**Daffodil International University**

**Department of Electrical and Electronic Engineering**

**EEE 324: Microprocessors and Interfacing Laboratory**

**Experiment No: 03**

**NAME OF THE EXPERIMENT: Single Character, String and Number Printing to Display using Assembly Language**

## Theory:

CPU communicates with peripherals through I/O registers called I/O ports. There are two instructions, IN and OUT, that access the port directly. But main inconvenience of using these instructions is that port addresses vary among computer models. So programmers usually use I/O service routines which are serviced through an interrupt.

To invoke a BIOS or DOS routine INT instruction is used.

INT interrupt\_number

INT 21h may be used to invoke a large number of DOS functions. A particular function is requested by placing a function number in the AH register.

For AH = 1

Waits for a key pressing from keyboard and

it stores ASCII value of the key to AL

it stores zero to AL if non-character key is pressed

For AH = 2

It get the ASCII value from DL register and prints the character to display at current cursor position.

After the character displayed, the cursor advances to next position.

AL gets the ASCII code after execution.

If DL contains ASCII of any control character controlled functions are performed.

 ASCII Symbol function

 7 BEL beep

 8 BS backspace

 9 HT Horizontal Tab

10 LF Line Feed (New Line)

 13 CR Carriage return /Start of current line (Enter)

For AH = 9

It start printing a string starting from the offset address stored in DX and continue printing characters until it gets a ‘$’ character.

For AH = 4CH or 76

Exit to dos

**The LEA instruction:**

LEA destination, source

LEA(load effective address) puts a copy of source offset address to destination

LEA DX,MSG puts the offset address of variable MSG into DX.

##  PROCEDURE

1. In a text editor (Notepad) write the following code. You can omit comments. Don’t write line number on editor window.
	* 1. TITLE ABC:ABC
		2. .MODEL SMALL
		3. .STACK 100H
		4. .DATA
		5. A DB 65
		6. B DB 'Bangladesh$'
		7. C DW 68
		8. .CODE
		9. MAIN PROC
		10. MOV AX,@DATA ;initialize DS
		11. MOV DS,AX
		12. MOV DL,A
		13. MOV AH,2
		14. INT 21H ;print character 'A' to display
		15. LEA DX,B
		16. MOV AH,9
		17. INT 21H ;print String 'Bangladesh' to display
		18. MOV AX,C
		19. MOV BL,10
		20. DIV BL ;Divide number by 10
		21. MOV DX,AX ;Now quotient is in DL and reminder in DH
		22. MOV AH,2
		23. ADD DL,48 ;to get ASCII code of the decimal character
		24. INT 21H
		25. MOV DL,DH
		26. ADD DL,48
		27. INT 21H
		28. MOV AH,4CH ;exit to DOS
		29. INT 21H
		30. MAIN ENDP
		31. END MAIN
2. Save the program as an asm file. i.e. filename should be like exp1.asm (select “all files” in “save as type”).
3. Save the program in a folder where MASM.EXE and LINK.EXE are present.
4. Write ‘cmd’ in RUN window and press OK/Enter
5. Browse to folder containing your file. You can use following DOS commands to do so.
	1. To change drive write C: or D: or E: and Enter to go to any drive you want.
	2. You will view the contents of a directory by using the dir command. The dir command stands for "directory." All the names that have <DIR> beside them are directories. Other file names also appear on the list. To view the contents of a directory one screen at a time type dir /p.
	3. To change current directories, you will use the cd command. The cd command stands for "change directory."

cd mda ↵ change current directory to mda which exists in current directory

cd .. ↵ change current directories to parent directory

cd \ ↵ change current directories to root directory of that drive

* 1. To make a subdirectory under current directory use the command md (make directory)

md mda ↵ creates a directory named ‘mda’ in the current directory

* 1. To delete a subdirectory under current directory use the command rd(remove directory)

rd mda ↵ removes the ‘mda’ directory from current directory

* 1. To rename a file, you will use the ren command. Syntax of this command is

ren oldname newname

To rename the readme.txt file to peach.asm, type the following command

ren readme.txt peach.asm ↵

* 1. To delete a file, you will use the del command. The del command stands for "delete."

To delete the pear.com file type command

del pear.com ↵

* 1. To copy file use copy command

copy readme.txt d:\mda↵ copies redme.txt file to mda directory

move readme.txt d:\mda↵ moves redme.txt file to mda directory

* 1. To format a drive or disk use the command format

format c: ↵ will format c drive

you may need to confirm by pressing y↵

1. Run the file MASM.EXE with your asm file name after it

MASM.EXE EXP1.ASM↵ or

MASM EXP1↵

You will see

C:\mda>masm exp1

Microsoft (R) Macro Assembler Version 5.10

Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [exp1.OBJ]:

1. Press enter to create exp1.OBJ file then

Source listing [NUL.LST]:

Press enter if list file is not needed

Press any name and enter if list file is needed

1. then you will see

Cross-reference [NUL.CRF]:

Press enter if cross reference file is not needed

Press any name and enter if cross reference file is needed

1. Then you see

48528 + 446379 Bytes symbol space free

 0 Warning Errors

 0 Severe Errors

1. If there is warning or error you will see what error happened in which line. Like, due to writing the instruction MOV AH,DX the warning arrives

Exp1.ASM(23): warning A4031: Operand types must match

48528 + 446379 Bytes symbol space free

 1 Warning Errors

 0 Severe Errors

Description

File name

Line no

Error Code

1. Fix error in the source file and run MASM again if there is an error.
2. Run the file LINK.EXE with your created object file name after it

LINK.EXE EXP1.OBJ↵ or

LINK EXP1↵

C:\mda>link exp1

Microsoft (R) Overlay Linker Version 3.60

Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [exp1.EXE]:

1. Press enter to create exp1.exe file then

List File [NUL.MAP]:

Press enter if .map file is not needed

Press any name and enter if list file is needed

1. Again,

Libraries [NUL.LIB]:

Press enter if list file is not needed or,

Press name of existing library file and enter to include that.

1. Run the created exp3.exe file

exp1.exe↵ or exp1↵

C:\asm>exp1

ABangladesh68

1. You see the output ‘ABangladesh68’

## Program Analysis:

The program first print character with ASCII code of variable A, which is 65

* + - * 1. INT 21H in line 15 checks AH and find 2
				2. So it prints ‘A’ because DL contains 65
				3. Then print the string stored in location B
				4. INT 21H in line 19 checks AH and find 9
				5. So it prints String starting from [DX],
				6. LEA DX,B stores the address of B to DX earlier
				7. So it prints ‘Bangladesh’ and stops when it gets a ‘$’ character
				8. Then print the decimal value of number stored in C.
				9. To divide 68 by 10 68 is stores in AX and 10 is stored in BL
				10. After division Quotient 6 is stored in AL and Reminder 8 in AH
				11. We move AH and AL to DH and DL, and store 2 in AH
				12. Now DL has 6 but ASCII code of 6 is 48+6=54 so we add 48 with DL
				13. INT 21H in line 28 checks AH and find 2
				14. So it prints 6 as DL contains 54
				15. Then reminder 8 is moved from DH to DL and ADD 48 with it
				16. INT 21H in line 31 checks AH and find 2
				17. So it prints 8 as DL contains 56

Modify the program to print a space between A and Bangladesh and print the number 68 in a new line. You have to know ASCII code of some characters. ASCII code of Space is 32 and new line is 10.

## Questions:

1. Write a code which will print “ F16?”
2. Why do we need to use a ‘$’ sign after Bangladesh? What happens if we do not use the sign?
3. Why do we need to use the instruction MOV DX, AX in line 25 ?
4. If you want to print the number of variable C in Octal/Binary format what would you do?