Explication: Chap 3 Assembleur / Code machine

1) La syntaxe de déclaration des segments est de la forme suivante :

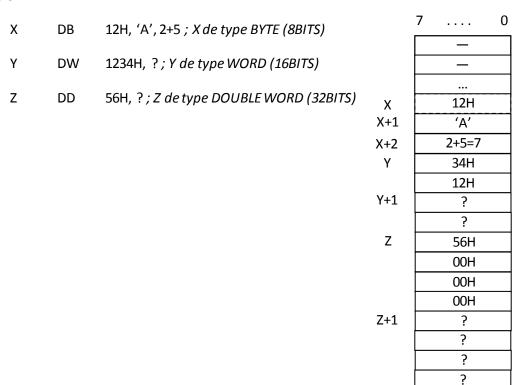
<Nom_Segment> **SEGMENT** Contenu du segment <Nom_Segment> **ENDS** <Nom_Segment>: est un nom au choix du programmeur. Exemple: MesVar **SEGMENT** ; Commentaire ; Mes instructions MesVar **ENDS** MonCode **SEGMENT** ASSUME DS:MesVar, CS:MonCode @MesVar ; AX ← l'adresse logique de base du segment MesVar Debut: MOV AX, MOV DS, AX@MonCode ; $AX \leftarrow l'$ adresse logique de base du segment MonCode MOV AX, AXMOV CS, ; Les instruction de mon programme : La somme de AX et BX ADD AX, BX ; $AX \leftarrow AX + BX$ MonCode ENDS **END** Debut

2) La syntaxe de déclaration des variables :

2.1) Variables simples:

<NomVariable> TYPE ValeurInitialisation1,, ValeurInitialisationN

Exemple:



3) Instructions assembleurs:

Elles sont de la forme : Operateur ArgDest (Argument destinataire), ArgSrce (Argument source)

- 3.1) Instructions de déplacement des données :
 - a) MOV ArgDest, ArgSrce; ArgDest ← ArgSrce

MOV AL,
$$20H$$
; $AL = 20H$

PUSH ArgSrce; Empiler ArgSrce = Placer La valeur de ArgSrce dans la pile

POP ArgDest; Depiler la valeur du sommet de Pile et l'affecter à ArgDest : ArgDest ← ; Sommet de pile

Exemple d'écriture d'un programme calculant la somme de deux variables X et Y

MyData SEGMENT

X DB 54H

Y DB 78H

MyData ENDS

MyCode SEGMENT

ASSUME DS : MyData, CS : MyCode

Main: MOV AX, @MyData

MOV DS, AX

MOV AX, @MyCode

MOV CS, AX

; LA SOMME

MOV AX, Y ; $AX \leftarrow Y$

ADD AX, X ; $AX \leftarrow AX=Y+X$

MyCode ENDS

END Main