

TP N°1

Familiarisation avec l'émulateur Emu8086 et instructions de transfert de données.

Objectifs du TP :

- Se familiariser avec l'émulateur EMU8086.
- Maîtriser les instructions de transfert de données et les calculs d'adresses.
- Maîtriser la notion de segmentation de la mémoire et le mode d'adressage.

I. Emulateur EMU8086

Ce programme est très utile pour ceux qui commencent à apprendre le langage Assembleur. Il compile le code source et l'exécute pas à pas dans un émulateur. Un émulateur est un programme permettant d'exécuter un logiciel (programme) sur une plate-forme auquel il n'est pas destiné.

Il est plus facile de manipuler des interfaces graphiques sur un PC que de travailler sur du matériel. Au lancement de l'émulateur, une fenêtre contenant la boîte de dialogue suivante s'affiche (figure 1). Cette dernière permet de choisir entre plusieurs options et de créer un nouveau fichier.

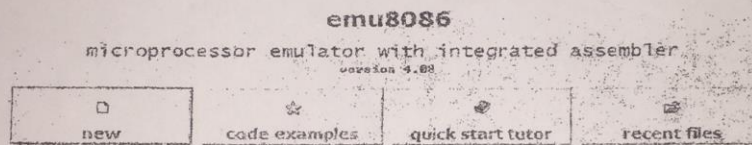


Figure 1 : Boîte de dialogue de bienvenue.

En cliquant sur le bouton *New*, la boîte de dialogue de la figure 2 s'ouvre. Cette fenêtre permet de choisir le type de fichier à créer. Dans ce premier TP, on utilisera des fichiers binaires (d'extension *.bin*). On sélectionne le format *BIN* et on clique sur *OK*.

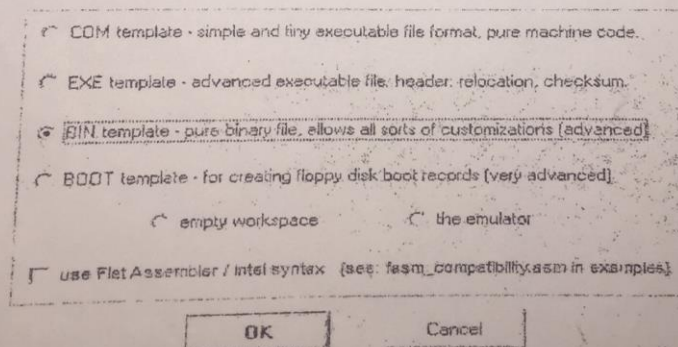


Figure 2 : Fenêtre de choix du type de fichier.

isir son programme.

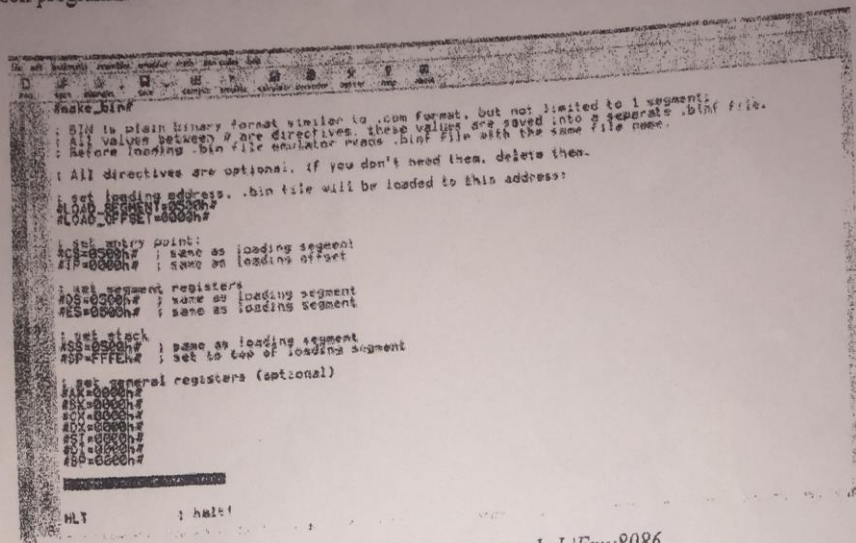


Figure 3 : Editeur de programme de L'Emu8086.

Une fois la saisie du programme est terminée, on peut soit le compiler ou l'émuler en cliquant sur le bouton correspondant dans la barre d'outils. On cliquant sur le bouton **Emulate**, et après la correction des erreurs syntaxiques, les deux fenêtres suivantes s'ouvrent (figure 4).

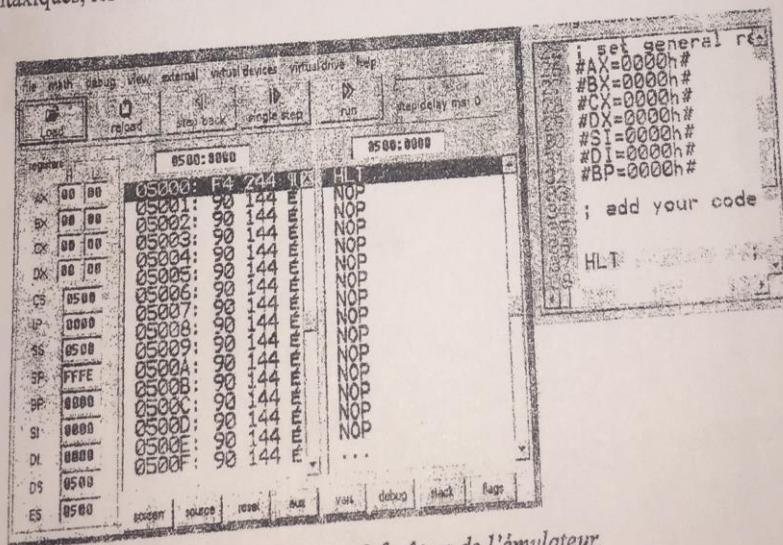


Figure 4 : Les 2 fenêtres de l'émulateur.

La petite fenêtre contient le code source du programme et la grande fenêtre représente l'émulateur. Ce dernier permet d'exécuter le programme tout entier ou instruction par instruction. Il permet aussi de visualiser (entre autres) les contenus de tous les registres du microprocesseur ainsi que les différents segments de la mémoire et autre composants du système.

II. Instructions de transfert de données et les modes d'adressage

C'est principalement l'instruction *MOV*, elle permet d'écrire une valeur directement dans l'un des registres du processeur ou dans un emplacement en mémoire. Elle permet aussi le transfert de données entre les registres ou entre registre et mémoire. Le transfert mémoire – mémoire n'étant pas possible.

La syntaxe de l'instruction *MOV* est comme suit :

MOV destination, source

L'instruction *MOV* réalise le transfert du contenu de *la source* vers *la destination*. Plusieurs modes d'adressage sont possibles.

1. *Mode immédiat* : utilisé pour charger (*écrire*) une valeur dans un registre. La donnée est spécifiée immédiatement après l'instruction.

Exemples :

MOV SI,1240h

MOV AH,10h

2. *Mode registre* : utile lors du transfert du contenu d'un registre vers un autre.

Exemple :

MOV AX,BX

3. *Mode direct* : permet d'écrire une valeur directement à une adresse effective donnée, ou de transférer le contenu d'un registre vers un emplacement mémoire ou vice versa. Un opérande est donné en mode direct s'il est désigné par son adresse effective dans la mémoire.

Exemples :

MOV [100H],24h

MOV AH,[150h]

MOV [200h],AL

4. *Mode indirect basé* : permet les mêmes opérations de transfert que le mode direct, mais dans ce cas, l'adresse effective est le contenu d'un registre de base (*BX* ou *BP*) avec éventuellement un déplacement constant.

Exemples :

MOV [BX],64h

MOV [BP + 22H],56h

MOV AL,[BX]

5. *Mode indirect indexé* : L'adresse effective est fournie par l'un des registres d'index *SI* ou *DI* avec éventuellement un déplacement constant.

Exemples :

MOV [SI],33h

MOV [DI + 82H],5h

MOV AL,[SI]

6. *Le mode indirect basé - indexé* : les modes indexé et basé sont combinés. L'adresse logique est donnée par *BX/BP* et *SI/DI*.

Exemple :

MOV AL,2[BX + SI]

MOV [BP + DI],AH

III. Exercice

- Donner une description détaillée de l'émulateur utilisé.
- Saisir le programme ci-dessous dans l'éditeur de l'émulateur 8086.
- 1. Indiquer le mode d'adressage pour chaque instruction. *2*
- 2. Commenter chaque instruction *✓*
- 3. Donner le résultat obtenu après l'exécution de chaque instruction (Single Step). *✓*
- 4. Dans le cas de l'adressage de la mémoire, donner l'adresse logique et calculer l'adresse physique.
- 5. Vérifier l'emplacement mémoire en cas d'un adressage mémoire par View-Memory.

MOV AX, 600H
MOV DS, AX
MOV AX, 700H
MOV SS, AX
MOV AX, 800H
MOV ES, AX
MOV AX, 1234H
MOV BX, AX
MOV CX, BX
MOV DL, CH
MOV [100H], DL
MOV [102H], AX
MOV CH, [102H]
MOV BX, 104H
MOV [BX], AL
MOV BP, 104H
MOV [BP], DL
MOV DH, [BX]+4

MOV DH, [BP]+4
MOV DI, 3H
MOV [DI], AX
MOV AX, 100H
MOV DS, AX
MOV DH, 78H
MOV [100H], DH
MOV [102H], CX
MOV BX, 100H
MOV BP, 100H
MOV DL, [BX]
MOV DL, [BP]
MOV BX, [BX]+2
MOV DH, [BP]+2
MOV SI, 1
MOV DL, [BX][SI]
MOV DL, [BP][SI]
HLT

AL AL
06 00
IP