

TP°02 : Gestion de la mémoire du microprocesseur **8086**

Objectifs :

- Utilisation des variables et les constantes dans un programme en assembleur 8086.
- Utilisation des différents modes d'adressage ;
- Savoir gérer la mémoire centrale du μ P.

1. Utilisation des variables dans le programme

- 1- Lancer emu8086
- 2- Choisissez l'extension .exe afin de pouvoir allouer les zones de code, data, extra et la pile dans le segment, puis reprendre le programme ci-dessous.

```
DSEG  SEGMENT 'DATA'
```

```
    ; insérez les données (variables) ici!
```

```
    val1 dw 10
```

```
    Val2 dw 65535
```

```
DSEG  ENDS
```

```
SSEG  SEGMENT STACK 'STACK'
```

```
    ; cette zone sera réservée pour stocker des données dans la pile dont la taille
```

```
    ; de cette région de la mémoire est 256 octets
```

```
    db 256 DUP(0)
```

```
SSEG  ENDS
```

```
CSEG  SEGMENT 'CODE'
```

```
    ; cette région sera réservée pour le programme
```

```
assume cs : CSEG, ds:DSEG, ss :SSEG
```

```
START PROC FAR
```

; Déterminer les registres segments :

MOV AX, DSEG

MOV DS, AX

MOV ES, AX

; insérer votre code

Mov ax,val1

mul val2

; exit to operating system.

MOV AX, 4C00h

INT 21h

START ENDP

CSEG ENDS

END START

3- Exécuter le programme (de préférence en mode pas à pas) :

- a- Donner l'adresse logique des variables val1 et val2.
- b- Donner les registres qui possèdent le résultat de la multiplication val1*val2.
- c- Quels sont les flags affectés après l'opération de la multiplication ?
- d- Utiliser un adressage direct pour effectuer la multiplication.
(Représenter les modifications requises pour réaliser cet adressage)
- e- Créer une nouvelle variable et nommer la val3 avec une valeur initiale égale à 256.
 - En Utilisant un adressage indirect basé, effectuer la division du résultat sur le contenu du val3.
 - Stocker le résultat à l'adresse 2000h
 - Empiler (stocker dans la pile) le résultat de la division.
- f- En utilisant un adressage basé, réaliser la somme des trois variables val1, val2 et val3.