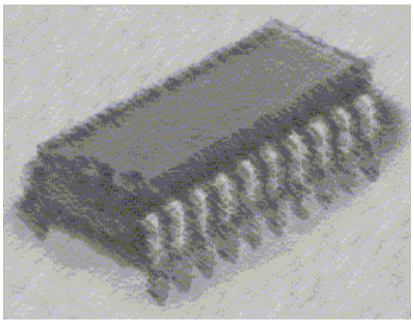

PRACTICE 2: Registers and memory positions

Objectives:

This practice helps student in the understanding of how data are stored in memory and registers and the different way of memory accessing.

Programs used:

Microsoft Assembler 5.1 will be used to assemble (MASM), link (LINK) and execute (CODE VIEW) assembly programs.

PRACTICE 2 ACTIVITIES

Some of the next activities may have coding mistakes in order of learning how errors are shown and corrected.

Exercise #	Exercise

1	Write, assemble, link and execute next program code:
	dosseg
	.model small
	.stack 100h
	.data
	   Terminar EQU 4Ch
	.code
	Inicio:
	   mov ax, @data
	   mov ds, ax
	
	   mov al, 10h
	   mov cl, 4
	   shl al, cl
	
	   mov ah, 4Ch
	   int 21h
	end Inicio

2	How did AX register change? Change shl al, cl instruction by shr al, cl instruction and do activity one again. Are there any changes in program execution?
3	Write, assemble, link and execute next program code: dosseg .model small .stack 100h .data

Exercise #	Exercise
	Texto DB "Please enter a maximum 8 character sentence: ",13,10,'$' MaximoMas1 DB 9 CaracteresLeidos DB 0 Cadena DB 9 DUP (0) Otralinea DB 13,10,'$' .code Inicio: mov ax, @data mov ds, ax mov ah, 9 lea dx, Texto int 21h mov ah, 0Ah lea dx, MaximoMas1 int 21h mov ah, 9 lea dx, Otralinea int 21h mov cl, MaximoMas1 xor SI, SI mov ah, 2 bucle: mov dl, Cadena[SI] int 21h inc SI loop bucle mov ah, 4Ch int 21h END Inicio
4	Modify above program to allow storing 20 character sentences.
5	Write next code inside the loop SUB DL, 20h and after mov dl, Cadena[SI] instruction ¿What will the program do?
6	Write, assemble, link and execute next program code: dosseg .model small .stack 100h .data

Exercise #	Exercise
	Dir1 DB 10h, 20h, 30h, 40h Dir2 DB 50h, 60h, 70h, 80h ..Resul DB ?, ?, ?, ? .code Inicio: mov ax, @data mov ds, ax mov cx, 4 mov bx, 0 clc Otro: mov al, Dir1[bx] adc al, Dir2[bx] mov Resul[bx], al inc bx loop Otro mov ah, 4Ch int 21h END Inicio
7	What did the above program do? Display Dir1, Dir2 and Resul variable (E.g. In Code View order line **> Wb Dir1 l 4**) Change program data using the Code View (E.g. In Code View order line >**Eb** and hit enter)
8	Modify above program to add 16 bits length numbers
9	Write, assemble, link and execute next program code (please, mind the BL register value when program finished): dosseg .model small. .stack 100h .data Texto DB ‘Please, enter a hexadecimal number of two digits between 0 and 9$’ .code Inicio: mov ax, @data mov ds, ax mov ah, 9 lea dx, Texto

Exercise #	Exercise
	int 21h xor bl, bl mov ah, 1 int 21h mov cl, 4 mov bl, al sub bl, 30h shl bl, cl int 21h sub al, 30h add bl, al mov ah, 4Ch int 21h END Inicio
10	Modify about activity to read 4 hexadecimal digits number
11	Write, assemble, link and execute next program code: dosseg .model small .stack 100h .data Numero DB 17h .code Inicio: mov ax, @data mov ds, ax mov bh, Number mov cl, 4 shr bh, cl mov ah, 2 mov dl, bh add dl, 30h int 21h mov dl, Number and dl, 0Fh add dl, 30h int 21h mov ah, 4Ch int 21h END Inicio

Exercise #	Exercise
12	What does the above program do?
13	Change number 17h by 34h and 0Fh. How does the program execution change?

PRACTICE 2:

Registers and memory positions

Create an assembly program that ask for a 16 bits number and:

- Store in varc2 the complement to 2 of introduced number. Store the negative value if positive number is entered and vice versa.
- Change the sign of the introduced number and store in varex. The representation system used is excess 2^{16-1}.