



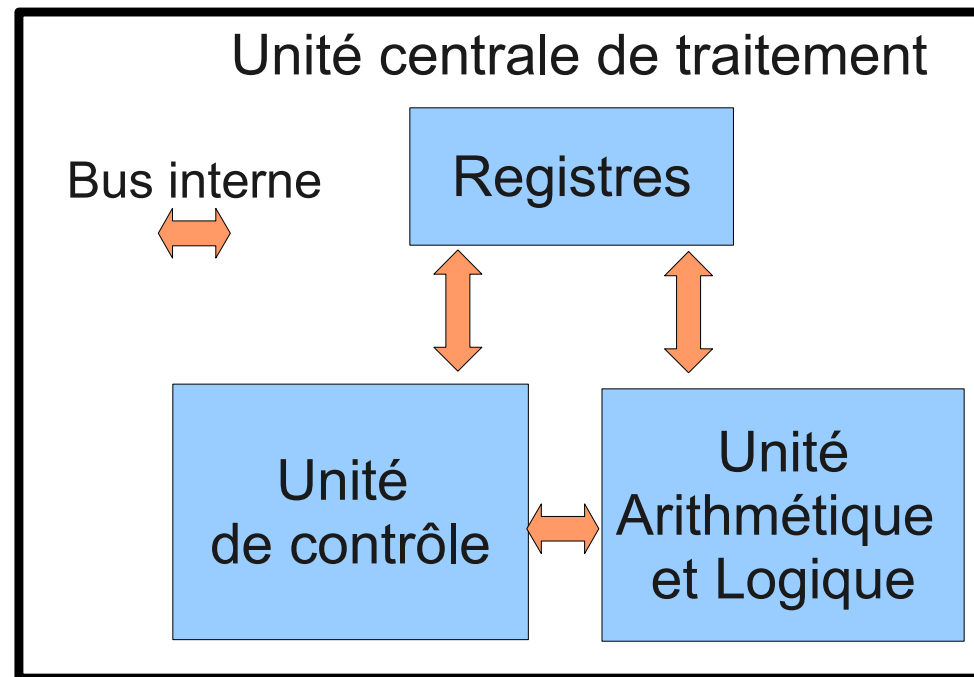
+ L'Unité Centrale de Traitement

+ Introduction

63

- L'exécution d'un programme par un ordinateur correspond à l'exécution d'une séquence d'instructions
- L'organe central permettant d'exécuter cette séquence d'instructions est l'unité de traitement

+ Architecture de l'unité de traitement



+ Les registres

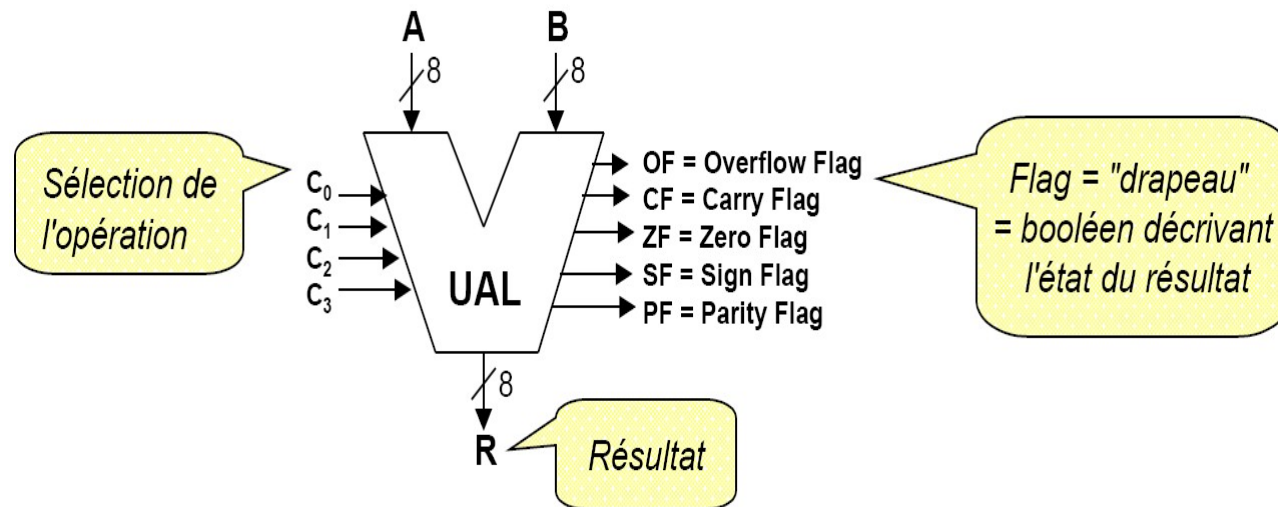
- Ce sont des unités de stockage rapides (temps d'accès faible), mais de faible capacité de stockage intégrées directement à l'UCT
- Il existe plusieurs types de registres dont certains correspondent à un rôle spécifique pour le fonctionnement de la machine tels que:
 - Le registre d'instruction (IR) qui contient l'instruction en cours d'exécution par l'unité de traitement. Cette instruction doit d'abord être décodée par l'unité de contrôle
 - Le compteur ordinal (CO), également appelé "compteur programme" (CP ou IP pour "Instruction Pointer" en anglais) qui contient l'adresse de la prochaine instruction que doit exécuter l'unité de traitement
 - Le registre d'état qui mémorise l'état de l'UAL à l'issue d'une opération. Ce registre est formé de bits "drapeaux" ou indicateurs
 - Les registres tampons qui mémorisent l'état des registres de données et d'adresse et qui sont des registres d'interface avec les bus de données et d'adresse

+ Les registres

- En plus de ces registres spécifiques, on trouve également des registres généraux qui servent à:
 - Stocker les opérandes de l'UAL
 - Stocker les résultats de l'opération réalisée par l'UAL
- Ces registres généraux sont utilisés pour stocker temporairement les données manipulées par l'UCT

+ L'Unité Arithmétique et Logique

- Réalise comme son nom l'indique des opérations arithmétiques (addition, soustraction, multiplication, division, ...) et des opérations logiques (ET, OU, NON, ...), ainsi que des opérations sur les bits telles que le décalage de bits dans les registres



+ L'unité arithmétique et logique

Il existe à la sortie de l'UAL des indicateurs ou flags qui sont des bits regroupés dans un registre appelé "registre d'état". Ils sont mis à jour dès qu'un chargement ou une opération logique ou arithmétique est effectuée. Les principaux indicateurs sont :

- **CF ou Carry Flag** (Drapeau de retenue) : Ce drapeau est armé si une opération arithmétique génère une retenue sur le bit le plus significatif (bit de poids fort). Le drapeau est désarmé dans les autres cas. Ce drapeau indique ainsi une condition de débordement en arithmétique entière non signée. Si CF=0, le résultat de l'opération est correct, sinon il ne l'est pas
- **PF ou Parity Flag** (Drapeau de parité) : Armé si l'octet de poids faible (octet le moins significatif) du résultat généré après une opération arithmétique contient un nombre pair de bits à 1. Le drapeau est désarmé dans les autres cas.
- **ZF ou Zero Flag** (Drapeau zéro) : Armé si le résultat d'une opération arithmétique vaut zéro. Le drapeau est désarmé dans les autres cas.
- **SF ou Sign Flag** (Drapeau de signe) : Armé si le résultat d'une opération arithmétique possède un bit de poids fort (bit le plus significatif) à 1, indiquant ainsi un nombre signé. Le drapeau est désarmé dans les autres cas (indiquant possiblement un résultat non signé, c'est-à-dire positif).

+ L'unité de commande ou de contrôle

- Contrôle l'ensemble des éléments électroniques de l'UCT en générant des signaux de commande par le décodage de l'instruction à traiter
- Le décodage de l'instruction courante disponible dans le registre instruction (RI) entraîne l'exécution de micro-instructions sollicitant le chemin de données, par la génération de signaux de commande
- En effet l'exécution d'une instruction se fait de manière matérielle grâce à l'UAL, mais pour cela il faut que les registres généraux stockant les opérandes soient correctement initialisés et que l'opération adéquate soit chargée dans l'UAL. C'est le rôle de l'unité de commande

+ L'unité de commande ou de contrôle

- Elle est elle-même constituée de plusieurs composantes, qui sont les éléments permettant d'exécuter des micro-instructions
- Chaque instruction est en effet constituée d'un ensemble de micro-instructions (et l'ensemble des instructions constitue le programme)
- Chaque micro-instruction est stockée dans une mémoire interne (non volatile) à l'unité de commande, et permet l'application d'une opération élémentaire (addition, soustraction, opération logique) qui sollicite l'ensemble de ce que l'on appelle le "chemin de données" ("datapath" en anglais):
 - Les registres
 - L'UAL
 - Les bus internes

+ L'unité de commande ou de contrôle

- Pour permettre l'exécution d'une instruction (et donc de l'ensemble de micro-instructions la constituant), l'unité de commande produit les signaux de commandes contrôlant les autres unités (registres, UAL, bus)
- L'unité de commande permet de faire l'interface entre les instructions logicielles et les circuits électroniques constituant une UCT

+ Instructions et cycle d'instruction

- Les instructions forment un ensemble de commandes spécifiant à l'UCT l'opération à effectuer
- Les instructions sont transmises au processeur au format binaire mais elles sont écrites par le développeur dans un format plus compréhensible pour lui : les mnémoniques (ou langage assembleur)
- La transformation d'un programme mnémonique en binaire se fait à l'aide d'un logiciel d'assemblage (appelé simplement "assembleur")

+ Instructions et cycle d'instruction

■ Exemple de langage mnémonique:

- `MOV dst,src` : place la valeur de `src` dans `dst`.
- `ADD dst,src` : additionne `src` à `dst` et stocke le resultat dans `dst`

- Avant l'exécution d'un programme, l'ensemble des instructions machines (binaires) constituant ce programme, se trouve dans une mémoire de masse
- Au moment du lancement du programme, l'ensemble des instructions est transféré dans la mémoire principale (pour un accès plus rapide par l'unité centrale de traitement)

+ Instructions et cycle d'instruction

- Une instruction binaire est une séquence de bits sur un nombre d'octets variables (de 1 à 4 octets généralement)
- La représentation binaire de l'instruction comprend 2 parties:
 - Un code d'opération (CODOP) désignant l'opération particulière que l'UCT doit effectuer. Le CODOP permet de savoir également la façon d'interpréter les opérandes
 - Les opérandes ou l'adresse des opérandes



+ Instructions et cycle d'instruction

■ Exemple:

opération voulue:

charger la valeur 12h dans le registre accumulateur (AL)



écriture sous forme de mnémonique (langage assembleur 8086)

mov AL, 12h



traduction en langage machine (assemblage)

10110000 00010010

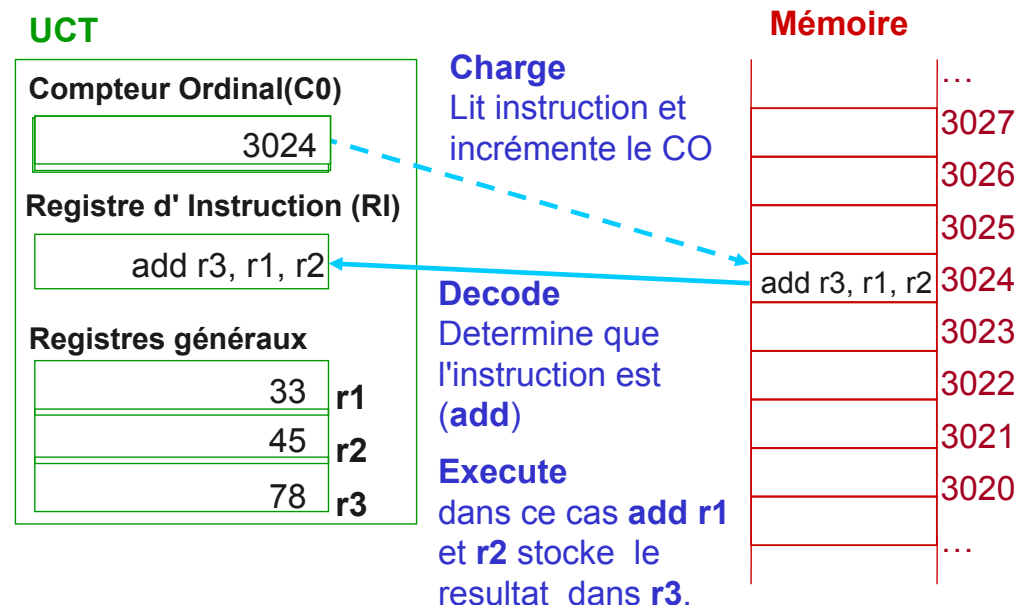
B 0 1 2

+ Instructions et cycle d'instruction

- L'instruction indique donc à l'UCT ce qu'elle doit exécuter, et sur quoi l'exécuter
- L'UCT exécute chaque instruction du programme en commençant par l'instruction 0 et en suivant les étapes du cycle d'instruction:
 1. Chargement: l'UCT lit en mémoire la prochaine instruction à exécuter
 2. Décodage: cette phase consiste à interpréter l'instruction qui se trouve dans le registre RI
 3. Exécution: l'opération spécifiée est exécutée
 4. Stockage: les résultats de l'opération sont stockés

+ Instructions et cycle d'instruction

- Après l'exécution d'un cycle, l'UCT exécute l'instruction suivante et donc relance un nouveau cycle (sauf si le programme est terminé)
- Exemple de cycle d'instruction:



+ Du logiciel au matériel

- Le lien entre un programme informatique et la partie électronique se fait à travers plusieurs niveaux
- La partie électronique est commandée par un ensemble de micro-instructions permettant de commander les différentes composantes de l'UCT et de synchroniser les différents éléments du chemin de données
- Les micro-instructions sont générées par l'unité de commande
- Chaque instruction correspond à une séquence de micro-instructions
- L'ensemble des instructions disponibles pour un processeur donné constitue le **langage machine**

+ Du logiciel au matériel

- Chaque instruction associée à un processeur est représentée par un code binaire (difficile à retenir pour le développeur)
- Le langage assembleur correspond à une traduction de chaque instruction binaire en un code (généralement une abréviation d'un mot anglais) appelé **mnémonique** plus compréhensible et plus intuitif
- Un programme informatique est constitué d'un ensemble d'instructions machines appelé **exécutable**
- Pour transformer un programme écrit en assembleur en langage machine, on utilise un logiciel dont le rôle est de traduire chaque instruction assembleur en une instruction en langage machine (assemblage)

+ Du logiciel au matériel

- L'ensemble des instructions à exécuter par l'UCT lors du déroulement d'un programme est stocké dans la mémoire centrale
- Chaque instruction occupe en mémoire un nombre variable d'octets (de 1 à 4 octets) en fonction de la complexité de l'instruction et en fonction des processeurs

+ Du logiciel au matériel

- Exemple: on observe en mémoire à l'adresse 0100h le contenu binaire suivant (écrit ici en hexadécimal dans un souci de lisibilité):

A1100103061201A31401

Cette séquence correspond à un programme permettant d'additionner le contenu de 2 cases mémoire et de stocker le résultat dans une troisième case mémoire

- La transcription de ce programme en langage machine et en assembleur serait:

Adr	Mémoire	Mnémonique	Actions
0100	A1 10 01	MOV AX, [0110]	Charger AX avec le contenu de 0110
0103	03 06 12 01	ADD AX, [0112]	Ajouter le contenu de 0112 à AX
0107	03 14 01	MOV [0114], AX	Stocker le contenu de AX en 0114