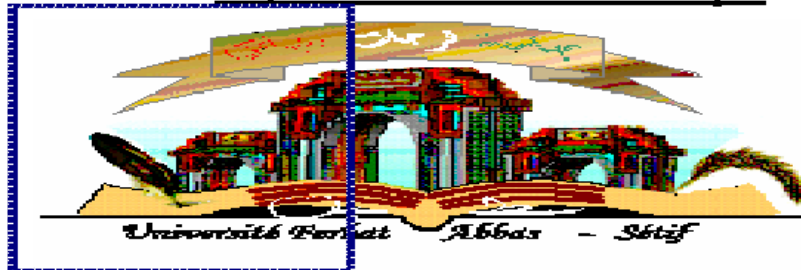


République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche
scientifique
Université Ferhat Abbas –Sétif-

Département électrotechnique



TEC 71

mécro _processeurs

TP réalisé par :

- Itouchene hichem.
- Issaadi idris.

❖ **Master 1**commande électrique

Année universitaire : 2017/2018



Introduction

EMU 8086 combine un éditeur avancé source, assembleur, désassembleur, émulateur logiciel (vertical pc) avec le déboguer et didacticiel et étape par étape.

Ce logiciel est extrêmement utile pour ceux qui commencent Just à étudier le langage d'assemblage. Il compile le code source et l'exécute à l'étape de l'émulateur par étape.

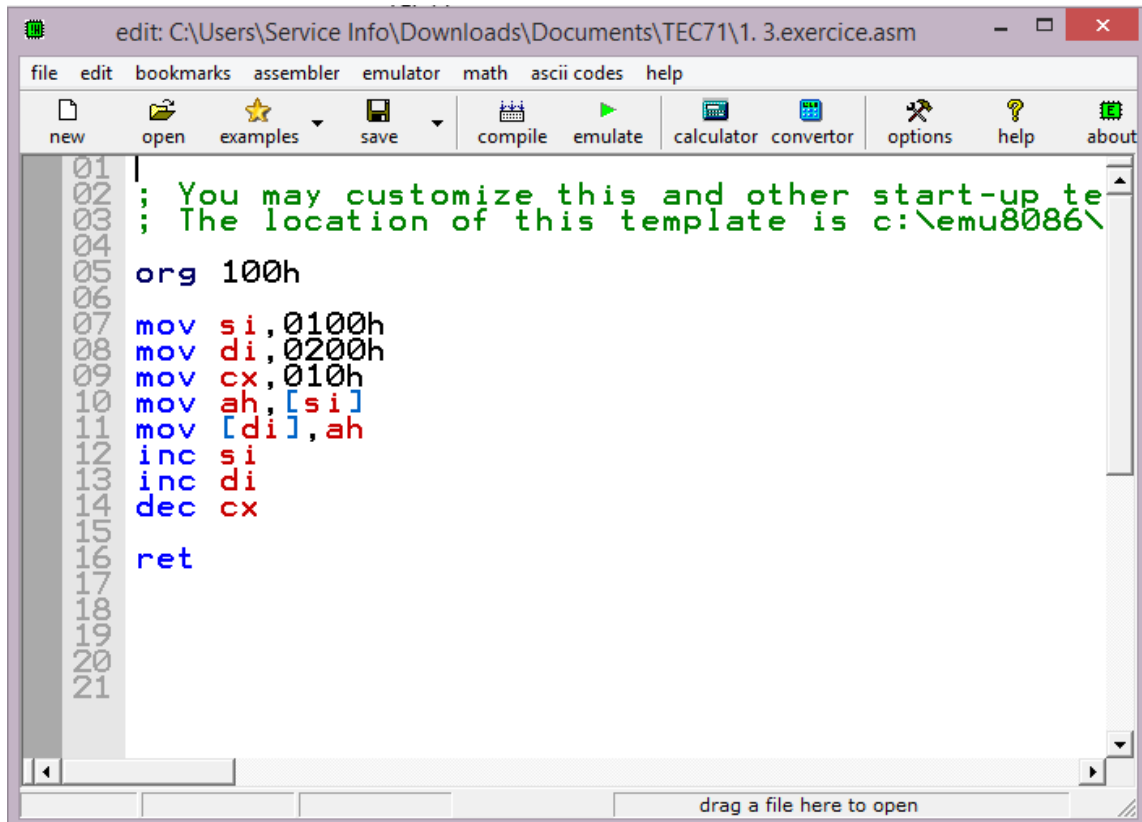
L'interface visuelle est très facile à travailler avec. Vous pouvez regarder les registres, les drapeaux et la mémoire pendant que votre programme s'exécute.

EMU 8086 a une syntaxe beaucoup plus facile que l'un des principaux assembleurs. Mais elle génère toujours un programme qui peut être exécuté sur un ordinateur qui exécute le code 8086 machine, une excellente combinaison pour les débutants.

TPN[®]1

-exercice 01 :

On utilise le **EMU 8086** pour écrire le programme suivant à partir de Cs :0100.



The screenshot shows a window titled "edit: C:\Users\Service Info\Downloads\Documents\TEC71\1. 3.exercice.asm". The menu bar includes "file", "edit", "bookmarks", "assembler", "emulator", "math", "ascii codes", and "help". The toolbar contains icons for "new", "open", "examples", "save", "compile", "emulate", "calculator", "convertor", "options", "help", and "about". The assembly code is as follows:

```
01 |
02 | ; You may customize this and other start-up te
03 | ; The location of this template is c:\emu8086\
04 |
05 | org 100h
06 |
07 | mov si,0100h
08 | mov di,0200h
09 | mov cx,010h
10 | mov ah,[si]
11 | mov [di],ah
12 | inc si
13 | inc di
14 | dec cx
15 |
16 | ret
17 |
18 |
19 |
20 |
21 |
```

At the bottom of the window, there is a text box that says "drag a file here to open".

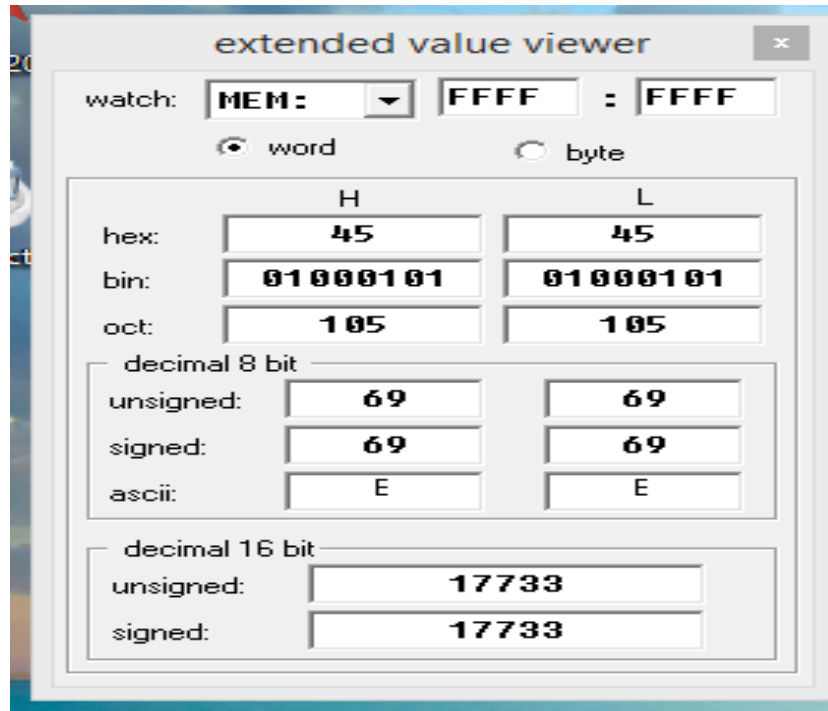
* ce programme prendre : **01 octets (8bits)**

B// le code de le machine : **DEC CX : 0710F**

C // le déplacement d'adresse pour **MOV AH ,[SI]** :.....

07100:	BE	190	¥	MOV SI, 00100h
07101:	00	000	NI	MOV DI, 00200h
07102:	01	001	☺	MOV CX, 00010h
07103:	BF	191	7	MOV AH, [SI]
07104:	00	000	NI	MOV [DI], AH
07105:	02	002	☹	INC SI
07106:	B9	185		INC DI
07107:	10	016	►	DEC CX
07108:	00	000	NI	RET
07109:	8A	138	è	NOP
0710A:	24	036	\$	NOP
0710B:	88	136	è	NOP
0710C:	25	037	%	NOP
0710D:	46	070	F	NOP
0710E:	47	071	G	NOP
0710F:	49	073	I	...

Remplissage d'emplacement de mémoire à partir de DS :0200 avec 45H



les valeurs des registres AX,CX,SI,DI,IP :

	AX	CX	SI	DI	IP
MOV SI,0100h	00 00	00 11	0000	0000	0100
MOV DI,0200h	00 00	00 11	0100	0000	0103
MOVCX,0010 h	00 00	00 11	0100	0200	0106
MOV AH,[SI]	00 00	00 10	0100	0200	0109

MOV[DI],A H	BE 00	00 10	0100	0200	010B
INC SI	BE 00	00 10	0100	0200	010D
INC DI	BE 00	00 10	0101	0200	010^E
DEC CX	BE 00	00 10	0101	0201	010F

*On exécute le programme complet par la command run :

Les valeurs finales des registres :

AX:BE 00

CX:00 0F

SI:0101

DI:0201

TPN°2

Travail à faire

```

edit: C:\Users\Service Info\Downloads\Documents\TEC71\tp_02_taf.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
002 ; You may customize this and other start-up templates;
003 ; The location of this template is c:\emu8086\inc\0_com_template.txt
004 org 100h
005 ;1
006 mov al,6bh
007 shr al,1
008 mov cl,3
009 shl al,cl
010 ;2
011 mov ax,0aaaah
012 mov cl,8
013 shl ax,cl
014 ;03
015 mov al,8ch
016 mov cl,3
017 sar al,cl
018 ;04
019 mov di,1000h
020 mov [di],0aah
021 mov cl,3
022 shl byte ptr [di],cl
023 ;5
024 mov al,6bh
025 ror al,1
026 rol al,3
027 ;06
028 stc
029 mov al,6bh
030 rcr al,3
031 ;07
032 clc
033 mov di,2000h
034 mov [di],0aah
035 mov cl,1
036 rcl byte ptr [di],cl
037
038 ret
line: 1 col: 2 drag a file here to open

```

Remplissage de tableau :

STATEMENT	SOURCE		DESTINATION			Status flags				
	Register /mémoire	contenets	Regist r/mémoire	Contets bifo re executio n	Contente after execution	A F	P F	S F	Z F	C F
SHR AL,1	1	1	R	6B	35	0	0	0	0	0
SHL AL,3	3	3	R	35	A8	0	1	0	0	1
						0	0	1	0	1
SHL AX,CL	8	8	R	AAAA	AA00	0	1	0	1	0
SAR AL,CL	3	3	R	AA8C	AAF1	0	1	1	0	0
						0	0	1	0	1
SHL BYTE PTR [DI],CL	3	3	MEM	AAF1	AAF1	0	0	1	0	1
						0	1	0	0	1
ROR AL,1	1	1	R	AA6B	AAB5	0	1	0	0	1
ROL AL,3						0	1	0	0	1

	3	3	R	AAD6	AAAD	0	1	0	0	0
						0	1	0	0	1
RCR AL,3	3	3	R	AADA	AAED	0	1	0	0	1
						0	1	0	0	0
RCL BYTE PTR [di],CL	1	1	MEM	AAED	AAED	0	1	0	0	0
						0	1	0	0	1

Exercice 02

```

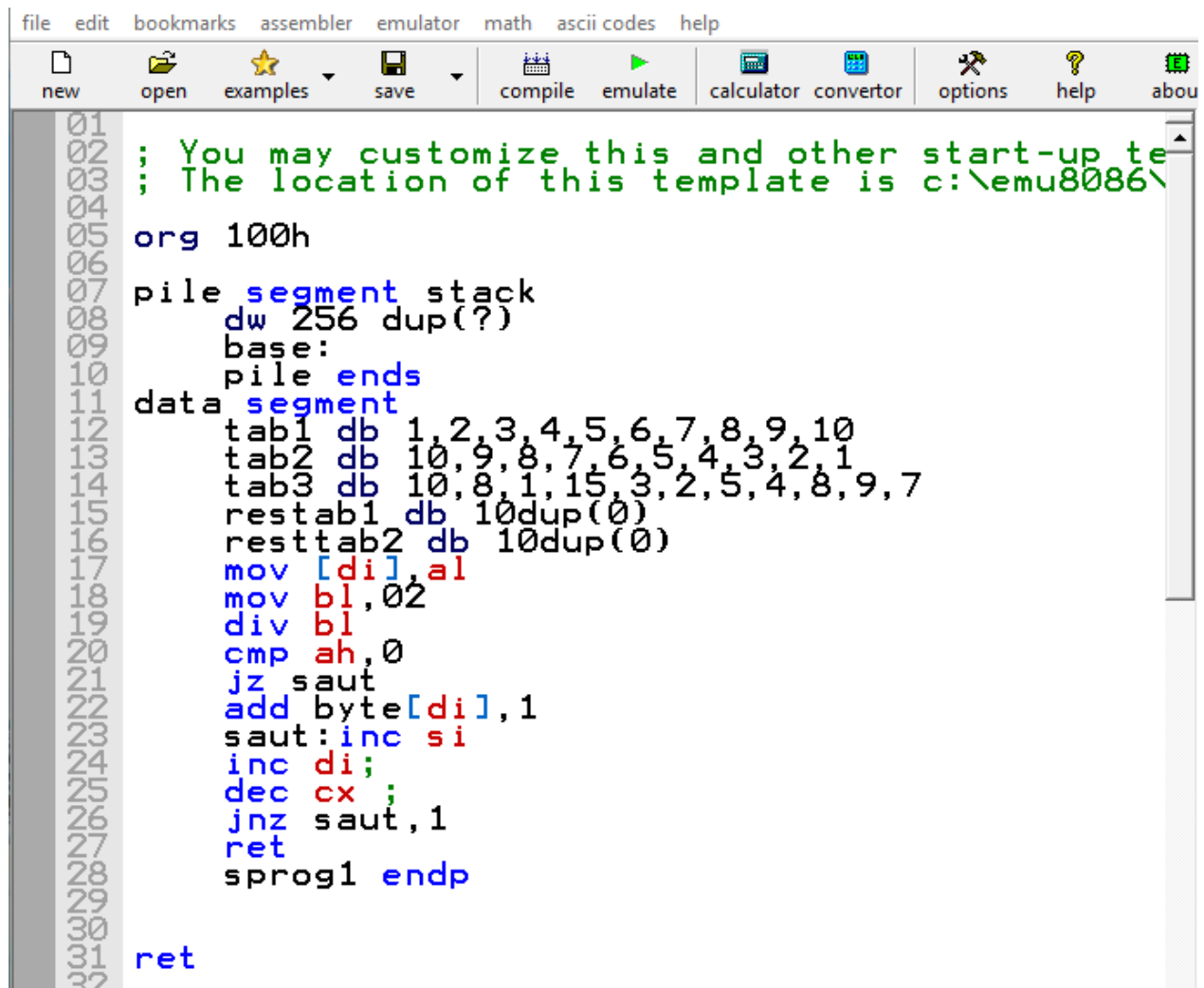
01 |
02 | ; You may customize this and other start-up templates;
03 | ; The location of this template is c:\emu8086\inc\0_com_te
04 |
05 | pile segment stack
06 |     dw 256 dup(?)
07 |     base:
08 | pile ends
09 | data segment
10 |     n1 dw 5
11 |     n2 dw 3
12 |     n3 dw 3 dup(?)
13 |     n4 db ?
14 | data ends
15 | code segment
16 |     assume cs:code,ds:data,ss:pile
17 |     main:
18 |     mov ax,data;initialisation du segment de donnees
19 |     mov ds,ax
20 |     mov ax,pile
21 |     mov ss,ax ; initialisation du segment de pile
22 |     mov sp,base ; initialise la pile a vide
23 |     mov ax,n1
24 |     cmp ax,n2
25 |     JE etiq
26 |     mov n4,0
27 |     jmp fin
28 |     etiq:
29 |     mov n4,1
30 |     fin:
31 |     mov ah,4ch ;les 2 lignes necessaires pour la...
32 |     int 21h ;....fin de programme
33 |     code ends
34 |     main
35 | end
36 | ret
37 |

```

- *La valeur de N4 à la fin de l'exécution du programme sera 1
- * les valeurs du N1 N2 N3 ne son pas modifiées
- * la pile est composée de 0.5 octets (04 bits).
- * le nombres des variables créées dans le segment de données est 4 variables.
- * la taille de segment de donnée = 04 bits

TPN°3

Exercice3

The image shows a screenshot of a software application titled "TPN°3". The interface includes a menu bar with options: file, edit, bookmarks, assembler, emulator, math, ascii codes, help. Below the menu bar is a toolbar with icons for new, open, examples, save, compile, emulate, calculator, convertor, options, help, and about. The main window displays assembly code with line numbers on the left. The code includes comments in green, and various assembly instructions in blue and red. The code defines a stack segment, a data segment with tables, and a procedure named sprog1.

```
01  
02 ; You may customize this and other start-up te  
03 ; The location of this template is c:\emu8086\  
04  
05 org 100h  
06  
07 pile segment stack  
08     dw 256 dup(?)  
09     base:  
10     pile ends  
11 data segment  
12     tab1 db 1,2,3,4,5,6,7,8,9,10  
13     tab2 db 10,9,8,7,6,5,4,3,2,1  
14     tab3 db 10,8,1,15,3,2,5,4,8,9,7  
15     restab1 db 10dup(0)  
16     restab2 db 10dup(0)  
17     mov [di],al  
18     mov bl,02  
19     div bl  
20     cmp ah,0  
21     jz saut  
22     add byte[di],1  
23     saut:inc si  
24     inc di;  
25     dec cx;  
26     jnz saut,1  
27     ret  
28     sprog1 endp  
29  
30  
31     ret  
32
```

TPN°4

Exercice04 :

```
01 ; You may customize this and other start-up
02 ; The location of this template is c:\emu808
03
04 org 100h
05
06 title
07 .model small;
08 .stack 100;
09
10 .data
11 .code
12 mov ax,@data;
13 mov ds,ax;
14 mov ah,00h;
15 mov al,12h;
16 int 10h;
17 mov cx,170;
18 mov dx,240;
19 mov ax,0c02h;
20 back:int 10h;
21 inc cx;
22 cmp cx,470;
23 jb back;
24 mov ah,07h;
25 int 21h;
26 mov ax,4c00h;
27 int 21h;
28 end ;
29
30
31
32
33
34
35
```

Activer Windows
Accédez aux paramètres pour activer Windows

