

Série n°3 : Registres et Adressages.

Exo 1 :

$$\textcircled{a} \text{ phy} = 16 \times \text{Segment} + \text{Offset}.$$

SS = SP ou BP

$$\text{SS} \cdot \text{SP} : F200H \times 16 + 0FFCH = F2FFCH$$

$$\text{SS} \cdot \text{BP} : F200H \times 16 + 9000H = F29000H$$

CS avec IP.

$$\text{CS} \cdot \text{IP} : 16 \times 5000H + 1000H = 51000H$$

DS : SI

$$\textcircled{a} \text{ phy} = 16 \times \overset{\text{DS}}{6000} + \overset{\text{SI}}{2000} = 62000H$$

$$\textcircled{a} \text{ phy} : \text{DS} \cdot \text{DI} = 16 \times 6000H + 8000H = 68000H$$

$$\textcircled{a} \text{ phy} = 16 \times \text{ES} \cdot \text{DI} = 04E50 + 8000 = 0CE50H$$

Exo 2 :

$$\textcircled{a} \text{ phy}_A = 16 \times \text{DS} + 0 = 00100H \quad \left| \begin{array}{l} 64K0 - 8K0 = 56K0 \end{array} \right.$$

$$\textcircled{b} \text{ phy}_B = \textcircled{a} \text{ phy}_A + 5000H = 05100H \quad \left| \begin{array}{l} 56K0 = F000H \end{array} \right.$$

$$\textcircled{c} \text{ phy}_C = \textcircled{b} \text{ phy}_B + 1FFFH = 100FFH \quad \left| \begin{array}{l} 8K0 = 1FFFH \end{array} \right.$$

$$\textcircled{d} \text{ phy}_D : 10500H \times 16 + 0 = 101000H$$

$$\textcircled{e} \text{ phy}_E = 10500H + F000H = 1F500H \quad \left| \begin{array}{l} 64K0 - 4K0 \\ = 60K0 \\ = F000H \end{array} \right.$$

$$@phy_F = 1F4FF + 1000 = 204FFH$$

$$4K = 1000H$$

$$G = 20800H$$

$$64K - 20K = 44K = B000H$$

$$H = 20800H + B000 - 1 = 2B7FFH$$

$$64K = FFFFH$$

$$I = 20800 + FFFF = 307FFFH$$

$$2) @A = 00100H$$

$$@B = 0E100H$$

$$@C = 100FFH$$

$$@D = @C + 1 = 10100H$$

$$@E = @D + \underbrace{60K}_{\text{space libre}} = 1F100H$$

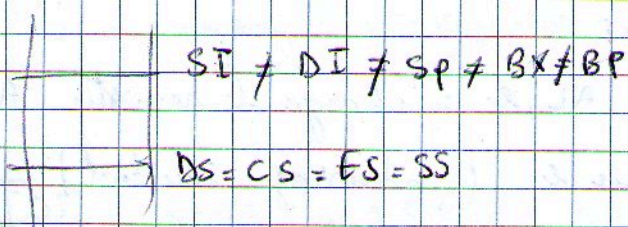
$$@F = @D + \underbrace{64K}_{\text{space libre}} = 200FFH$$

$$@G = @F + 1 = 20100H$$

$$@H = @G + \underbrace{44K}_{B000H} = 2B100H$$

$$@I = @G + FFFF = 300FFH$$

3)



Exo 3 :

1) call sous-programme

PUSH AX

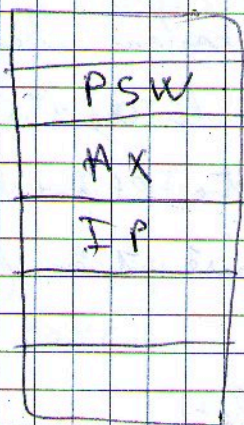
PUSH PSW \leftarrow registre d'état

POP PSW

POP AX

RET

2)



SS: SP + 6 (003CH) = SP

SS: SP + 4 (003AH)

SS: SP + 2 (0038H)

SS: SP (0036H)

Exo 4 :

- 1) `mov AL, 2` ; charge le registre AL avec la valeur 2 (adressage immédiat)
- 2) `mov AL, [25]` ; charge le registre AL par le contenu de la case mémoire d'offset 25 (adressage direct)
- 3) `mov [B4F], AX` ; charger le contenu du registre AX dans les cases mémoires B4F et B4F + 1 (B50)
- 4) `mov AX, [B4F]` ; charger le registre AX par le contenu de la case mémoire d'offset est B4F et B50.
- 5) `mov AL, [DI]` ; charger le contenu de la case mémoire pointée par DI dans le registre AL.
- 6) `mov BP, 05` ; charger le registre BP par le nbre 5.
- 7) `mov [BP], 05` ; Adressage basé et transfert de la valeur 05 vers `SS:[BP]`
(`[BP]` représente l'adresse de l'offset).
- 8) `mov AL, DS:[BX]` ; charger AL avec le contenu de l'offset `[BX]` adressage basé.

10) `mov DS:[DI], AL`; transférer le contenu de AL à l'offset.

11) `mov AL, SS[DI]`; charger le registre AL, avec le contenu de la case mémoire dont l'offset est pointé par DI

Exo 5:

$BX = D74F$

$BL = 4F$

$BH = D7$

Exo 6:

$SI = 2000H$; $AX = B5C7H$

`mov [SI], AX`

$DS:SI = C7H$

$DS:SI+1 = B5H$

`mov BX, [DI]`; charger le registre BX avec le contenu de deux cases mémoires dont l'offset est

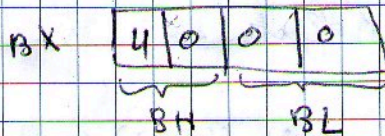
4044H
4045H

`mov BX, DI` charger le contenu de DI dans le registre BX.

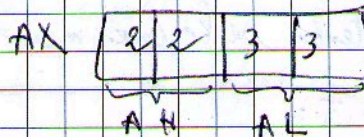
BX	40	44
	BH	BL

Exo 7:

mov bx, 4000H



mov ax, 2233H



mov [bx], ax

mov word [bx+2], 4455H

mov byte [bx+4], 66

mov byte [bx+5], 77

4005H	77
4004H	66
4003H	44
4002H	55
4001H	22
4000H	33

Exo 8:

a) mov [SI], AL

DS:SI

$$@ \text{phy} = \text{DS} \times 16 + \text{SI}$$

$$= 0B9C0 + 0021$$

$$= 0B9E1 \text{ H}$$

le mode d'adressage : adressage index

mov [SI+1], AH

$$\text{offset SI, 1} = 0021 + 1 = 0022 \text{ H}$$

$$@ \text{logique} = 0B9C \times 16 = 0B9C0 \text{ H}$$

$$@ \text{ physique} = 0B9C0H + 0022H = 0B9E2H$$

mode d'adressage : index.

mov [SI], AX;

$$@ = DS \times 16 + SI = 0B9C0 + 0021 = 0B9E1$$

mode d'adressage : index.

mov [DI + 100], CS;

$$\text{adresse physique} : 0B9C0 + [DI + 100]$$

$$= 0B9C0 + [DI + 100]$$

$$= 0B9C0H + (2042H + 100)$$

$$= 0B9C0H + 2042H + 64H$$

$$@ \text{ phy}_1 = 0B9C + 20A6H = 0DA66H$$

$$@ \text{ phy}_2 = 0DA66H + 1 = 0DA67H$$

adresse indexé.

mov [BX], DX;

$$@ \text{ phy} = 16 \times DS + BX$$

$$= 16 \times 0B9C H + 100C H = 0B9C0H + 100C H$$

$$= 0C9CCH$$

adresse basé

mov [SI + BX + 81H], CX;

$$@phy = 16DS + SI + BX + 81H$$

$$= 16 \times 0B9CH + 0021H + 100CH + 81H$$

$$= 06A6FH$$

adressage : basé indexé.

g) mov [BP + 150H], AX

$$@phy = 16 \times SS + BP + 150H$$

$$= 0C400H + 555DH = 1195DH$$

adressage basé.

h) mov [BP][DI + 75H], BX;

$$@phy = 16 \times SS + BP + DI + 75H$$

$$@phy_1 = 138B7H$$

$$@phy_2 = 138B8H$$

adressage basé indexé.

Exo 9 :

à revoir le support de cours page 47

"Utilisation de la pile pour le passage de paramètres"