

TD N° 5

Exercice 1 :

1. Définir dans le segment de donnée deux cases mémoires, en mots, A et B.

A DW ?

B DW ?

2. Ecrire un programme permettant d'échanger les contenus de A et B.

ORG 100h	Directive
Mov AX, A Mov DX, B Mov B, AX Mov A, DX	Instructions
A DW 10 B DW 100	Déclarations

Exercice 2 :

1. Ecrire un programme qui fait la somme de deux variables de type mot (précédemment déclarées) et qui sauvegarde le résultat dans une troisième variable de type mot. Faites des tests avec des valeurs positives et négatives.

ORG 100h	Directive
Mov AX, A Add AX, B Mov C, AX RET	Instructions
A DW 10 B DW -100 C DW 0	Déclarations

Valeur de C = FFA6h = -90 (Complément à deux en mode 16 bits)

2. Ecrire un programme qui fait la somme de deux variables de type octet (précédemment déclarées) et qui sauvegarde le résultat dans une troisième variable de type mot.

ORG 100h	Directive
Xor AX, AX ; Equivalent à : Mov AX, 0 Mov AL, A Add AL, B Mov C, AX ; On ne peut pas passer AL (8 bits) RET	Instructions
A DB 10 B DB -100 C DW 0	Déclarations

3. Ecrire un programme qui fait la somme de deux variables de type mot (précédemment déclarées) et qui sauvegarde le résultat dans une variable de type octet si cela est possible ou dans une variable de type mot si le résultat ne tient pas dans une variable de type octet. Faire des tests avec les valeurs signées.

On ne peut pas affecter une variable de type mot à un type octet. On sauvegarde le résultat de la somme dans une variable de type mot. Voir la réponse de la question 1. On peut résoudre l'exercice en utilisant les branchements (Série 6).

Exercice 3 :

Tracer le programme ci-dessous en indiquant à chaque fois la valeur des indicateurs CF et OF

a. Mov AL, 64h ; CF = 0 et OF = 0
Mov BL, 2 ; CF = 0 et OF = 0
Mul BL ; CF = 0 et OF = 0

b. Mov AL, 64h ; CF = 0 et OF = 0
Mov CL, 3 ; CF = 0 et OF = 0
Mul CL ; CF = 1 et OF = 1

CF = OF = 0 quand AH (dans le cas de 8 bit) ou DX (dans le cas de 16 bit) est égal à zéro.

Exercice 4 :

Dans un registre de 8 bits, effectuer des opérations sur des nombres signés en donnant leurs résultats et en positionnant les indicateurs :

C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀
1	0	1	1	0	0	0	0
	1	0	1	1	0	0	0
+		1	0	1	1	1	0
=	0	1	1	0	1	1	0

SF = 0 CF = C₇ = 1

ZF = 0 OF = C₆ xor C₇ = 1

C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀
1	1	1	1	0	0	0	0
	1	1	1	1	0	0	0
+		0	0	0	1	0	0
=	0	0	0	0	0	0	0

SF = 0 CF = C₇ = 1

ZF = 1 OF = C₆ xor C₇ = 0

Exercice 5 :

Donner le contenu de AL et l'état des indicateurs ZF, CF, SF et OF après l'exécution des programmes suivants :

- MOV AL, 79h
ADD AL, 30h
; Le contenu de AL est A9h, ZF = 0, CF = 0, SF = 1 et OF = 1

- MOV AL, A0h
ADD AL, A0h
; Le contenu de AL est 140h, ZF = 0, CF = 1, SF = 0 et OF = 1

- MOV AL, 0h
SUB AL, 1h
; Le contenu de AL est FFh, ZF = 0, CF = 1, SF = 1 et OF = 0

Exercice 6 :

1. Écrire un programme qui calcule $x \times 2^n$, où x et n sont deux variables positives sur 16 bits, stockées aux adresses 130h et 132h. Le résultat sera rangé dans 134h. On utilisera une instruction de décalage bit à bit.

ORG 100h

... ; Initialisation des cases mémoires avec x et n

Mov AX, [130h]

Mov CL, [132] ; [132h] doit être de type octet !

SHL AX, CL

Mov [134h], AX

RET

2. A l'aide de l'instruction SHR, écrire un programme qui divise par 8 la valeur contenue à l'adresse 0130h, et qui range le résultat en 0131h.

ORG 100h

... ; Initialisation de la case mémoire avec x

Mov AL, [130h]

Mov CL, 3

SHR AL, CL ; ou bien directement : SHR AX, 3

Mov [131h], AL

RET

Exercice 7 :

1. Écrire un programme qui calcule $r = x \times y$, où x , y et r sont des variables naturelles de

- a. 16 bits, rangées respectivement en 130h, 132h et 134h

...

Mov AX, [130h]

Mov CX, [132h]

Mul CX

Mov [134h], AX

Mov [136h], DX

b. 8 bits, rangées respectivement en 80h, 81h et 82h

...

Mov AL, [80h]

Mov CL, [81h]

Mul CL

Mov [82h], AL

Mov [83h], AH

2. Écrire un programme qui calcule x / y , où x, y sont des variables naturelles :

a. x est 16 bits et y est de 8 bits, rangées respectivement en 130h, 132h et le résultat à partir de 134h

...

Mov AX, [130h]

Mov CL, [132h]

Div CL

Mov [134h], AL

Mov [135h], AH

b. x est de 32 bits et y est de 16 bits, rangées respectivement en 80h, 90h et le résultat à partir de 82h

...

Mov AX, [80h]

Mov DX, [82h]

Mov CX, [90h]

Div CX

Mov [82h], AX

Mov [84h], DX