Faculty Member:_____ **Dated:** _____

Semester:_____ **Section:** _____

Department of Electrical Engineering and Computer Science

EE-222 Microprocessor Systems

Lab2: Saving Values in Registers and Debug them using TASM Debugger

Name	Reg. No.	Report Marks / 10	Viva Marks / 5	Total/15

EXPERIMENT 02

SAVING VALUES IN REGISTERS AND DEBUG THEM USING TASM DEBUGGER

OBJECTIVES:

1. Getting introduced to assembly language
2. Learning some basic commands
3. Introduction to the syntax of assembly language programming
4. Learning the use of turbo assembler (TASM)

EQUIPMENT:

SOFTWARE:

- Turbo assembler (TASM)

DISCUSSION:

Before starting coding in assembly we should get familiarized with some basic coding parameters, assembly language syntax and some basics of microprocessors.

Introduction to Registers:

There are four type of registers in microprocessors:

1. AX
2. BX
3. CX
4. DX

AX, BX are mainly used for arithmetic operations and saving address. CX is used for saving the values for counts which is used in executing loop instructions and DX is mainly used for I/O operations.

PROGRAM STRUCTURE:

MEMORY MODELS:

The size of code and data a program can have is determined by specifying memory model using the .MODEL directive. The models used are SMALL, LARGE, and HUGE but the appropriate one is .SMALL. The model directive should come before any segment definition.

DATA SEGMENT:

The data segment is used for all the variables definitions. We use **.DATA** directive followed by variable and constant declaration.

STACK SEGMENT:

The purpose of stack segment is to set aside a block of memory to store the stack. The declaration syntax is:

.stack size

If we write:

.stack 100h

100 bytes would be reserved for stack. If size omitted 1KB is the default size.

CODE SEGMENT:

Code segment contains all the programming instructions. The declaration syntax is:

```
.model small

.stack 100h

.data

        ; Data definitions go here

.code

.startup

        ; Instructions go here

.end
```

The last line here should be END directive followed by the exit.

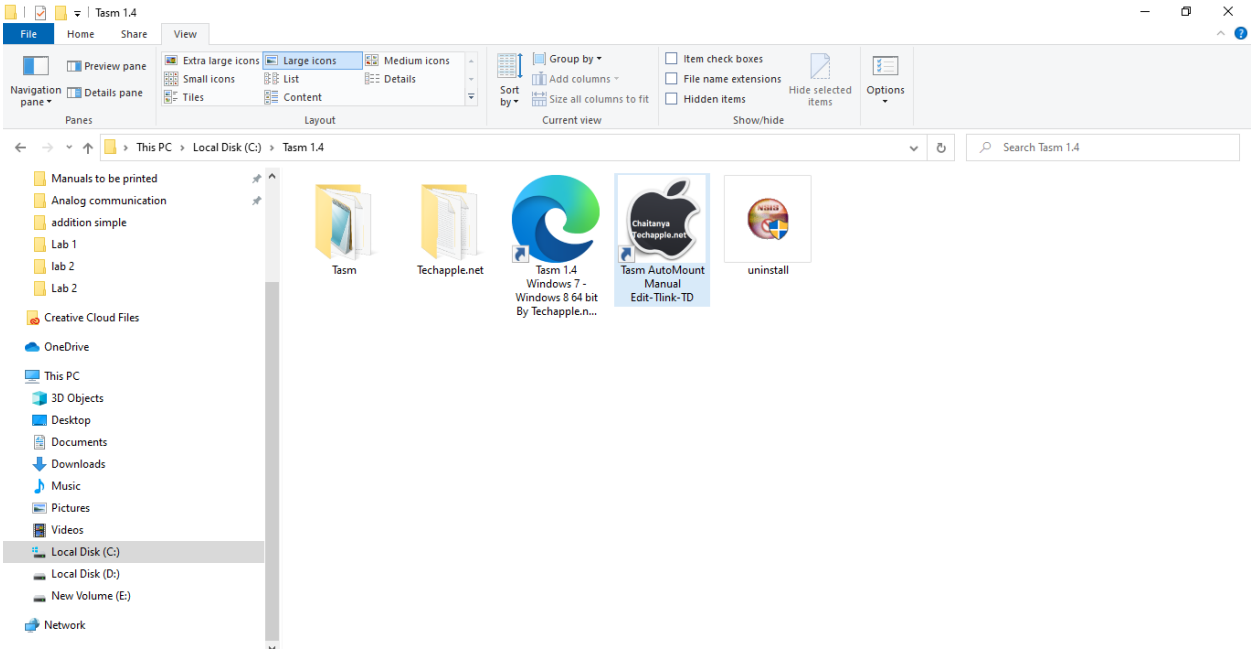
INT 21h:

INT 21h may be used to invoke a large number of DOS functions; a particular function is requested by placing a function number in AH register and invoking INT 21h.here we are interested in following functions.

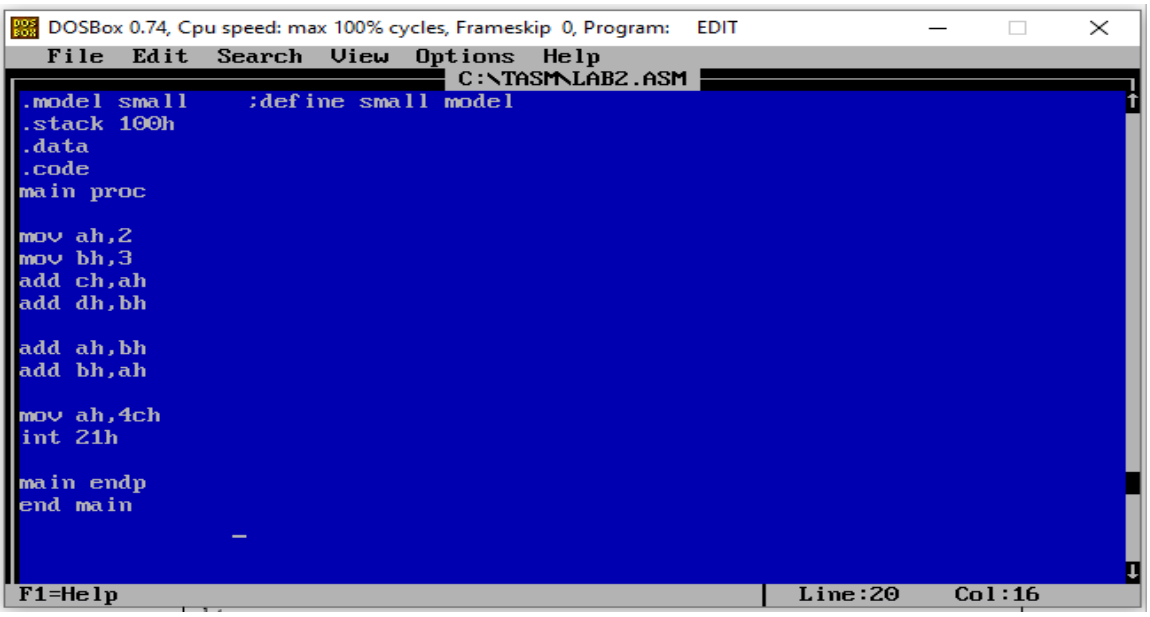
FUNCTION NUMBER	FUNCTIONS
1	Single key input
2	Single character output
9	Character string output

GETTING STARTED WITH TASM:-

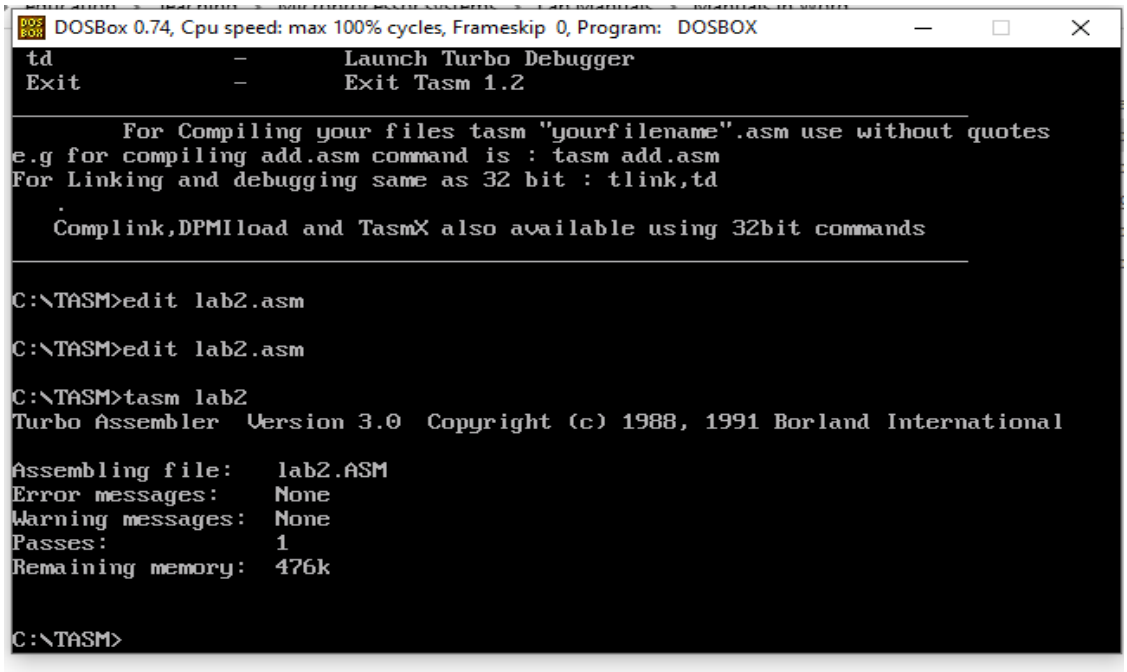
- Open the command prompt and switch to the directory where TASM has been installed
- Go to Tasm 1.4 i.e. the address of the directory may look like C:\Tasm 1.4
- You can open TASM by double clicking Tasm Auto mount icon.
- Each step with screenshot is given to make each step easily understandable



- Enter the edit command, that will open the TASM file editor



- Write your code and save your program in the bin directory of TASM, the file extension should be ".asm" as you are programming in assembly language
- Exit the editor
- Now enter "TASM ABC.ASM"



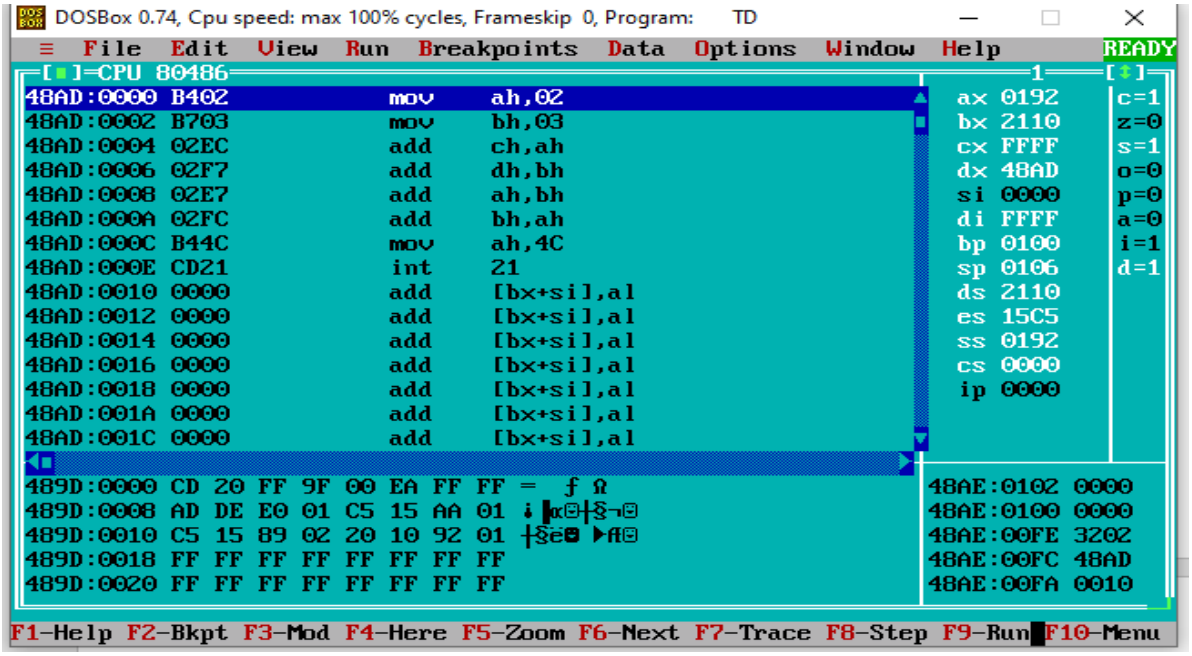
- If there is no error then enter "tlink ABC", this will generate the "exe" file against your code, by just entering the name of the file i.e. ABC now your program will be executed

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program:  DOSBOX
        For Compiling your files tasm "yourfilename".asm use without quotes
e.g for compiling add.asm command is : tasm add.asm
For Linking and debugging same as 32 bit : tlink,td
   .
   Complink,DPMIload and TasmX also available using 32bit commands
_____________________________________________________________________

C:\TASM>edit lab2.asm

C:\TASM>edit lab2.asm

C:\TASM>tasm lab2
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   lab2.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k


C:\TASM>tlink lab2
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>
```

- To view the registers use command “td ABC”, and press enter.

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program:  DOSBOX

C:\TASM>edit lab2.asm

C:\TASM>tasm lab2
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   lab2.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k


C:\TASM>tlink lab2
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>td lab2
```

- A new window will be opened.



- Press F8 key to check the execution of the code step by step.
- Follow the above steps and execute the required code.

SAMPLE CODE:-

PROGRAM DESCRIPTION:

Start with moving values in registers using Mov instruction and then using Add instruction to add the register values.

```
.model small    ;define small model
.stack 100h
.data
.code
main proc

mov ah,2
mov bh,3
add ch,ah
add dh,bh

add ah,bh
add bh,ah

mov ah,4ch
```

```
int 21h

main endp
end main
```

EXERCISES:

1. Write an assembly code to display the data of registers before the termination command than edit your code so that the data will be displayed on new line.
2. Write an assembly code for example code discussed above but now display all the values of destination register after every instruction i.e Mov, Add etc
3. Write an assembly code to user input any small alphabet than print its capital version on the screen (Hint: use ASCII table)