Architecture des ordinateurs

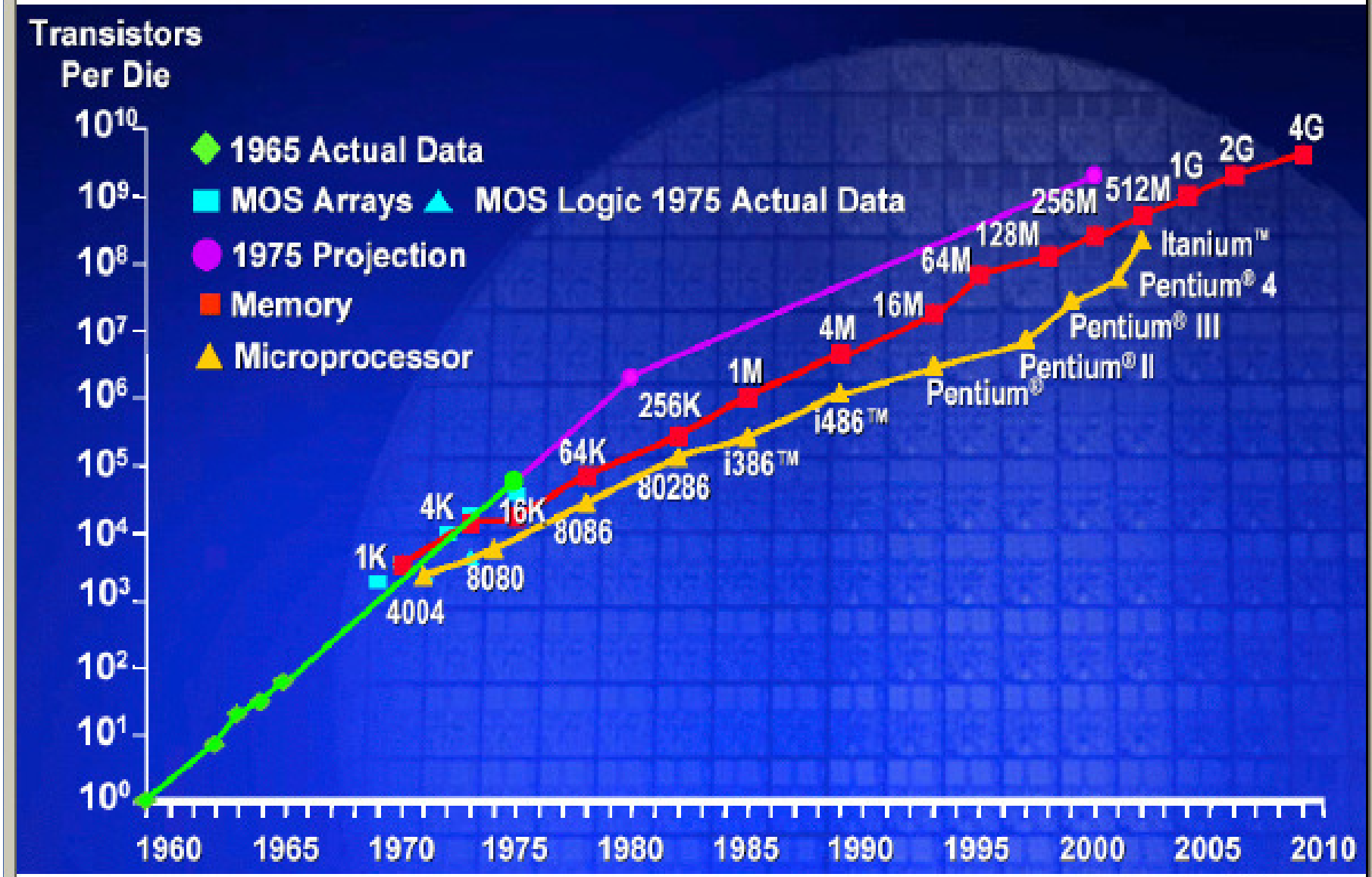
Le microprocesseur

Microprocesseur

- (noté **CPU**, pour *Central Processing Unit*) est un circuit électronique cadencé au rythme d'une horloge interne grâce à un cristal de quartz qui, soumis à un courant électrique, envoie des impulsions, appelées « **top** » ». La **fréquence d'horloge** (appelée également **cycle**, correspondant au nombre d'impulsions par seconde, s'exprime en Hertz (Hz).
- interprétation et d'exécution des instructions
- Organise les tâches précisées par le programme et d'assurer leur exécution.
- Il doit aussi prendre en compte les informations extérieures au système et assurer leur traitement.
- C'est le cerveau du système.
- Leur puissance continue de s'accroître et leur encombrement diminue régulièrement respectant toujours, pour le moment, la fameuse *loi de Moore* (1).

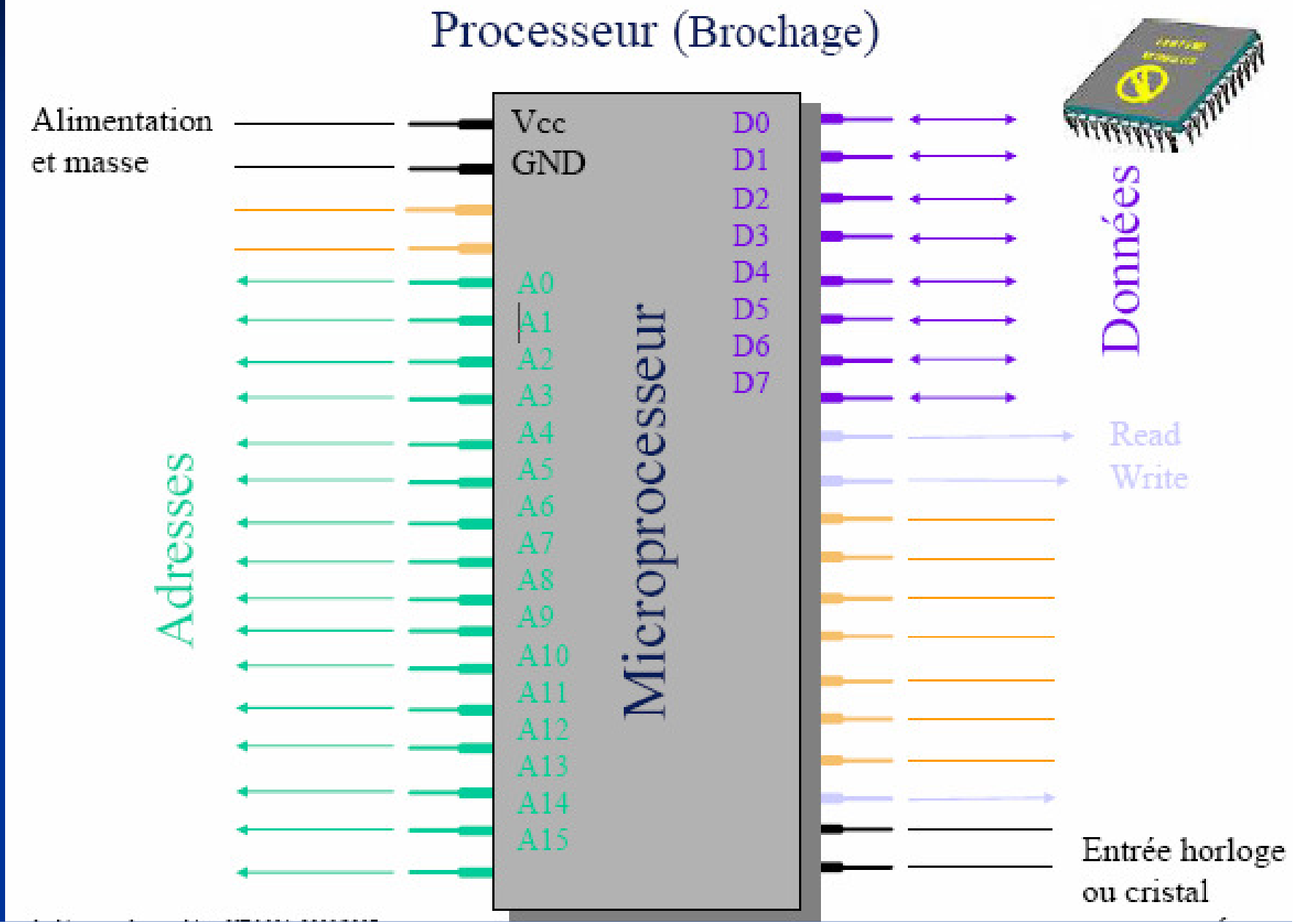
(1) Moore (un des co-fondateurs de la société Intel) a émis l'hypothèse que les capacités technologiques permettraient de multiplier par 2 tous les 18 mois le nombre de transistors intégrés sur les circuits.

Loi de Moore



Source : www.intel.com

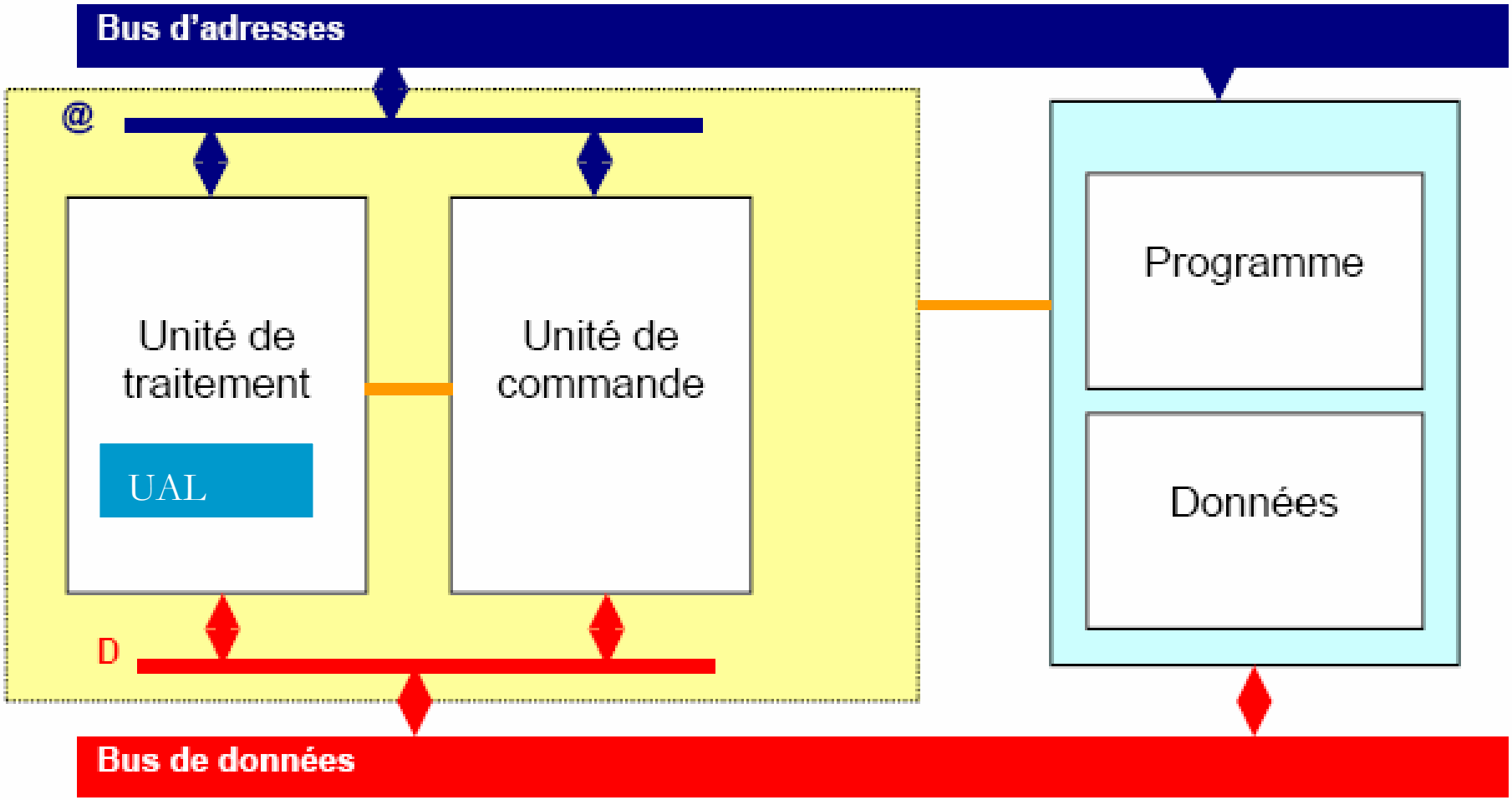
Microprocesseur



Microprocesseur

- Un microprocesseur est construit autour de deux éléments principaux :
  - Une unité de commande(contrôle)
  - Une unité de traitement(UAL)

Microprocesseur



Unité de commande

- Le rôle de l'unité de contrôle (ou unité de commande) est de :
  - coordonner le travail de toutes les autres unités (UAL , mémoire,….)
  - et d'assurer la synchronisation de l'ensemble.
  - séquencer le déroulement des instructions

- Elle assure :
  - la recherche (lecture) de l'instruction et des données à partir de la mémoire,
  - le décodage de l'instruction et l'exécution de l'instruction en cours
  - et prépare l'instruction suivante.
- Pour cela, elle est composée par :

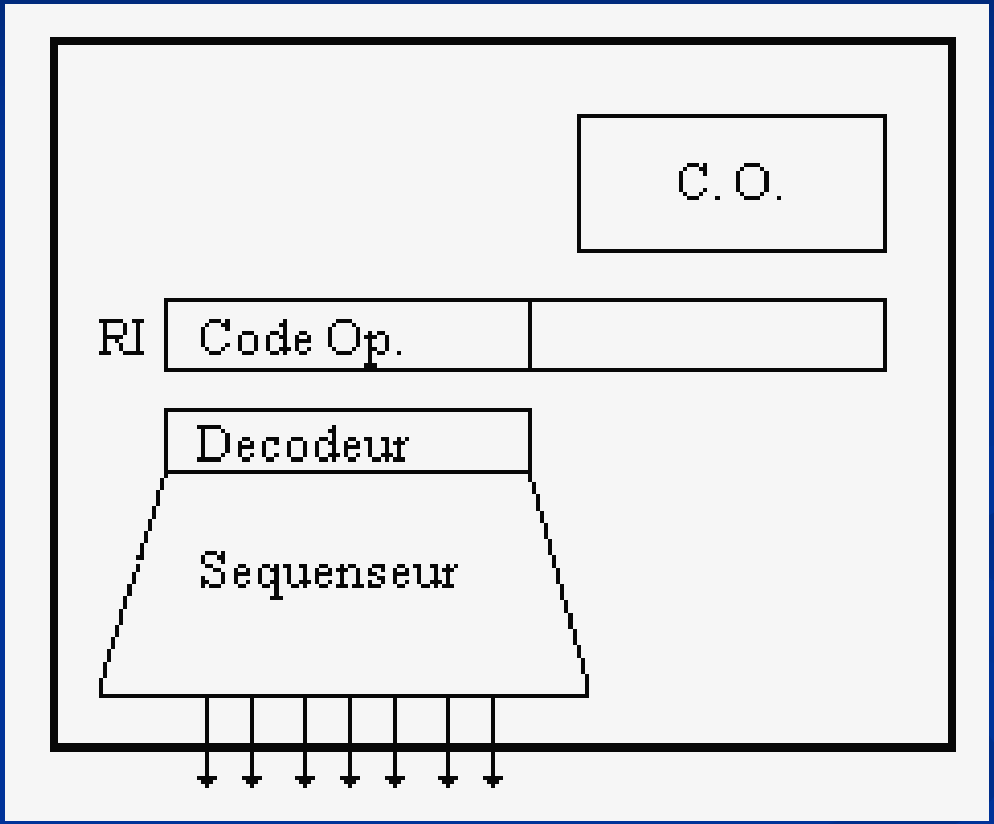
Unité de commande

- Un registre instruction (RI) : contient l'instruction en cours d'exécution. Chaque instruction est décoder selon sont code opération grâce à un décodeur.
- Un registre qui s'appel compteur ordinal (CO) ou le compteur de programme (CP) : contient l'adresse de la prochaine instruction à exécuter (pointe vers la prochaine instruction à exécuter). Initialement il contient l'adresse de le première instruction du programme à exécuter.
- Un séquenceur : il organise (synchronise) l'exécution des instruction selon le rythme de l'**horloge**, il génère les signaux nécessaires pour exécuter une instruction.

Unité de commande : horloge

- Horloge: Définit le cycle de base : cycle machine (clock cycle)
- Utilisée pour synchroniser chaque étape des cycles de recherche et d'exécution
- Temps exécution d'un cycle de recherche ou d'exécution:Prend un certain nombre de cycles de base
- Cycle CPU = temps d'exécution minimal d'une instruction (recherche + exécution)

Unité de commande



Remarques

- Le microprocesseur peut contenir d'autres registres autre que CO,RI et ACC.
- Ces registres sont considérés comme une mémoire interne (registre de travail) du microprocesseur.
- Ces registres sont plus rapide que la mémoire centrale , mais le nombre de ces registre est limité.
- Généralement ces registres sont utilisés pour sauvegarder les données avant d'exécuter une opération.
- Généralement la taille d'un registre de travail est

UAL

- L'unité arithmétique et logique réalise une opération élémentaire (addition, ,soustraction, multiplication, . . .).
- L'UAL regroupe les circuits qui assurent les fonctions logiques et arithmétiques de bases (ET,OU,ADD,SUS,…..).
- L'UAL comporte un registre accumulateur (ACC) : c'est un registre de travail qui sert a stocker un opérande (données)au début d'une opération et le résultat à la fin.

UAL

- L'UAL comporte aussi un registre d'état : Ce registre nous indique l'état du déroulement de l'opération .
- Ce registre est composé d'un ensemble de bits. Ces bits s'appels indicateurs (drapeaux ou flags).
- Ces indicateurs sont mis à jours (modifiés)après la fin de l'exécution d'une opération dans l'UAL.
- Les principeaux indicateurs sont :
  - Retenue : ce bit est mis à 1 si l'opération génère une retenue.
  - Signe :ce bit est mis à 1 si l'opération génère un résultat négative.
  - Débordement :ce bit est mis à 1 s'il y a un débordement.
  - Zero : ce bit est mis à 1 si le résultat de l'opération est nul.

Schéma d'une UAL

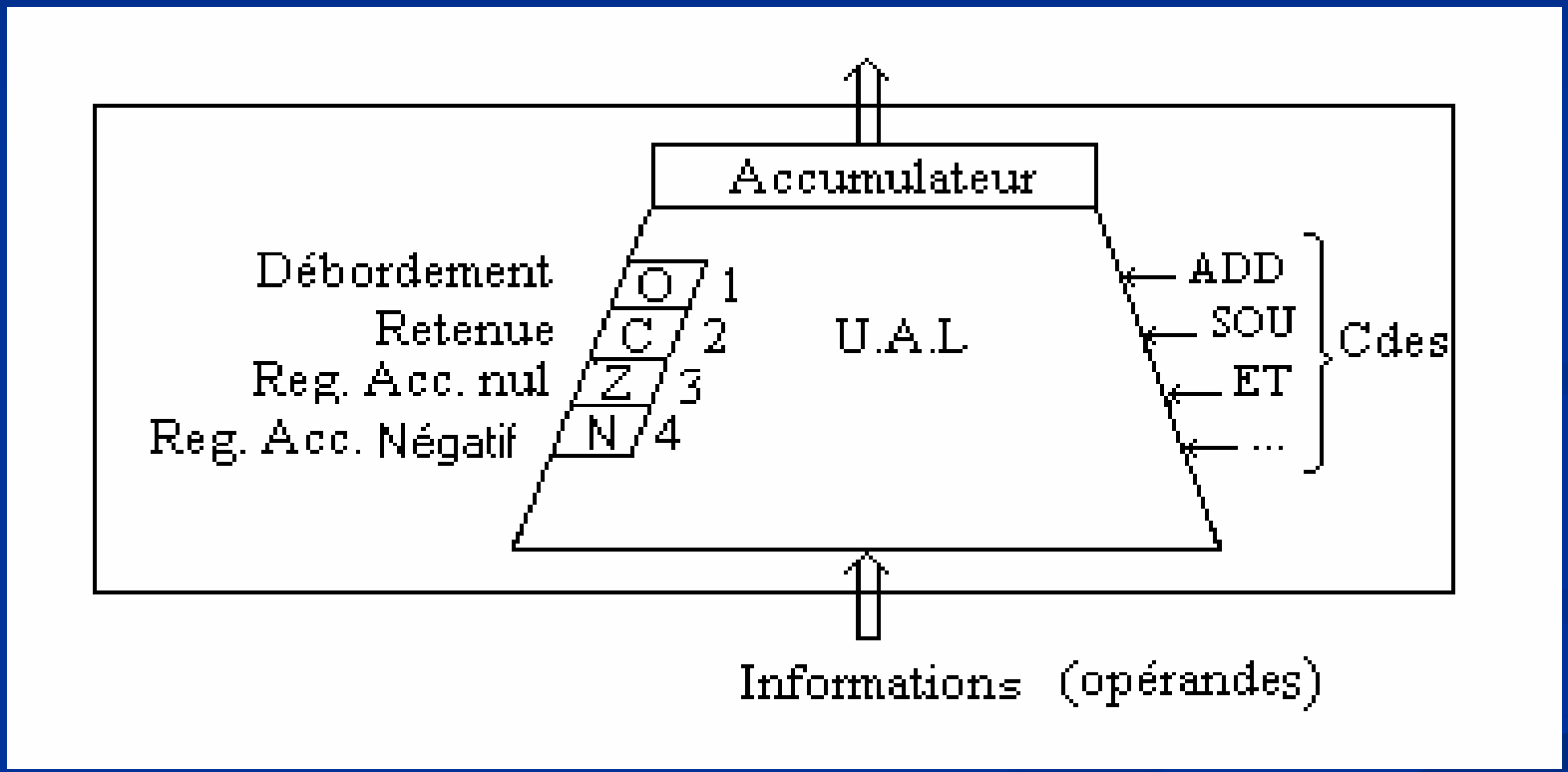
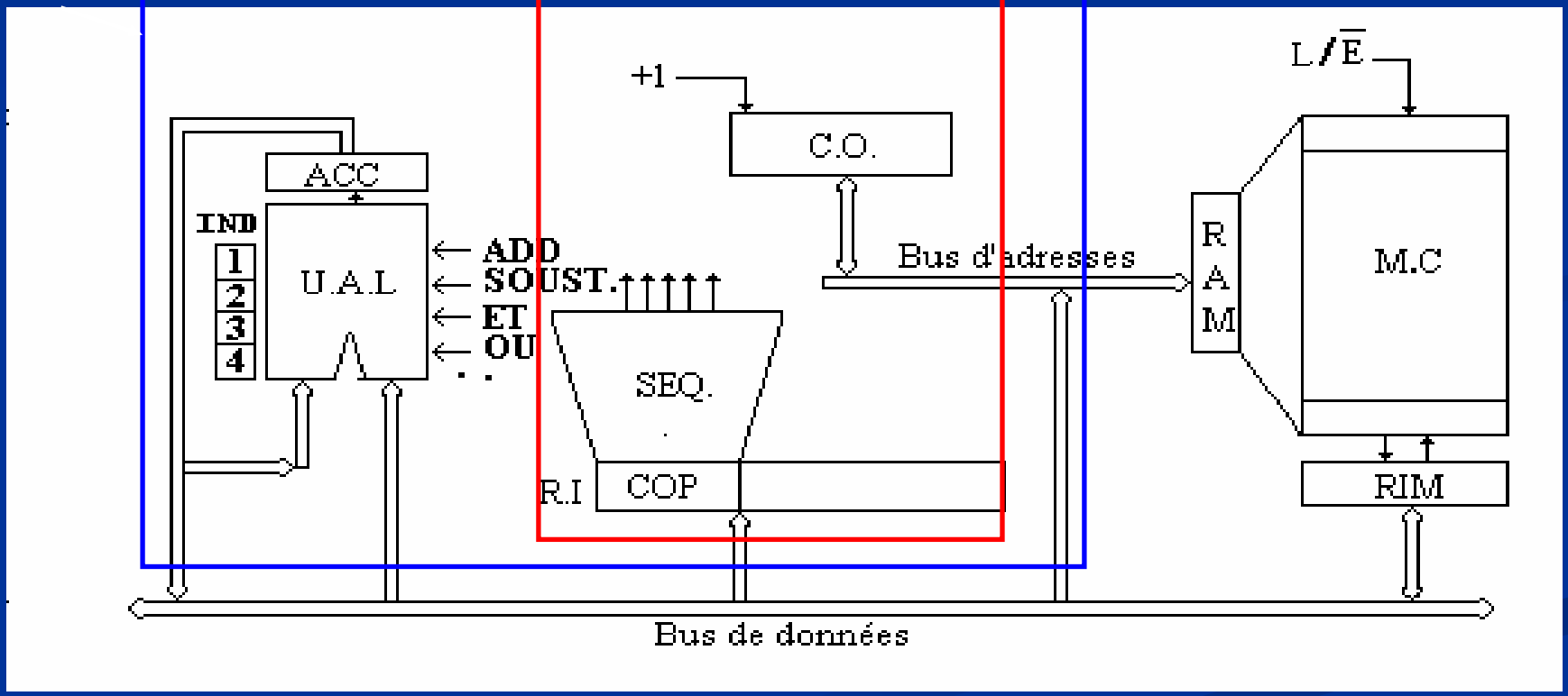
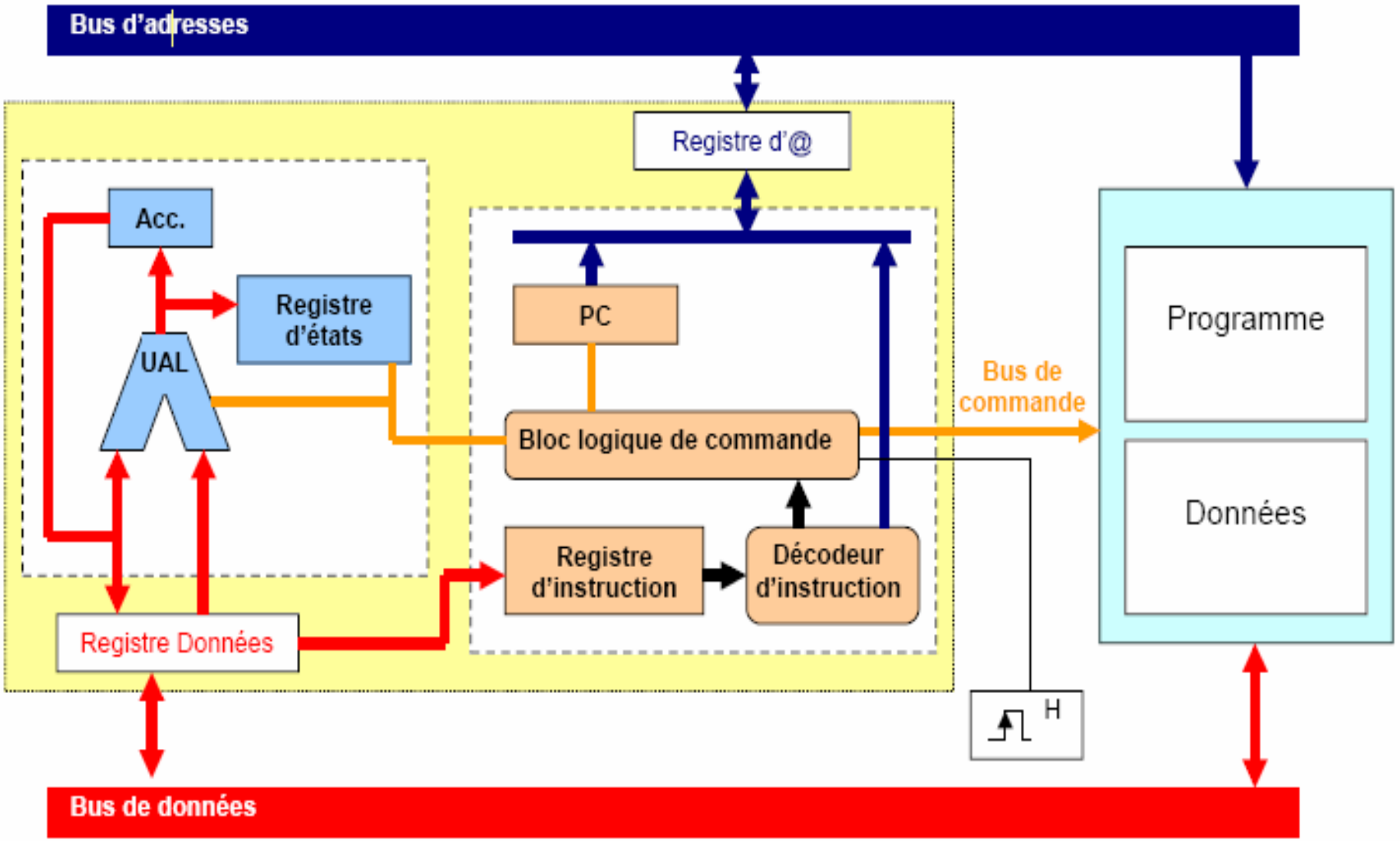


Schéma détaillé d'une machine

UC

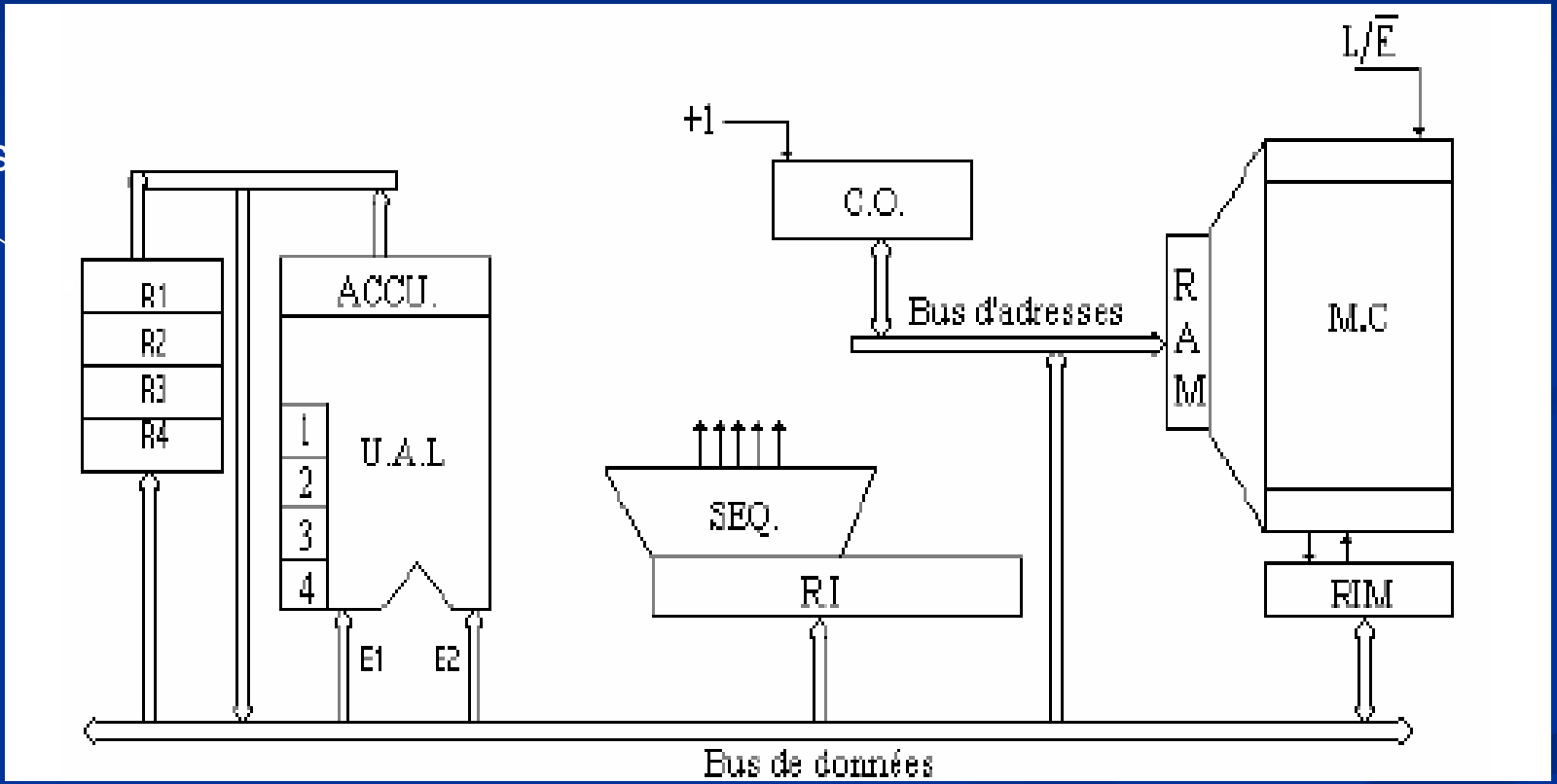


Microprocesseur



Une machine avec des registres de travail

registres



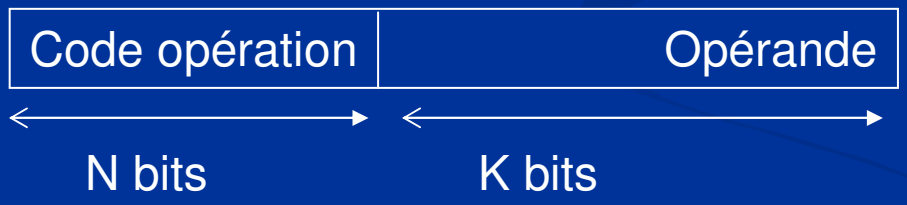
Jeu d'instructions

- La première étape de la conception d'un microprocesseur est la définition de son jeu d'instructions. Le jeu d'instructions décrit l'ensemble des opérations élémentaires que le microprocesseur pourra exécuter.

- Chaque microprocesseur possède un certain nombre limité d'instructions qu'il peut exécuter. Ces instructions s'appelles jeu d'instructions.
- Le jeu d'instructions décrit l'ensemble des opérations élémentaires que le microprocesseur peut exécuter.
- Les instructions peuvent être classifiées en 4 catégories :
  - Instruction d'affectation : elle permet de faire le transfert des données entre les registres et la mémoire
    - Écriture : registre → mémoire
    - Lecture : mémoire → registre
  - Les instructions arithmétiques et logiques (ET , OU , ADD,….)
  - Instructions de branchement (conditionnelle et inconditionnelle)
  - Instructions d'entrées sorties.

Codage d'une instruction

- Les instructions et leurs opérandes (données) sont stocké dans la mémoire.
- La taille d'une instruction (nombre de bits nécessaires pour la représenter en mémoire) dépend du type de l'instruction et du type de l'opérande.
- L'instruction est découpée en deux parties :
  - Code opération (code instruction) : un code sur N bits qui indique quelle instruction.
  - La champs opérande : qui contient la donnée ou la référence (adresse) à la donnée.



•Le format d'une instruction peut ne pas être le même pour toutes les instructions.
•Le champs opérande peut être découpé à sont tours en plusieurs champs

Mode d'adressage

- Un mode d'adressage définit la manière dont le microprocesseur va accéder à l'opérande.
- différents modes d'adressage dépendent des microprocesseurs mais on retrouve en général :
- l'adressage de registre où l'on traite la données contenue dans un registre
- l'adressage immédiat où l'on définit immédiatement la valeur de la donnée
- l'adressage direct où l'on traite une données en mémoire
- Selon le mode d'adressage de la donnée, une instruction sera codée par 1 ou plusieurs octets.

Mode d'adressage

- La champs opérande contient la donnée ou la référence (adresse) à la donnée.
- Le mode d'adressage définit la manière dont le microprocesseur va accéder à l'opérande.
- Le code opération de l'instruction comportent un ensemble de bits pour indiquer le mode d'adressage.

- Les modes d'adressage les plus utilités sont :
  - Immédiat
  - Direct
  - Indirect
  - Indexé
  - relatif

Machine à 3 adresses

- Dans ce type de machine pour chaque instruction il faut préciser :
  - l'adresse du premier opérande
  - du deuxième opérande
  - et l'emplacement du résultat

Code opération	Opérande1	Opérande2	Résultat

Exemple :

ADD A,B,C (C←B+C)

•Dans ce type de machine la taille de l'instruction est grand .
• Pratiquement il n'existent pas de machine de ce type.

Machine à 2 adresses

- Dans de type de machine pour chaque instruction il faut préciser :
  - l'adresse du premier opérande
  - du deuxième opérande ,
- l'adresse de résultat est implicitement l'adresse du deuxième opérande .

Code opération	Opérande1	Opérande2

Exemple :

ADD A,B (B←A +B)

Machine à 1 adresses

- Dans de type de machine pour chaque instruction il faut préciser uniquement l'adresse du deuxième opérande.
- Le premier opérande existe dans le registre accumulateur.
- Le résultat est mis dans le registre accumulateur.

Code opération	Opérande2

Exemple :

ADD A (ACC←(ACC) + A)

Ce type de machine est le plus utilisé.

Adressage immédiat

- L'opérande existent dans le champs adresse de l'instruction

Code opération	Opérande

Exemple :
ADD 150

ADD	150

Cette commande va avoir l'effet suivant : ACC←(ACC)+ 150

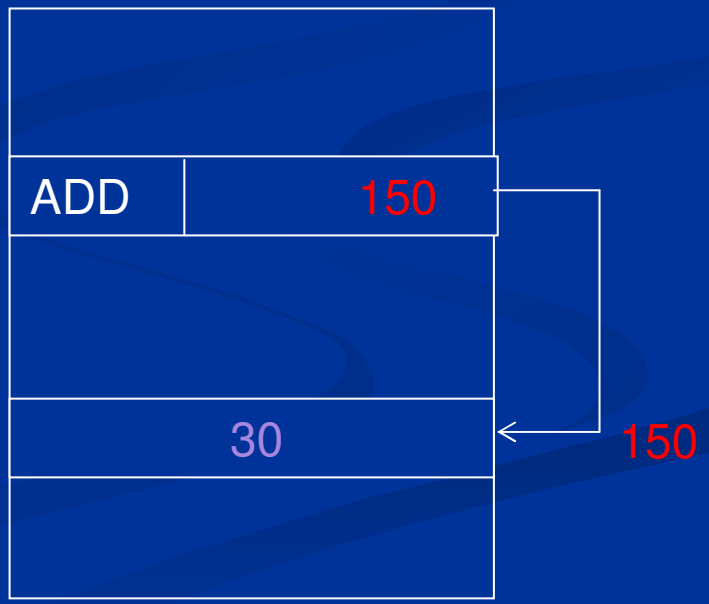
Si le registre accumulateur contient la valeur 200 alors
après l'exécution son contenu sera égale à 350

Adressage direct

- Le champs opérande de l'instruction contient l'adresse de l'opérande (emplacement en mémoire)
- Pour réaliser l'opération il faut le récupérer (lire) l'opérande à partir de la mémoire. ACC ← (ACC)+ (ADR)

Exemple :
On suppose que l'accumulateur continent la valeur 20 .

A la fin de l'exécution nous allons avoir la valeur 50 (20 + 30)



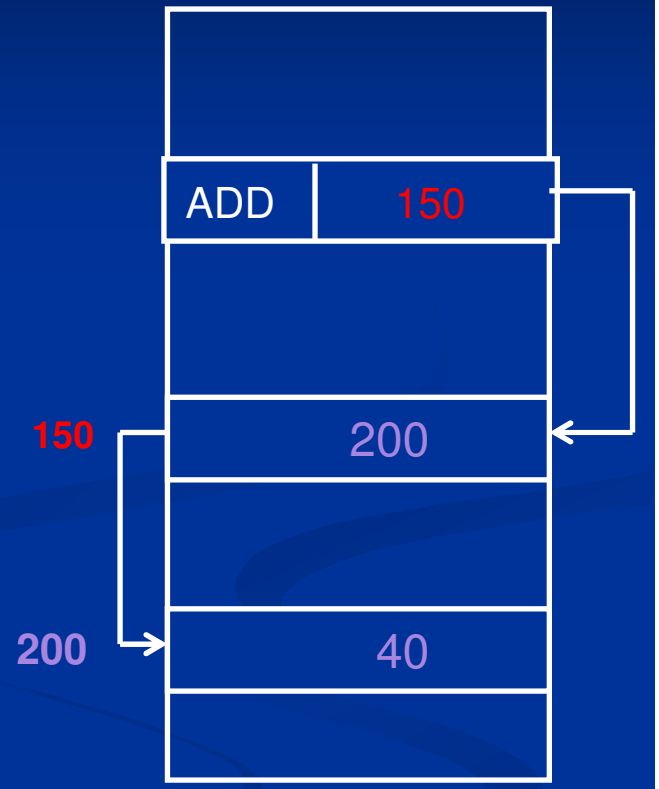
Adressage indirect

- La champs adresse contient l'adresse de l'adresse de l'opérande.
- Pour réaliser l'opération il faut :
  - Récupérer l'adresse de l'opérande à partir de la mémoire.
  - Par la suite il faut chercher l'opérande à partir de la mémoire.

ACC← (ACC)+ ((ADR))

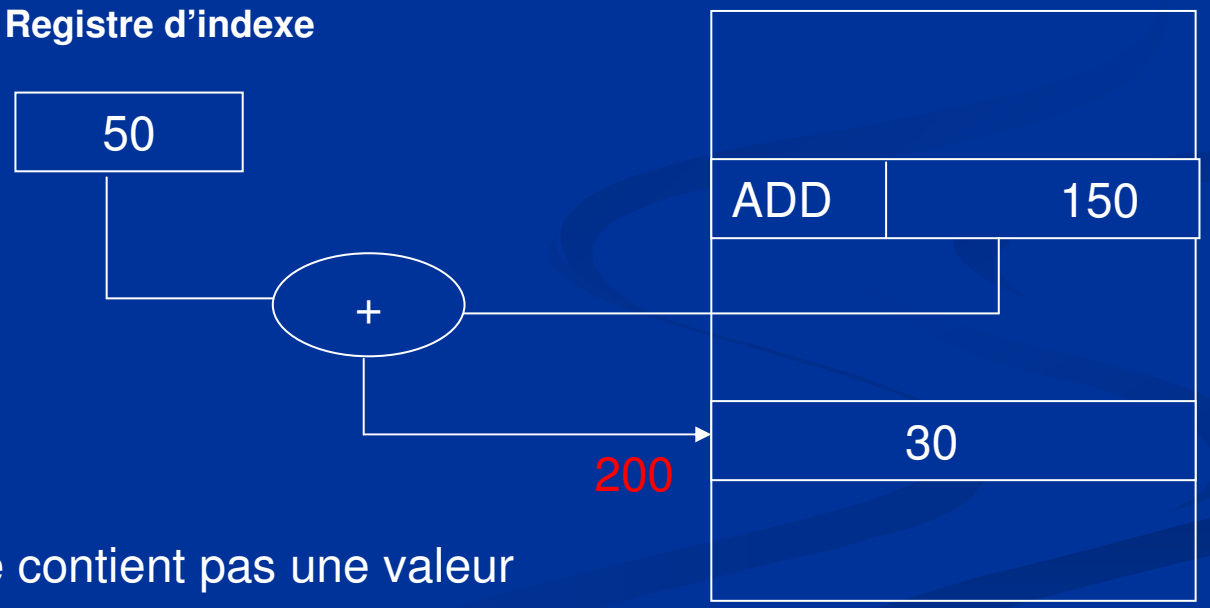
- Exemple :
- Initialement l'accumulateur contient la valeur 20
- Il faut récupérer l'adresse de l'adresse (150).
- Récupérer l'adresse de l'opérande à partir de l'adresse 150 (la valeur 200)
- Récupérer la valeur de l'opérande à partir de l'adresse 200 (**la valeur 40**)

Additionner la valeur 40 avec le contenu de l'accumulateur (20) et nous allons avoir la valeur **60**



Adressage indexé

- L'adresse effectif de l'opérande est relatif à une zone mémoire.
- L'dresse de cette zone se trouve dans un registre spécial (registre indexe).
- Adresse opérande = ADR + (X)



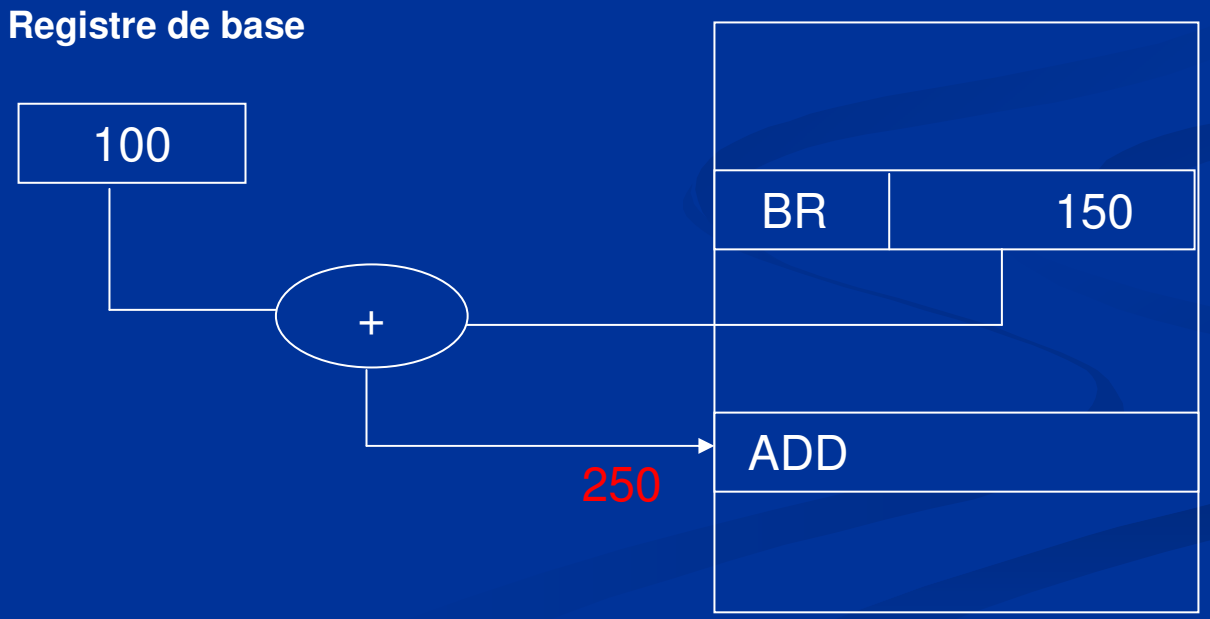
Remarque : si ADR ne contient pas une valeur immédiate alors

Adresse opérande = (ADR)+ (X)

Adressage relatif

- L'adresse effectif de l'opérande est relatif a une zone mémoire.
- L'dresse de cette zone se trouve dans un registre spécial (registre de base).
- Ce mode d'adressage est utilisée pour les instructions de branchement.

  Adresse = ADR + (base)



Cycle d'exécution d'une instruction

- Le traitement d'une instruction est décomposé en trois phases :
  - Phase 1 : rechercher l'instruction à traiter et décodage
  - Phase 2 : rechercher de l'opérande et exécution de l'instruction
  - Phase 3 : passer à l'instruction suivante

- Chaque phase comporte un certain nombre d'opérations élémentaires (microcommandes) exécutées dans un ordre bien précis (elle sont générées par le séquenceur).
- La phase 1 et 3 ne change pas pour l'ensemble des instructions , par contre la phase 2 change selon l'instruction et le mode d'adressage

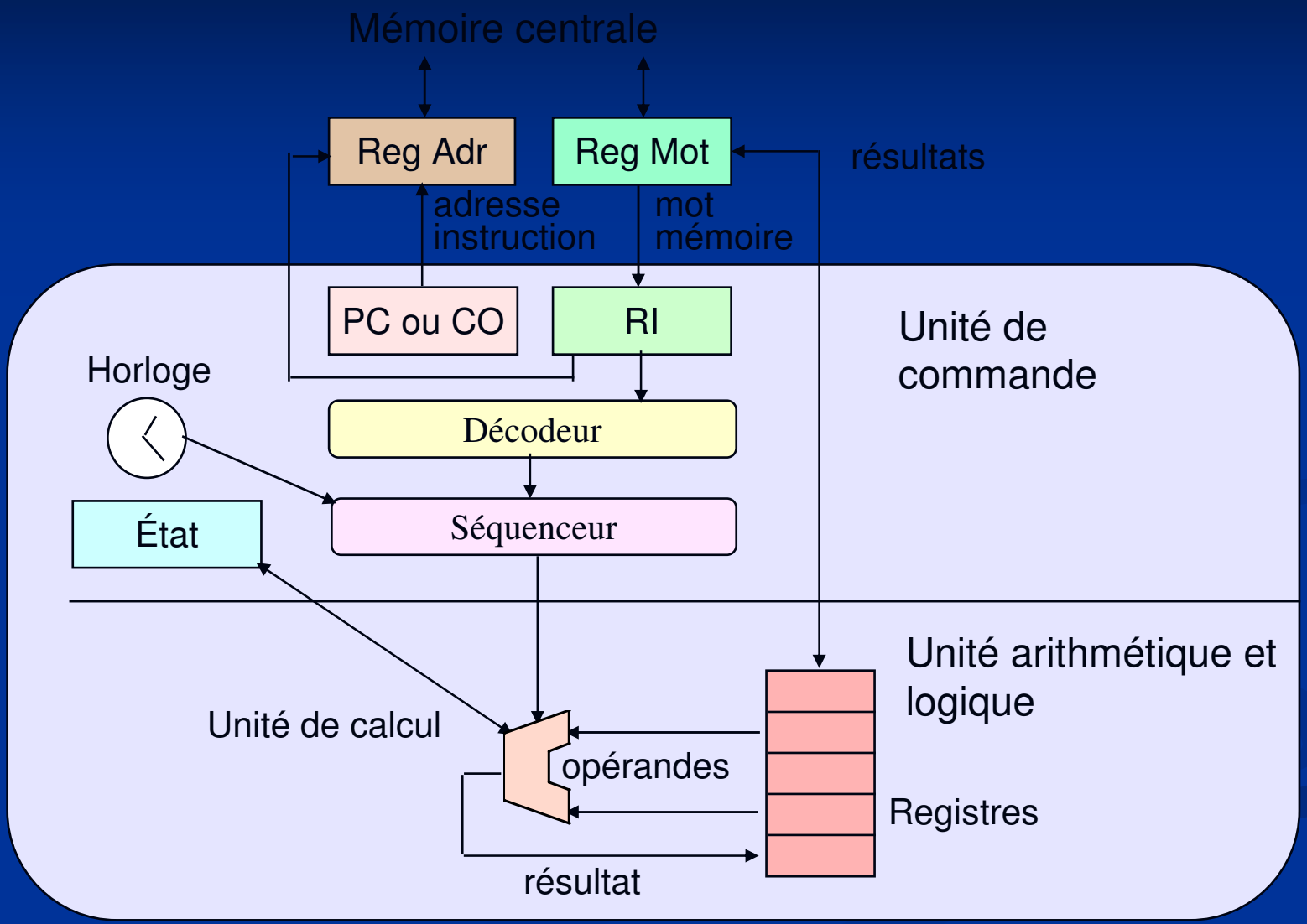
- Exemple1 : déroulement de l'instruction d'addition en mode immédiat
  ACC←(ACC)+ Valeur

  - Phase 1 : (rechercher l'instruction à traiter)
    - Mettre le contenu du CO dans le registre RAM RAM← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert du contenu du RIM dans le registre RI RI← (RIM)
    - Analyse et décodage
  - Phase 2 : (traitement)
    - Transfert de l 'opérande dans l'UAL UAL ← (RI).ADR
    - Commande de l'exécution de l'opération (addition)
  - Phase 3 : (passer à l'instruction suivante)
    - CO ← (CO)+ 1

- Exemple 2 : déroulement de l'instruction d'addition en mode direct ACC←(ACC)+ (ADR)
  - Phase 1 : (rechercher l'instruction à traiter)
    - Mettre le contenu du CO dans le registre RAM RAM← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert du contenu du RIM dans le registre RI RI← (RIM)
    - Analyse et décodage
  - Phase 2 : (décodage et traitement)
    - Transfert de l'adresse de l 'opérande dans le RAM RAM← (RI).ADR
    - Commande de lecture
    - Transfert du contenu du RIM vers l'UAL UAL ←(RIM)
    - Commande de l'exécution de l'opération (addition)
  - Phase 3 : (passer à l'instruction suivante)
    - CO ← (CO)+ 1

- Exemple 3 : Déroulement de l'instruction d'addition en mode indirect ACC←(ACC)+ ((ADR))

  - Phase 1 : (rechercher l'instruction à traiter)
    - Mettre le contenu du CO dans le registre RAM RAM← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert du contenu du RIM dans le registre RI RI← (RIM)
    - Analyse et décodage
  - Phase 2 : (décodage et traitement)
    - Transfert de l'adresse de l 'opérande dans le RAM ← (RI).ADR
    - Commande de lecture /* récupérer l'adresse */
    - Transfert du contenu du RIM vers le RAM RAM←(RIM)
    - Commande de lecture /* récupérer l'opérande */
    - Transfert du contenu du RIM vers l'UAL UAL ← (RIM)
    - Commande de l'exécution de l'opération (addition)
  - Phase 3 : (passer à l'instruction suivante)
    - CO ← (CO)+ 1


Mémoire centrale
Reg Adr
Reg Mot
résultats
adresse instruction
mot mémoire
PC ou CO
RI
Unité de commande
Horloge
Décodeur
État
Séquenceur
Unité arithmétique et logique
Unité de calcul
opérandes
Registres
résultat