

TP N° 1 : Initiation à l'outil EMU8086

Le but de ce TP est de familiariser avec l'outil EMU8086, c'est un logiciel plus pratique pour réaliser des programmes en langage assembleur.

Les étapes à suivre pour écrire un programme en assembleur

- Lorsque nous démarrons l'outil EMU 8086, l'application nous propose une interface suivant : (voir fig. 1)

Si nous appuyons sur le bouton « code exemple » nous avons accès à une liste d'exemple.

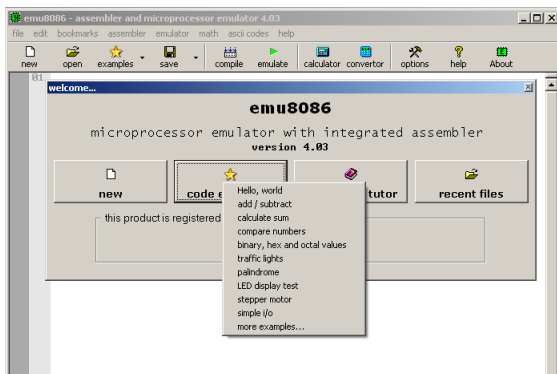


Fig. 1

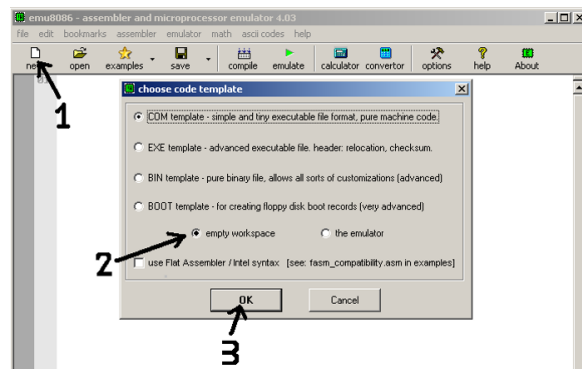


Fig. 2

- Pour une création d'édition d'un programme nous le faisons en 3 étapes, New, Empty workspace et valider (voir fig.2)

- Pour écrire et exécuter un programme en assembleur en utilise la fenêtre édition du programme (voir fig.3). un exemple de programme simple qui permet de faire une addition entre les contenus de deux registres AX et BX. Le programme est donné comme suit :

MOV AX, 5 ; transférer une donnée (05H) au registre AX.
MOV BX, 10 ; transférer une donnée (10H) au registre BX.
ADD AX, BX ; additionner les contenus des deux registres
HLT ; arrêt du programme

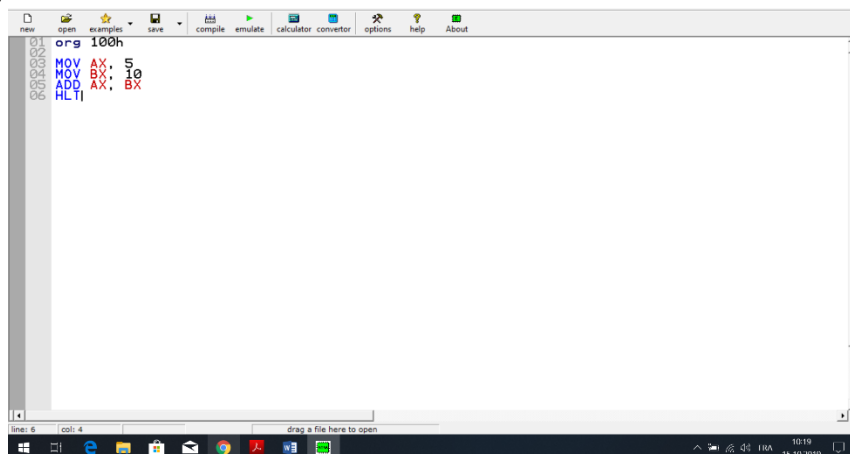


Fig. 3

- Pour exécuter le programme de la fig.3, il faut cliquer sur EMULATE. L'assembleur attribue aux instructions traduites en langage machine des adresses relatives. Pour désigner une instruction contenue dans une case de la mémoire de 1 Mo, nous avons besoin d'une adresse codée sur 20 bits (cela dépasse les capacités des registres internes du 80x86). Pour contourner ce problème, l'adresse est codée avec 2 registres :
 - a) le registre CS (Code Segment) contient l'adresse de la base du segment de code.
 - b) le registre IP (Instruction Pointer) contient une adresse relative à cette base (nous parlons de déplacement ou d'offset).
 - c) l'adresse absolue (ou adresse effective) est obtenue en utilisant le couple d'un segment (numéro du segment, déplacement) soit : $CS * 16_{(10)} + IP$. (voir fig.4)

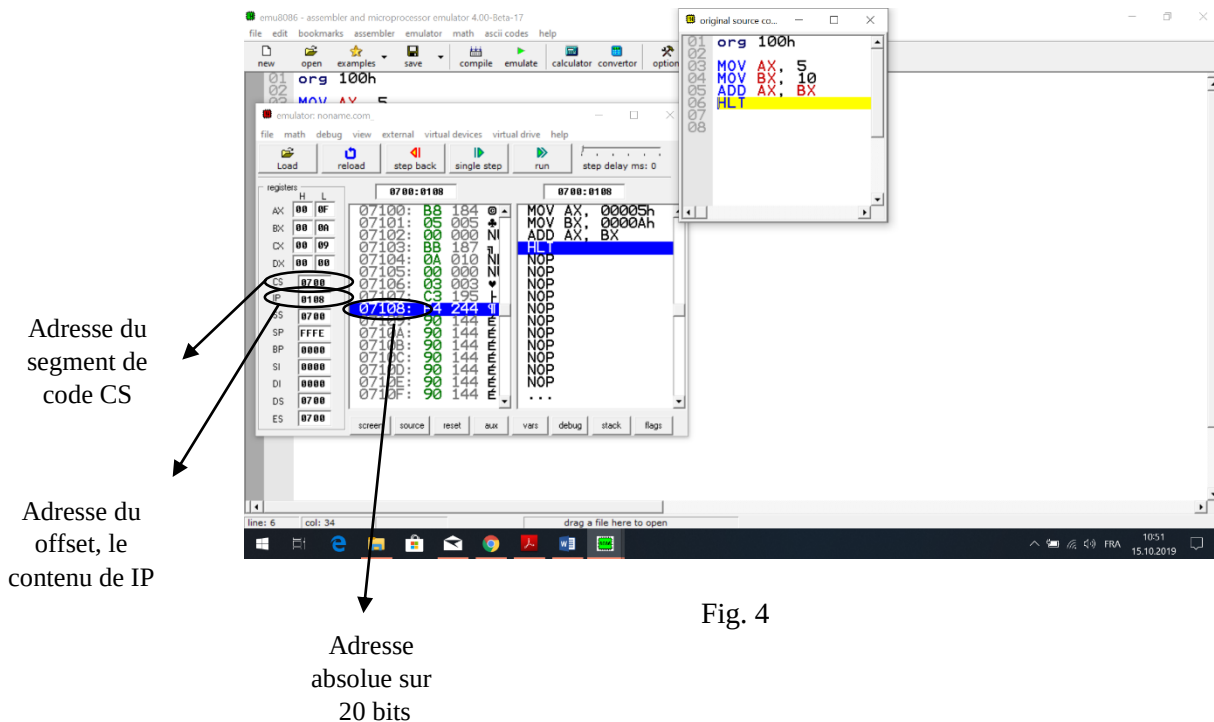


Fig. 4