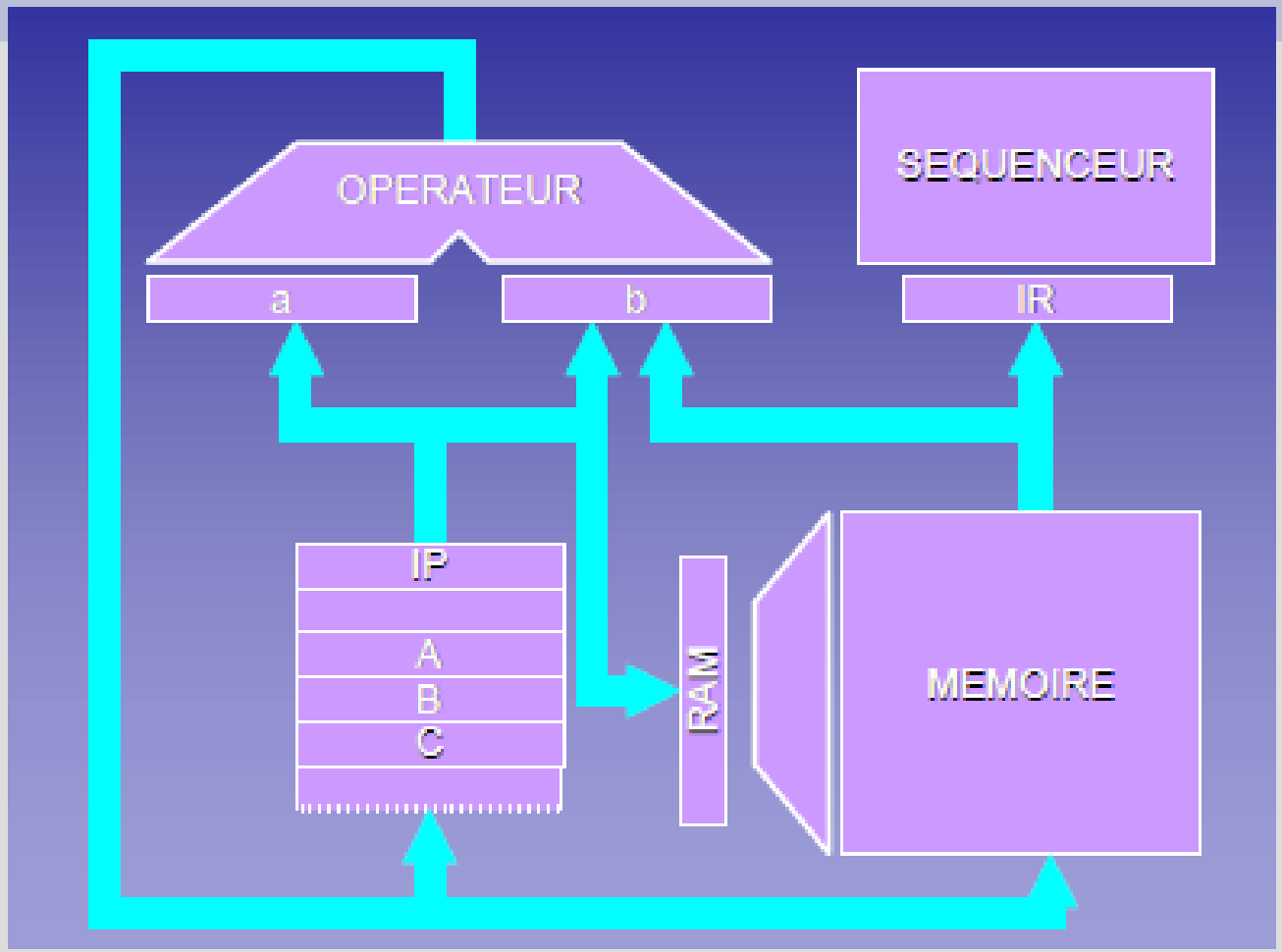


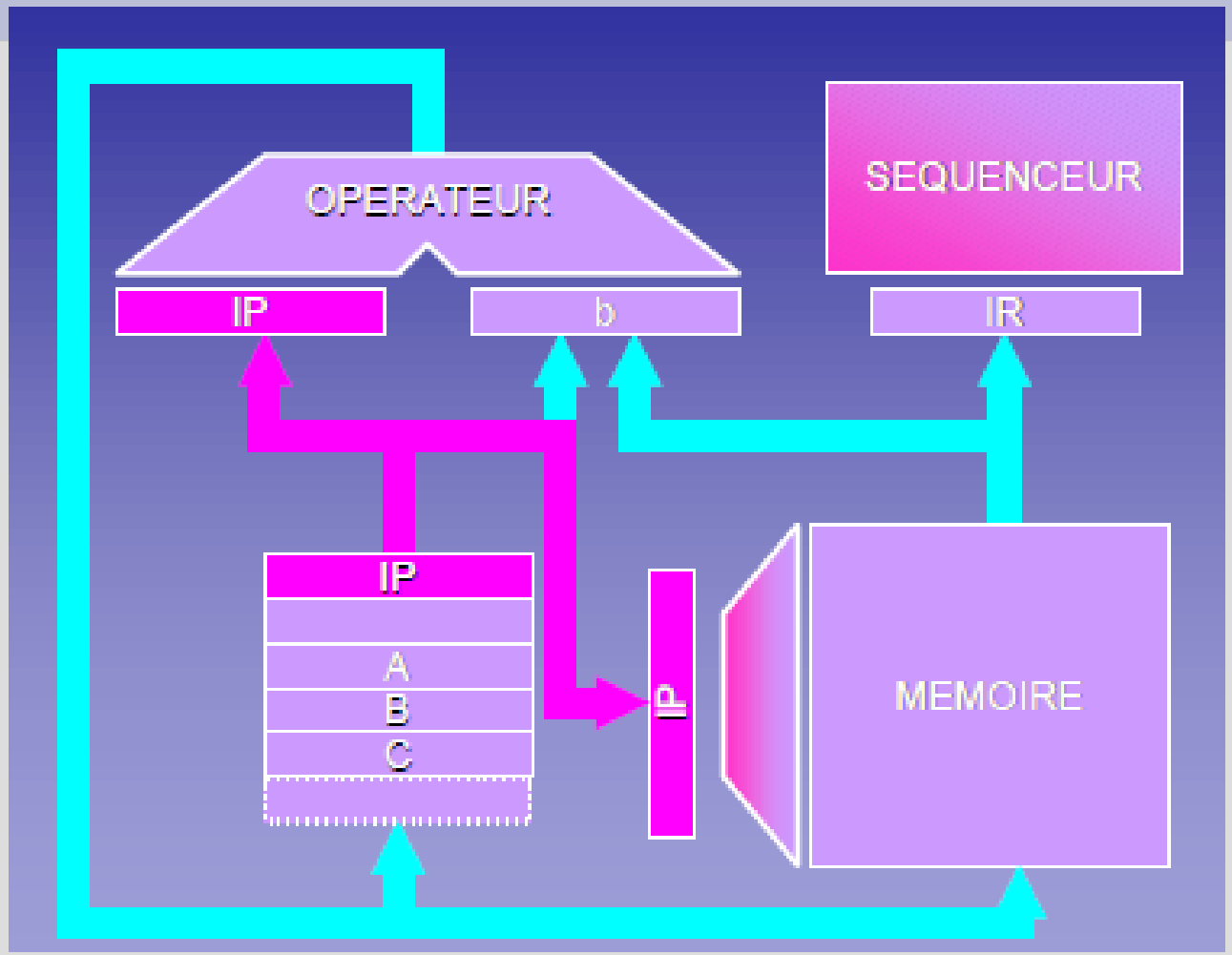
Figure 3.2 Computer Components: Top-Level View

Un ordinateur simplifié…

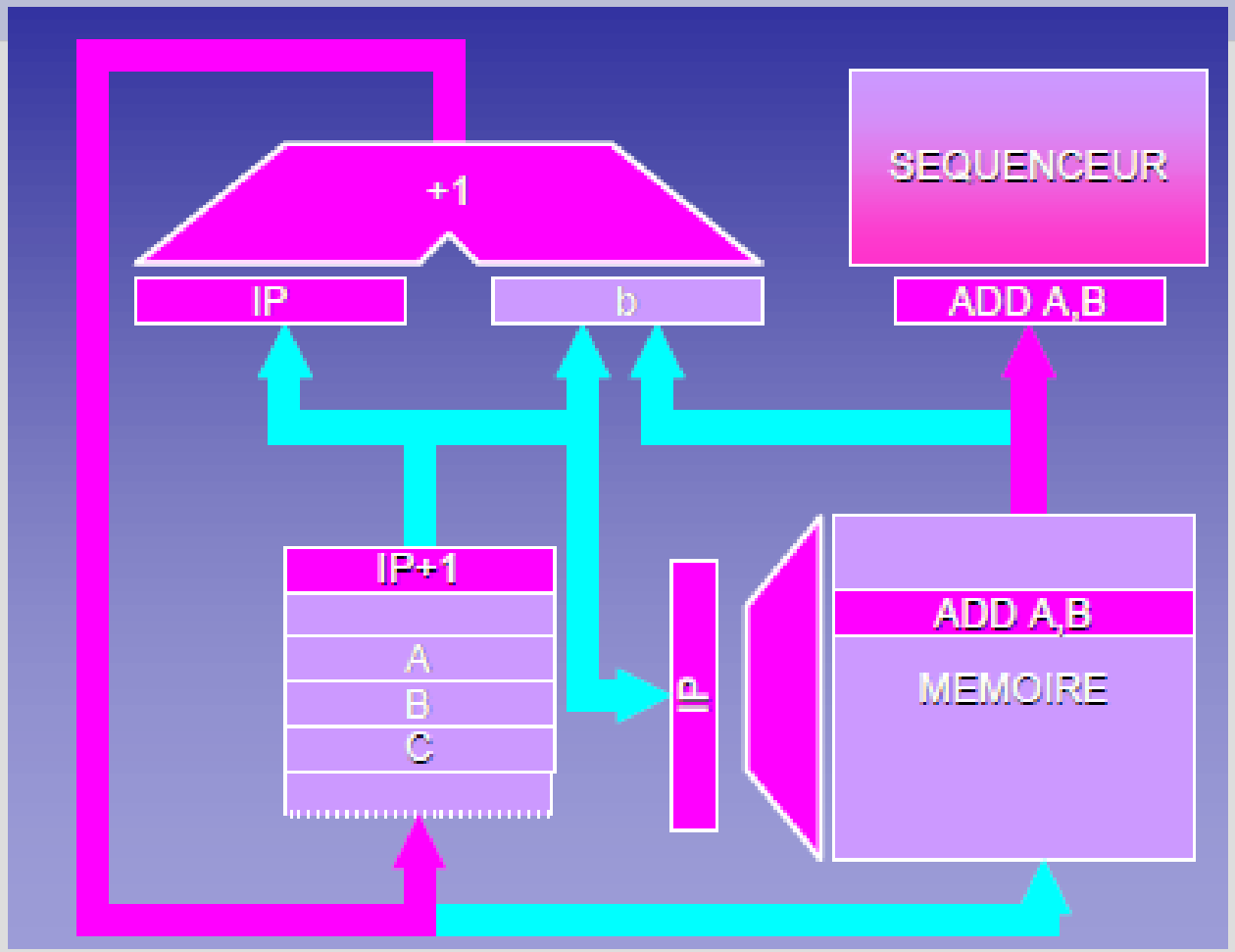
IP ~
CO~
CP~
PC



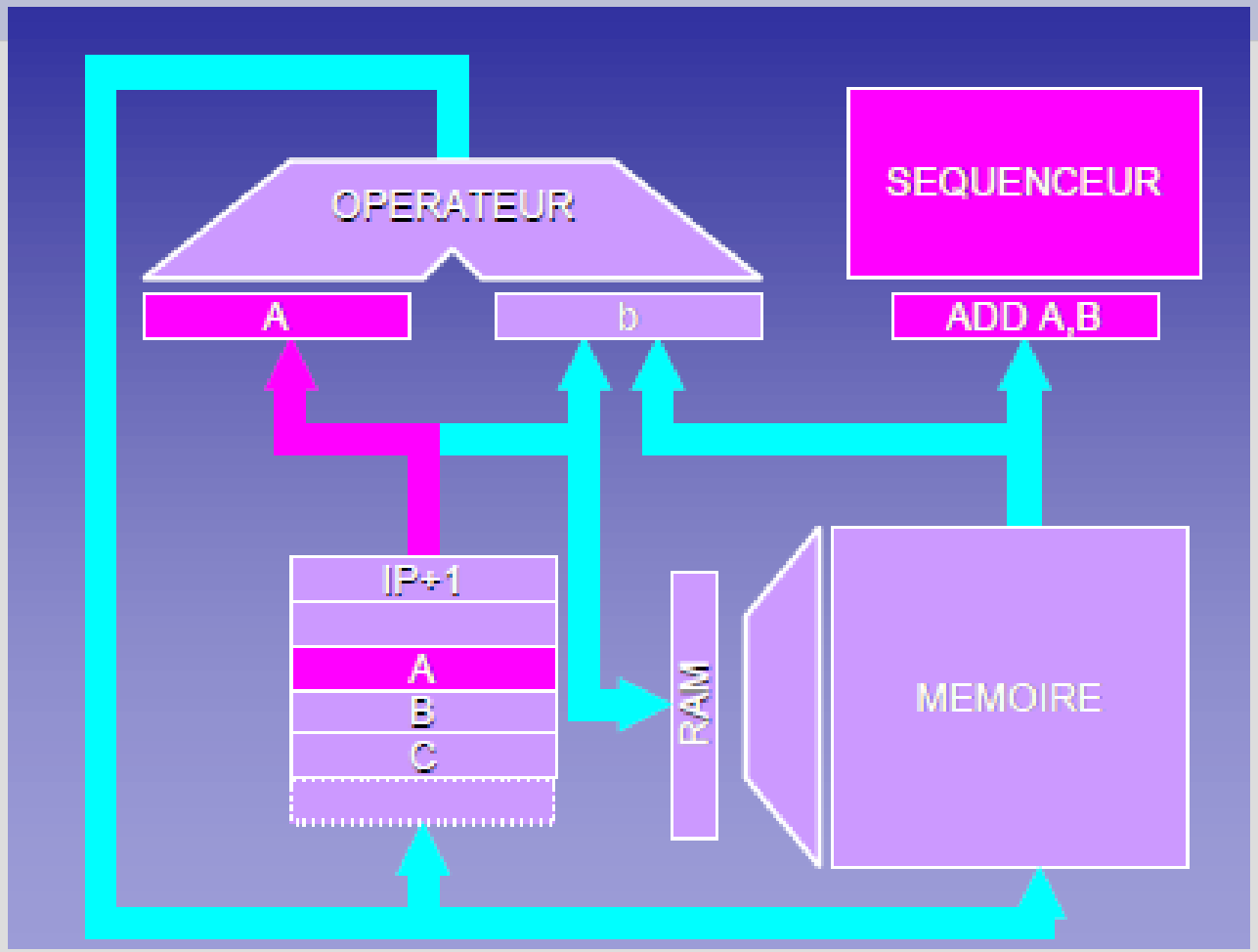
Lecture de l’instruction



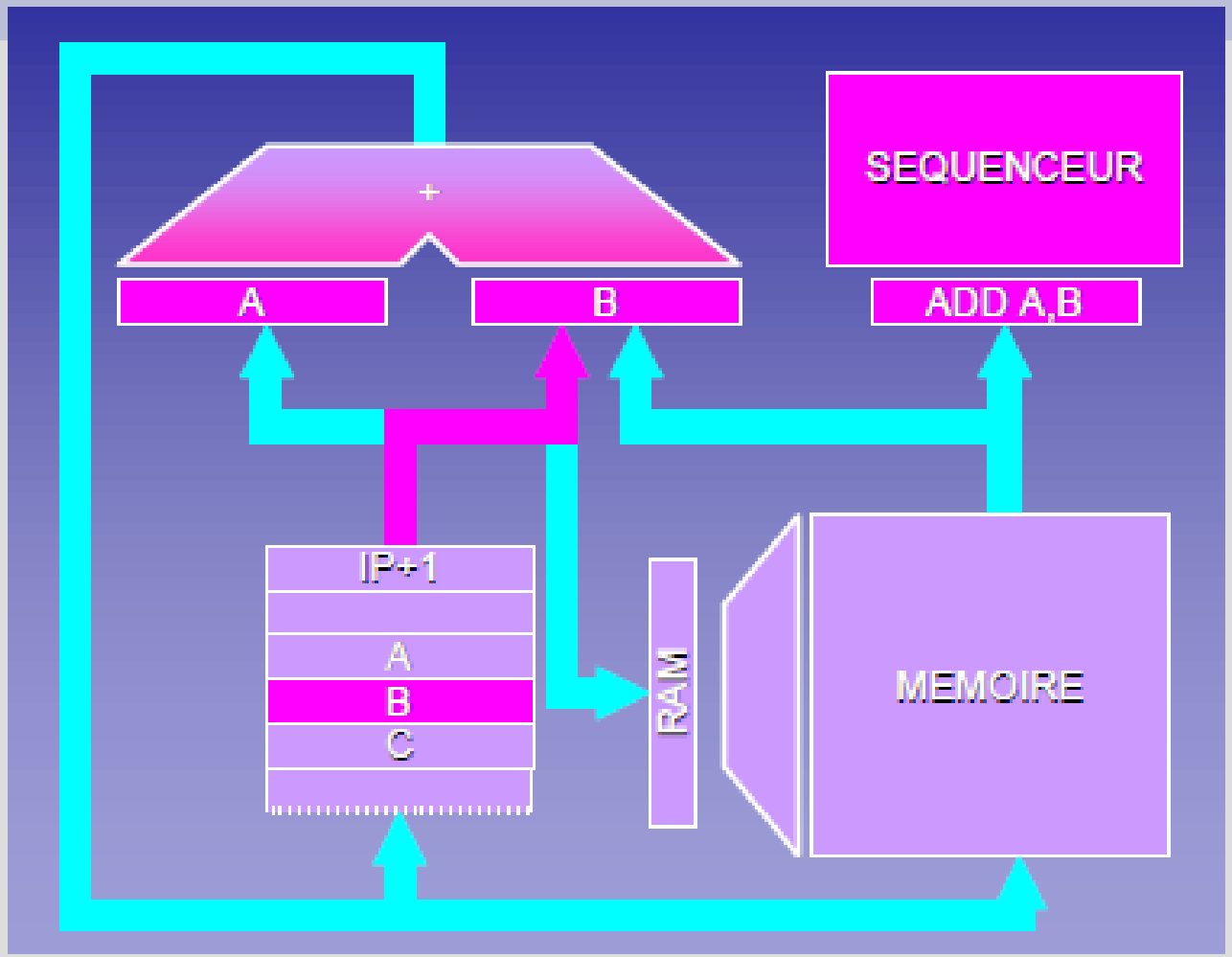
L’instruction arrive
on prépare la suite…



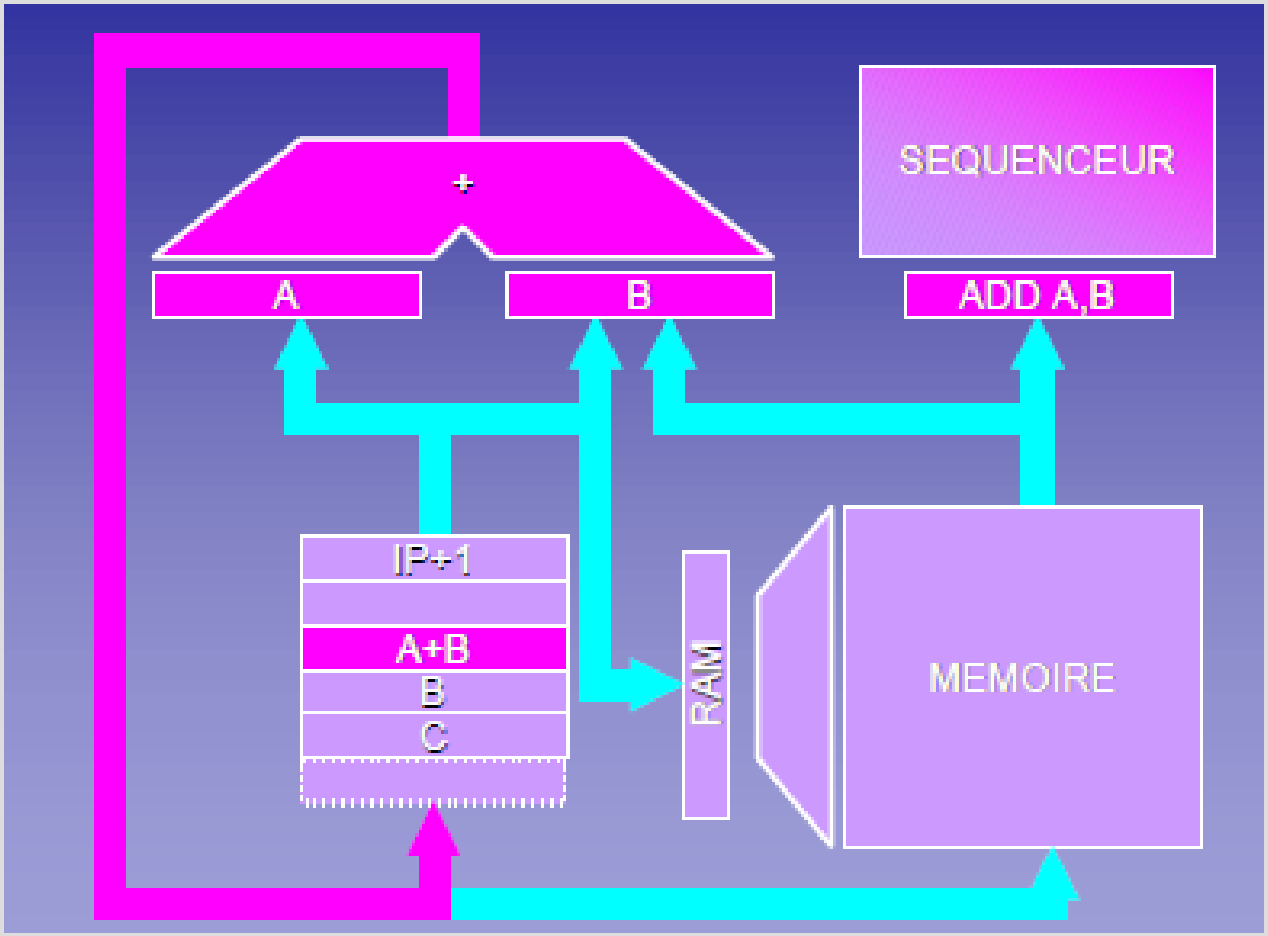
Les opérandes arrivent!



Prêt pour le calcul!!!



On range le résultat
et on recommence …



Et si on en exécute plusieurs?

- **Le processeur exécute (interprète) les instructions élémentaires à la suite**
- **Une séquence d'opérations peut décrire tous les problèmes**

notion de Programme

Déroulement du programme

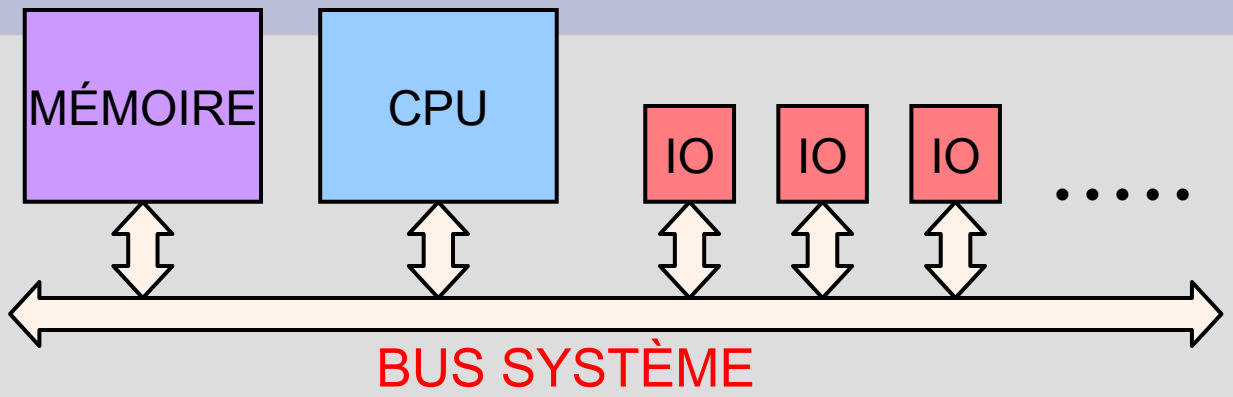
- **Le déroulement du programme est contrôlé par le Compteur Ordinal qui pointe vers la prochaine instruction à exécuter.**
  - **La séquentialité est intrinsèque au modèle VN**
- **Les instructions sont exécutées en séquence sauf en cas de saut. (Jump)**

Les données

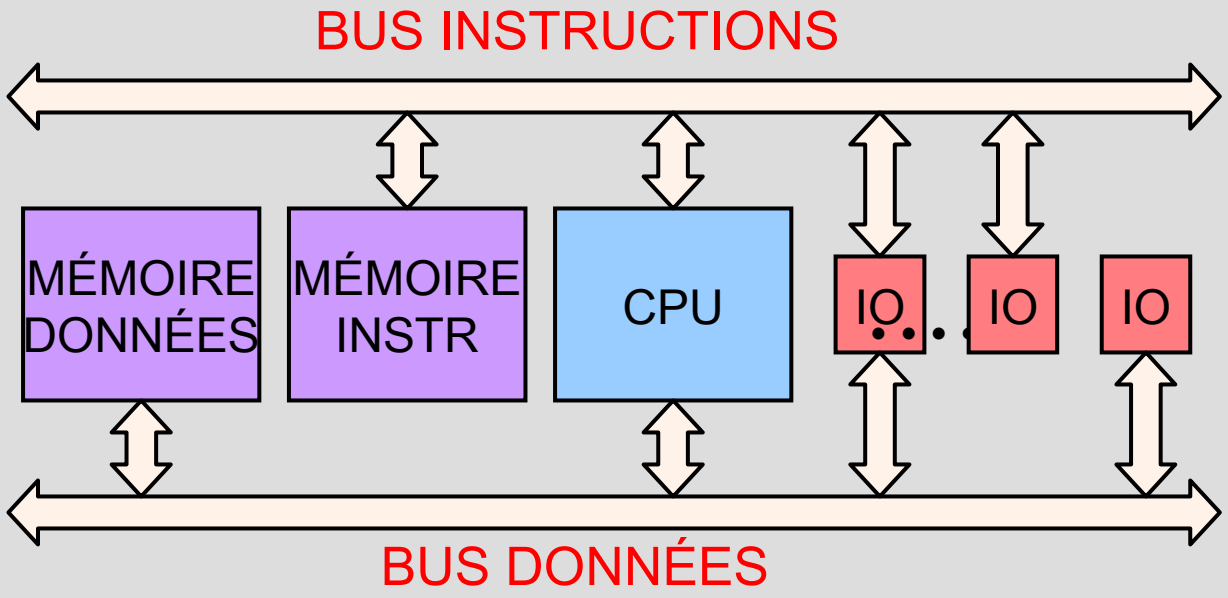
- **La mémoire conserve les données et les instructions, on parle de mémoire banalisée.**
- **Les instructions sont amenées une à une vers le processeur**
- **Les échanges entre mémoire / processeur se font via le système de communication : souvent le bus**

Von Neumann vs. Harvard

- Von Neumann



- Harvard



Séquencement des instructions

- **L'exécution d'une instruction passe par plusieurs étapes successives, chacune étant considérée comme une micro-opération.**

- **Certaines de ces actions correspondent à une activité mémoire, d'autres à une activité processeur.**

- **Pour effectuer une instruction, il faut toujours effectuer les actions suivantes:**

Séquencement des instructions

1. **Aller chercher l'instruction en mémoire;**
2. **Calculer l'adresse de la prochaine instruction, incrémenter le compteur ordinal**
3. **Décoder le code de l'opération**
4. **Calculer les adresses des opérandes si nécessaire**
5. **Extraire les opérandes éventuelles de la mémoire**
6. **Exécuter l'instruction**
7. **Calculer l'adresse du résultat**
8. **Ranger le résultat en mémoire**

Phases 1 2 3

- 1: Lit instruction suivante
  - Bus **Adresse** Õ PC
  - Bus **Commande** Õ « Lire instruction »
  - RI Õ Bus **Donnée**
- 2: Incrémente compteur ordinal
  - PC Õ PC + taille(instruction)
- 3 :Décode Instruction *ex: Add A,(123)*
  - *A Õ A + contenu @123.*

Phases 4 5

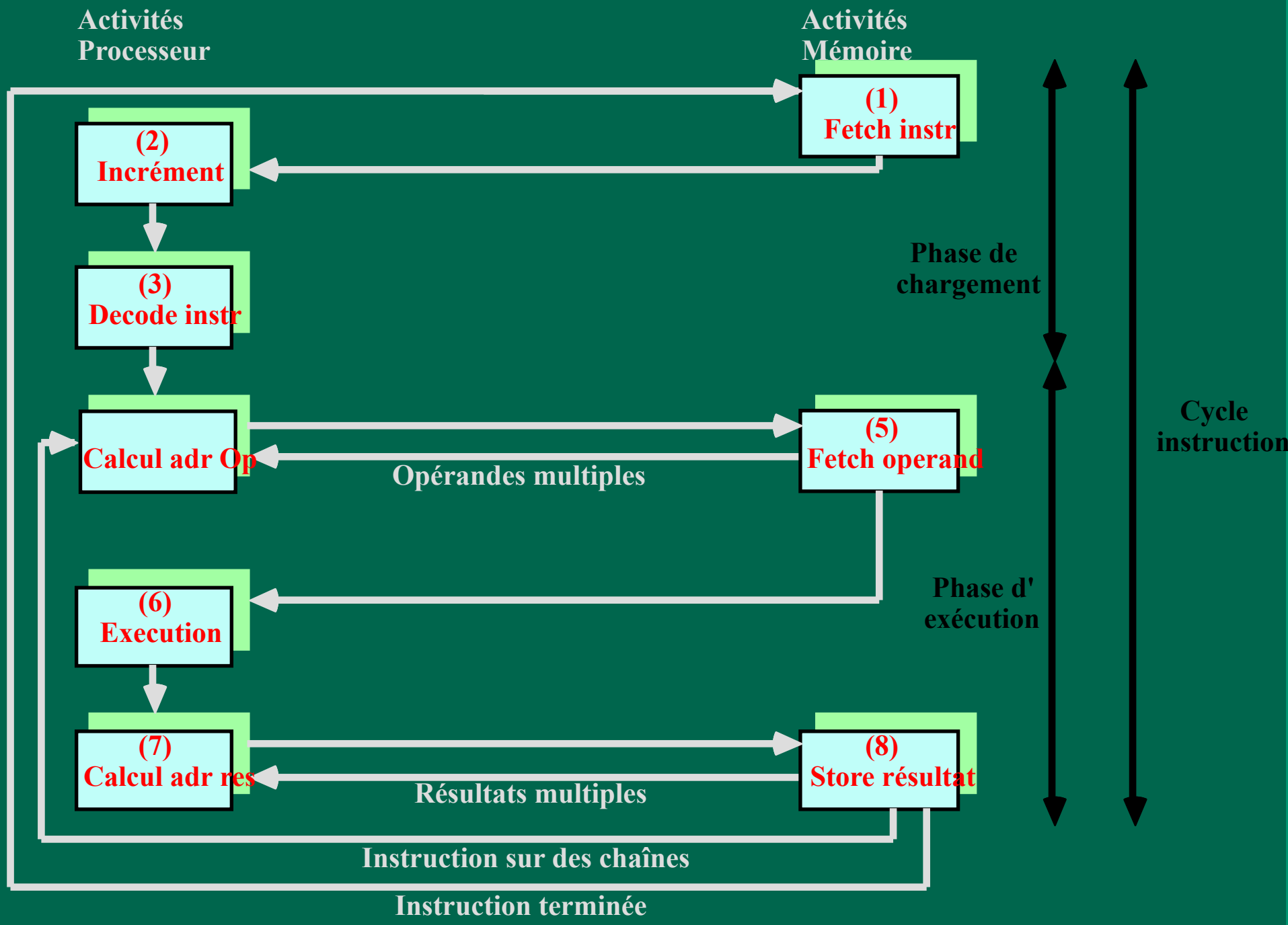
- 4 et 5 : Lit données (facultatif)
  - Bus **A** Õ 123
  - Bus **C** Õ « lire donnée »
  - tmp Õ Bus **D**
- Transfert données Õ UAL
  - UAL.E1 Õ A
  - UAL.E2 Õ tmp
  - UAL.Inst Õ « addition »

Phases 6 7 8

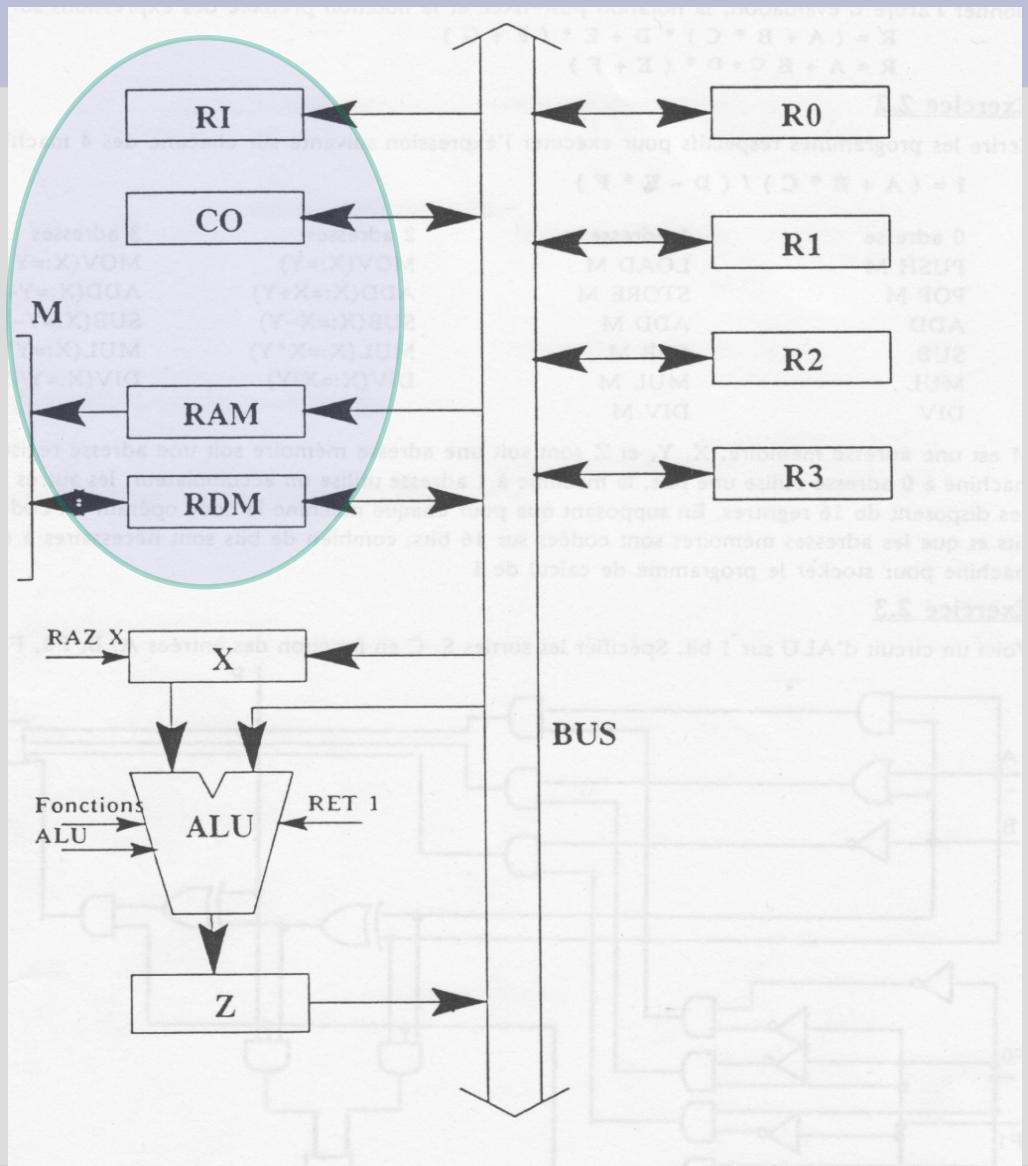
- 6: UAL calcule opération
  - Activation de l’additionneur intégral
  - Õ tmp’
- 7 et 8 : UC range résultat
  - A Õ tmp’

- Recommence
  - Lit & Exécute instruction suivante

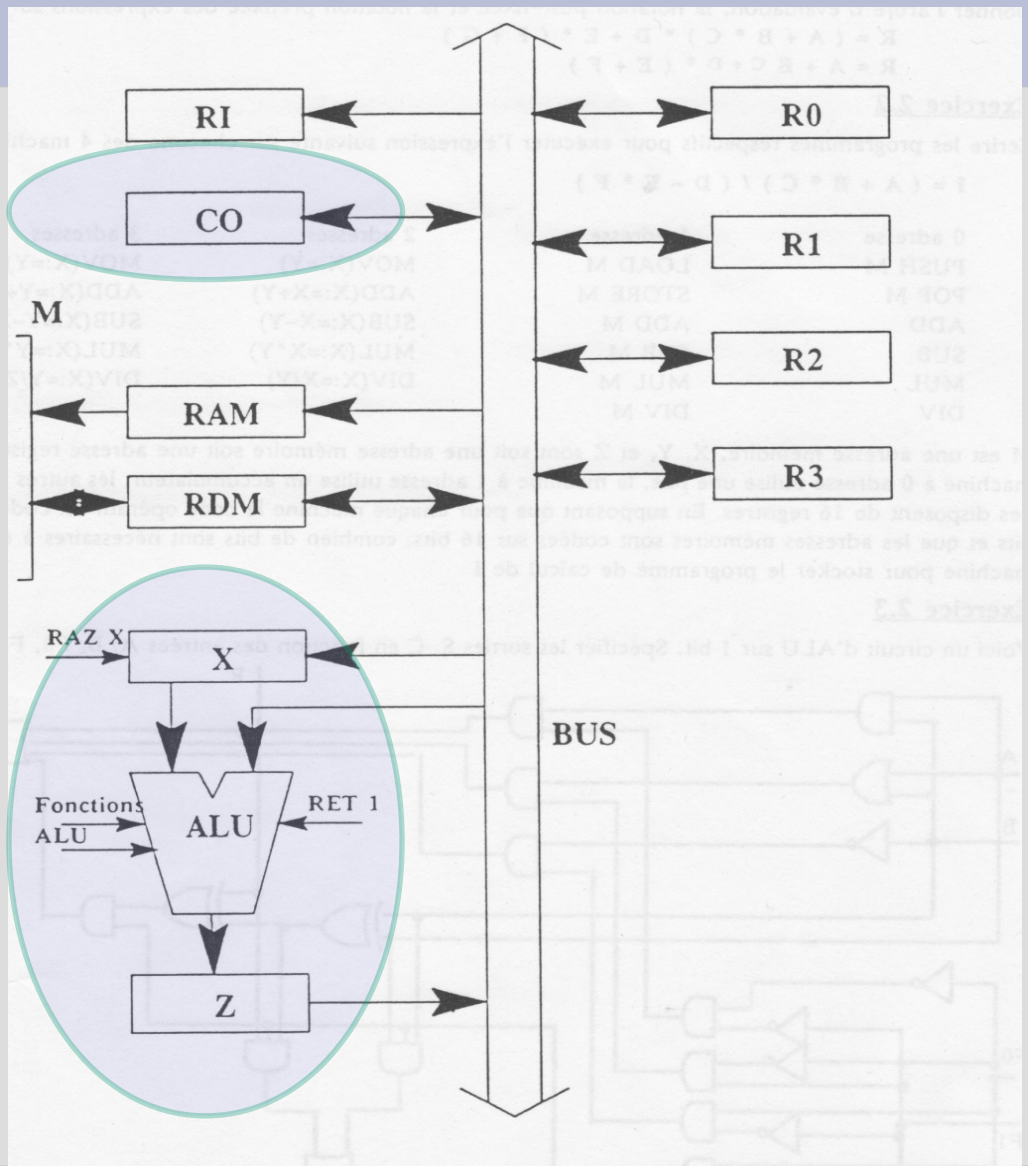
- Pas de repos pour un processeur…


Activités Processeur
Activités Mémoire
(1) Fetch instr
(2) Incrément
(3) Decode instr
Calcul adr Op
(5) Fetch operand
Opérandes multiples
(6) Execution
(7) Calcul adr res
(8) Store résultat
Résultats multiples
Instruction sur des chaînes
Instruction terminée
Phase de chargement
Phase d' exécution
Cycle instruction

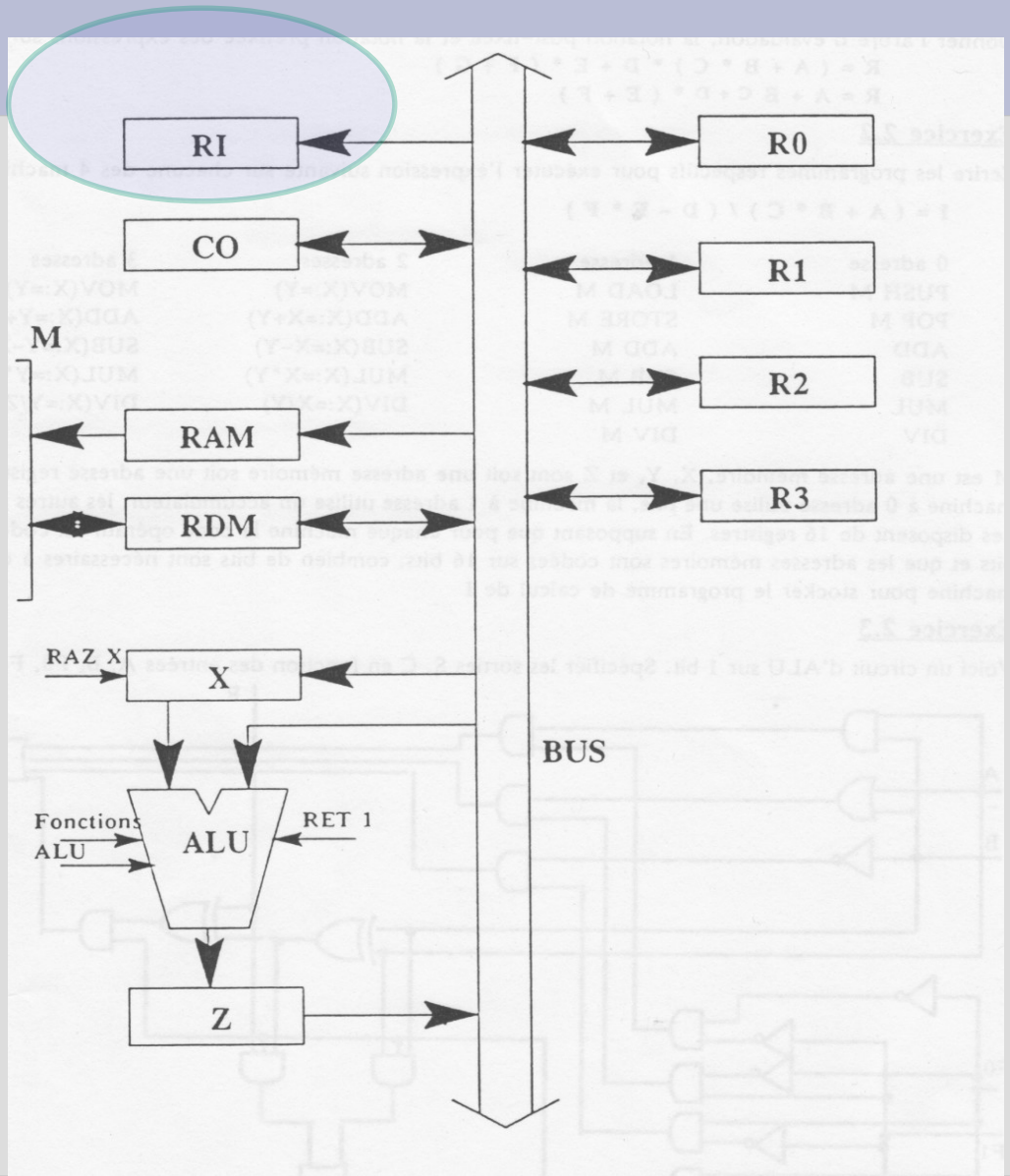
Aller chercher l'instruction en mémoire



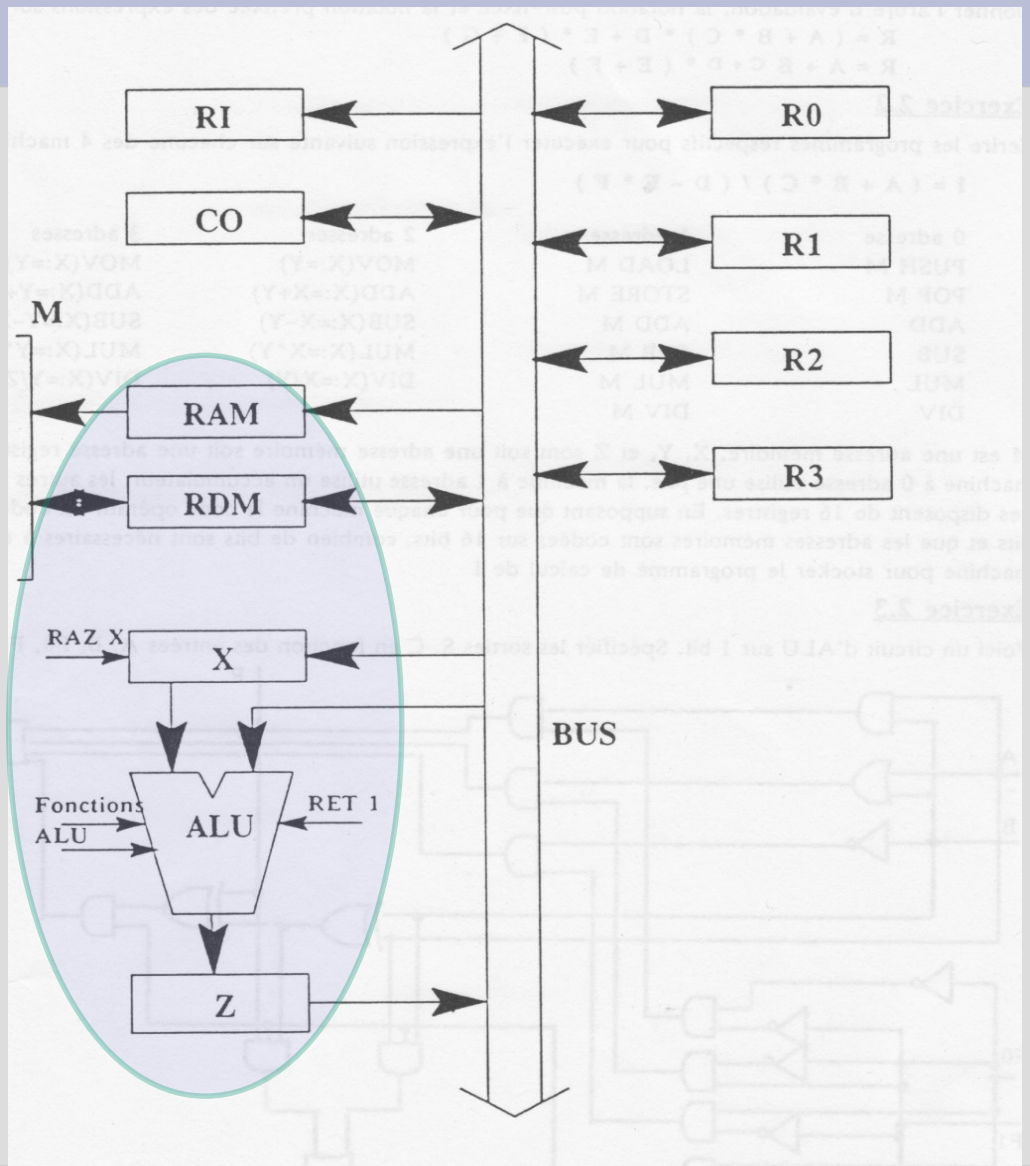
Calculer l'adresse de la prochaine instruction



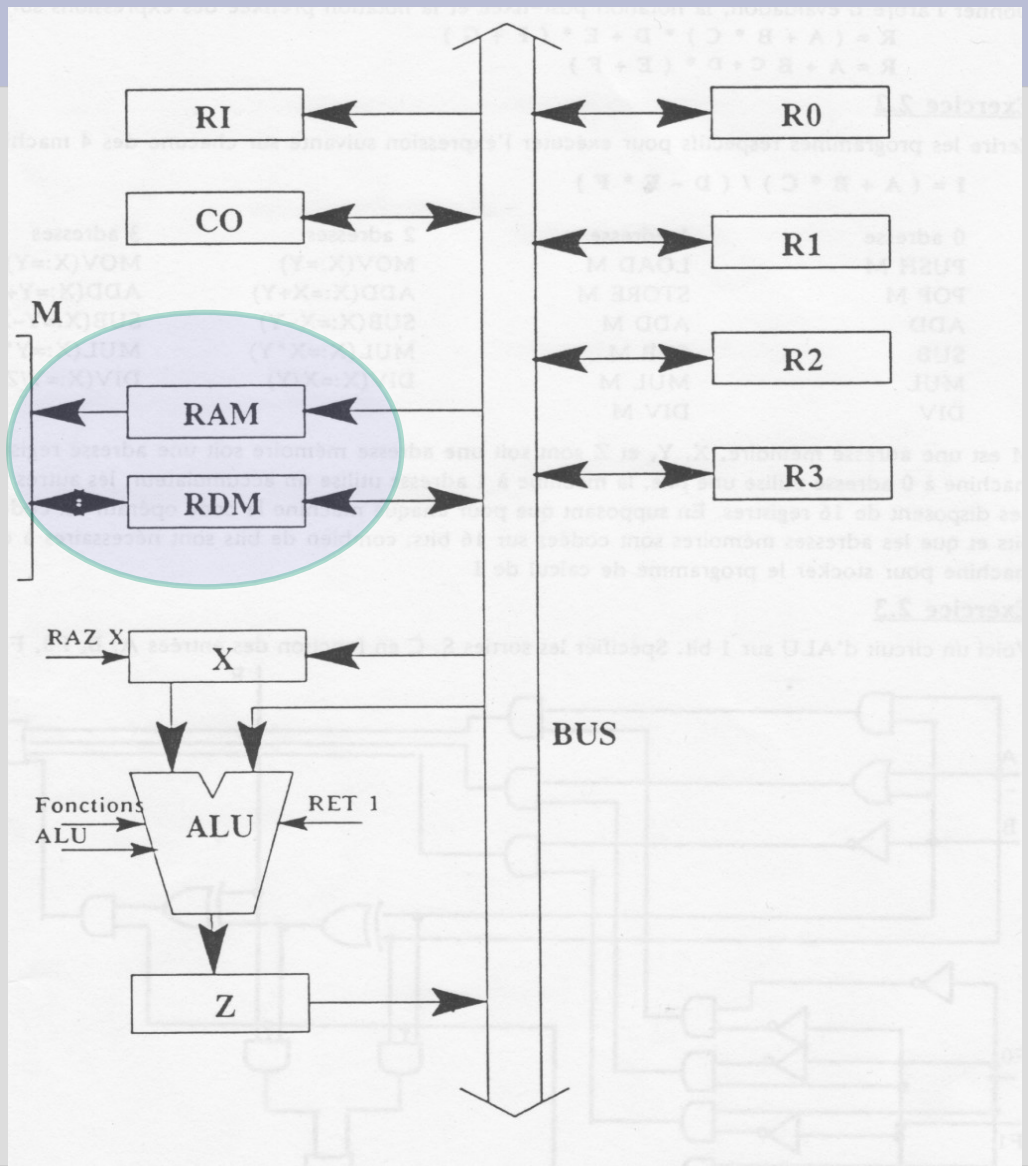
Décoder le code de l'opération



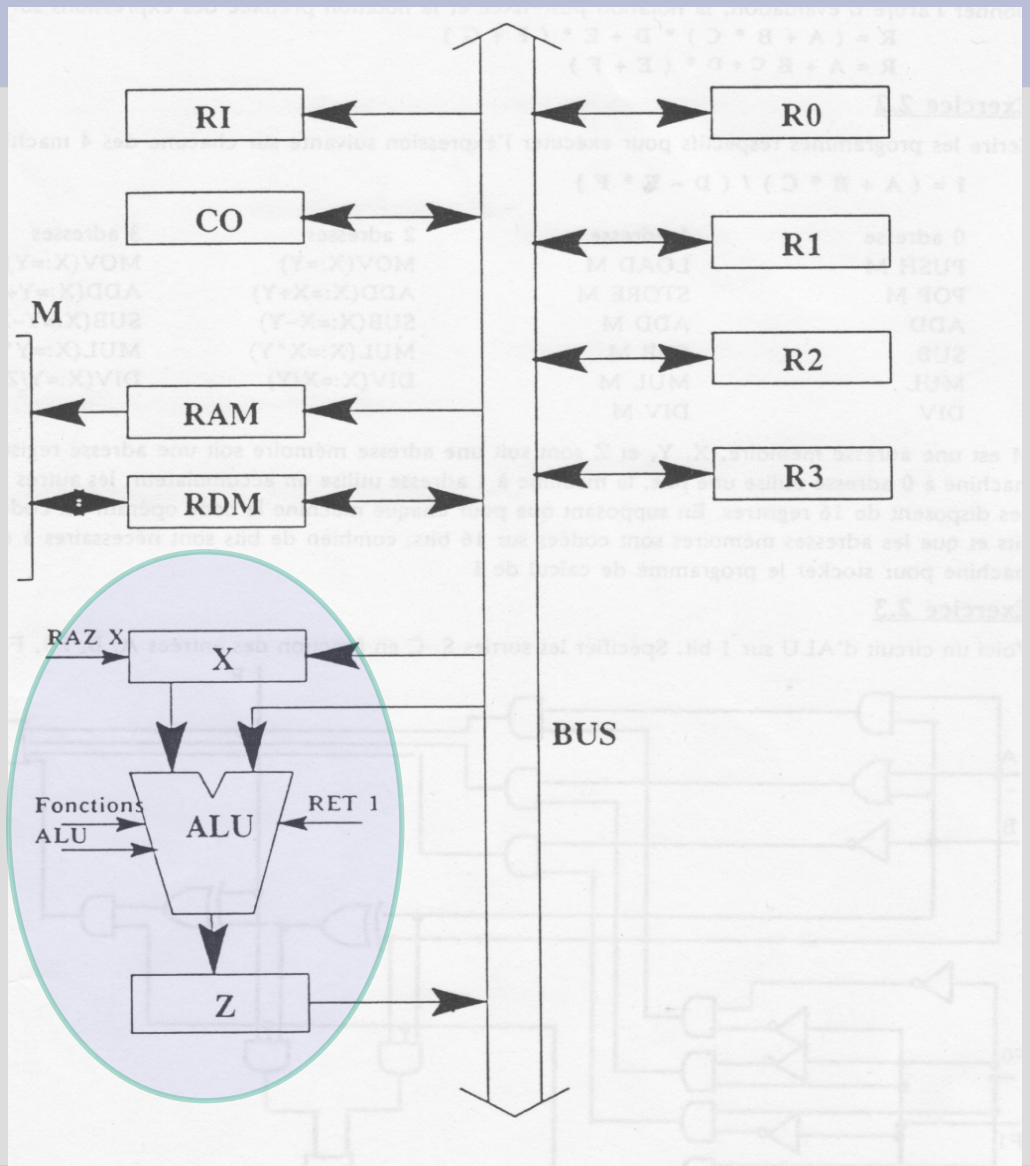
Calculer les adresses des opérandes si nécessaire



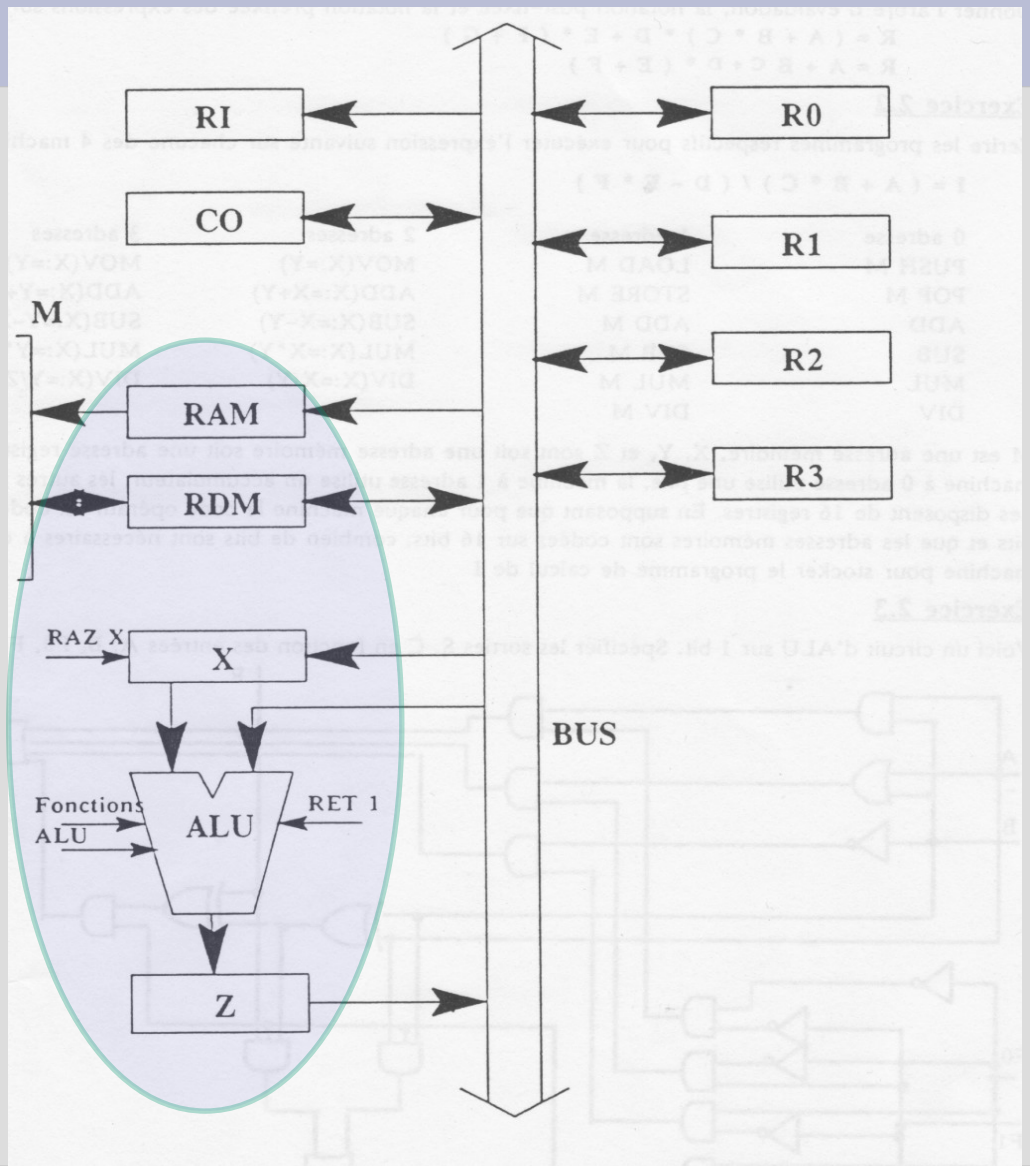
Extraire les opérandes éventuelles de la mémoire



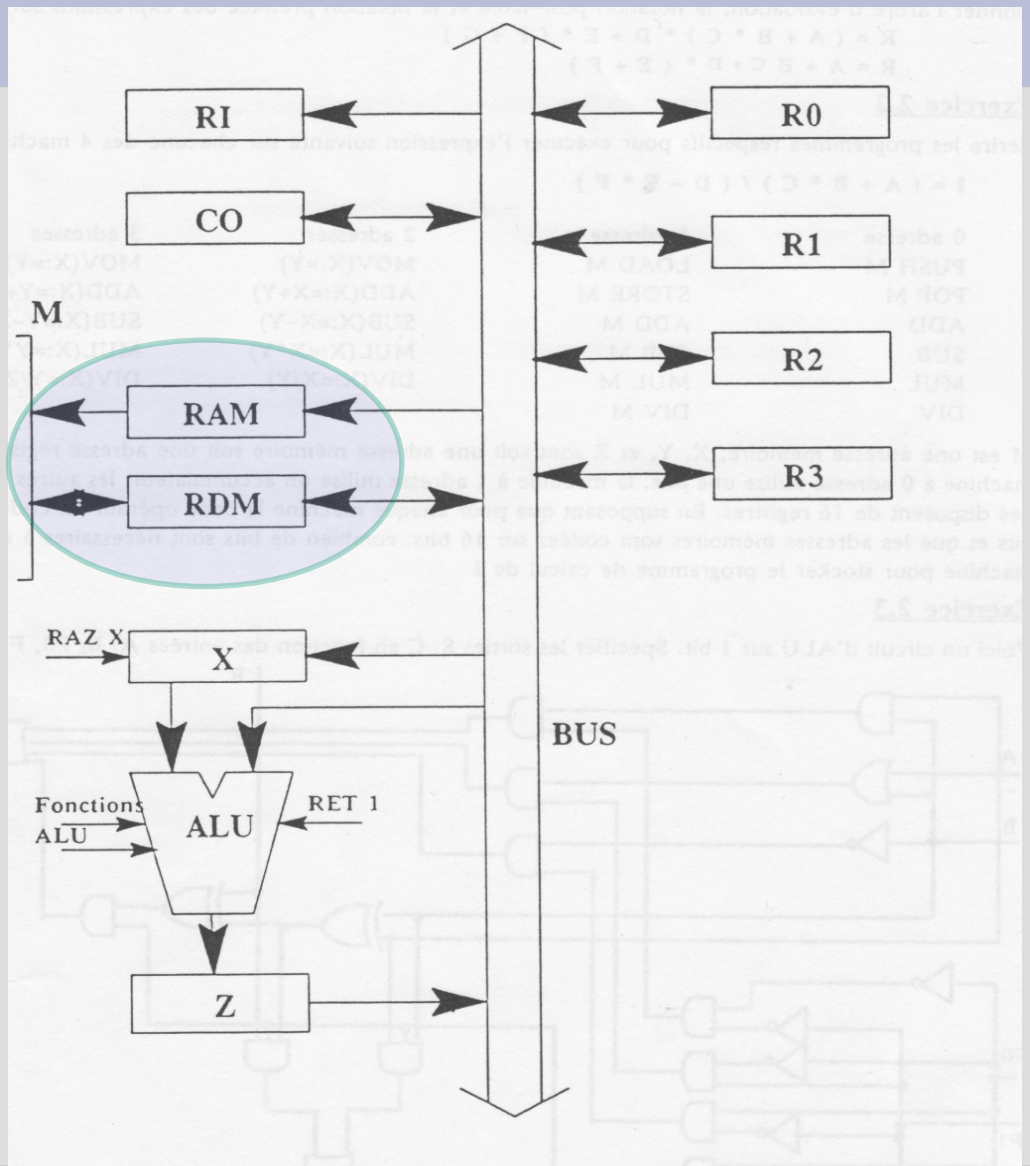
Exécuter l'instruction



Calculer l'adresse du résultat



Ranger le résultat en mémoire



Cycle du processeur

- L'exécution d'une instruction peut être découpée en plusieurs phases successives.
- Deux phases au moins sont définies:
  - la phase de chargement 'Fetch' et
  - phase d'exécution 'Execute'

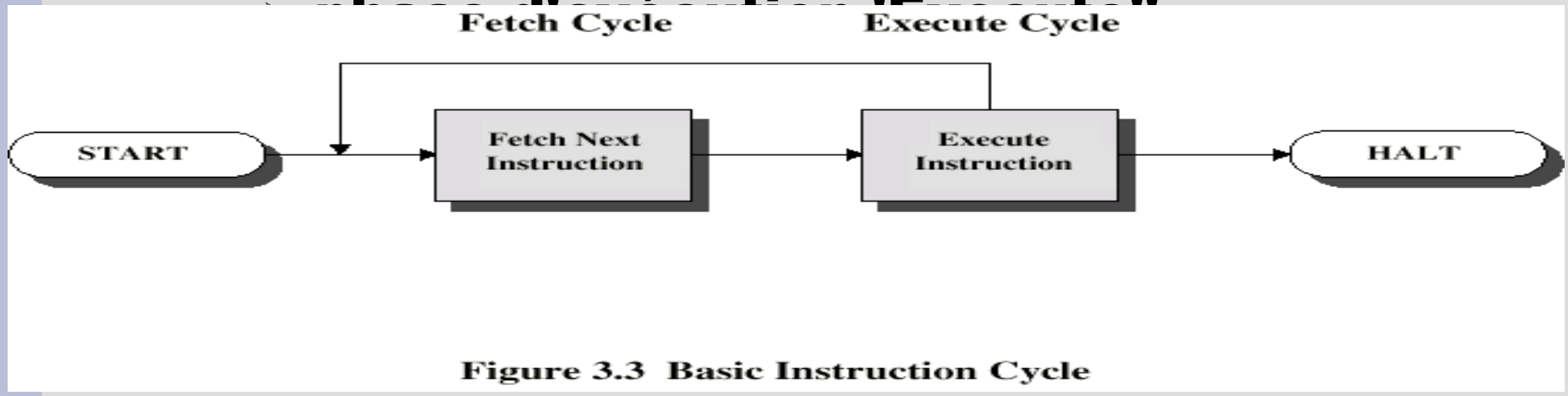


Figure 3.3 Basic Instruction Cycle

Différentes classes d'instruction

Instructions UAL	*Instructions Mémoire*	*Instructions Branchement*
Lecture instruction	Lecture instruction	Lecture instruction
Incrémentation CP	Incrémentation CP	Incrémentation CP
Décodage de l'instruction	Décodage de l'instruction	Décodage de l'instruction
Lecture des opérandes	Calcul de l'adresse mémoire	Calcul de l'adresse de branchement
Exécution	Accès mémoire	Exécution
Ecriture du résultat	Rangement du résultat	