



Centre Universitaire
Belhadj Bouchaib
Ain Témouchent

Techniques de programmation 1 (Microprocesseur).

1. Généralités

- Un **microprocesseur** se présente sous la forme d'un circuit intégré muni d'un nombre généralement important de broches.
- **Fonctionnement d'un microprocesseur** : Un microprocesseur exécute un programme. Le programme est une suite d'instructions stockées dans la mémoire. Une instruction peut être codée sur un ou plusieurs octets.
- Une instruction est définie par son code opératoire, valeur numérique binaire difficile à manipuler par l'être humain. On utilise donc une notation symbolique pour représenter les instructions : **les mnémoniques**. Un programme constitué de mnémoniques est appelé programme en assembleur. Les instructions peuvent être classées en groupes :
 - instructions de transfert de données ;
 - instructions arithmétiques ;
 - instructions logiques ;
 - instructions de branchement ...

2. Programmation en assembleur du microprocesseur 8086

2.1. Les instructions de transfert

Exemple 1 :

mov ax,bx : charge le contenu du registre **bx** dans le registre **ax**. Dans ce cas, le transfert se fait du **registre bx** (source) vers un autre **registre ax** (destination).

Exemple 2 :

mov al, 12H : charge le **registre al** (destination) avec la **valeur immédiate** (source) **12H**. La donnée est fournie immédiatement avec l'instruction.

Exemple 3 :

mov bl, [1200H] : cette instruction réalise le transfert du contenu de la **case mémoire** (source) d'adresse effective (offset) **1200H** vers le **registre bl** (destination). L'instruction comporte l'adresse de la case mémoire où se trouve la donnée.

Exemple 4 :

mov [1200H], al : cette instruction réalise le transfert du contenu du **registre al** (source) vers la **case mémoire** (destination) d'adresse effective (offset) **1200H**. L'instruction comporte l'adresse de la case mémoire où la donnée va être transférée.

Exemple 5 : mov [1200H], 12H : cette instruction réalise le transfert de la **valeur immédiate** (source) **12H** vers la **case mémoire** (destination) d'adresse effective (offset) **1200H**. L'instruction comporte l'adresse de la case mémoire ou la donnée va être transférer.

2.2. Les instructions arithmétiques

- Addition : ADD opérande 1, opérande2

L'opération effectuée est : opérande1 \leftarrow opérande1 + opérande2.

Exemple :

- **add ah,[1100H]** : ajoute le contenu de la case mémoire d'offset **1100H** à l'accumulateur **AH** (adressage direct);
- **add ah,[bx]** : ajoute le contenu de la case mémoire pointée par **BX** à l'accumulateur **AH** (adressage basé);

- Soustraction : SUB opérande 1, opérande 2

Exemple :

- **sub ah,[1100H]**
- **sub ah,[bx]**

- Multiplication : MUL opérande, où opérande est un registre ou une case mémoire.

Cette instruction effectue la multiplication du contenu de AL par un opérande sur 1 octet ou du contenu de AX par un opérande sur 2 octets. Le résultat est placé dans AX si les données à multiplier sont sur 1 octet (résultat sur 16 bits), dans (DX,AX) si elles sont sur 2 octets (résultat sur 32 bits).

Exemple :

- **mov al,51**
- **mov bl,32**
- **mul bl**

- Division : DIV opérande, où opérande est un registre ou une case mémoire. Cette instruction effectue la division du contenu de AX par un opérande sur 1 octet ou le contenu de (DX,AX) par un opérande sur 2 octets. Résultat : si l'opérande est sur 1 octet, alors AL = quotient et AH = reste ; si l'opérande est sur 2 octets, alors AX = quotient et DX = reste.

Exemples :

- **mov ax,35**
- **mov bl,10**
- **div bl**

→ **AL = 3 (quotient) et AH = 5 (reste)**

- **mov dx,0**
- **mov ax,1234**
- **mov bx,10**
- **div bx**

→ **AX = 123 (quotient) et DX = 4 (reste)**