**Logo, company name

Description automatically generated**

**Department of IT and Computer Science**

**Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan**

**COMP-261L Computer Organization & Assembly Language Lab**

**Lab Report: 05**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 3rd**

**Submission Date: 21 Oct, 2021**

**Submitted to: Lab Engr. Rafi Ullah**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Signature**

**Lab No. 5**

Jump and Control Instructions

**Objectives:**

After completing this lab, you should be able to:

* Understand the meaning of labels.
* Understand the working and use of JMP and related instructions.
* Understand conditional and unconditional jumps
* Understand the concept of subroutines

**Tools/Software Required:**

EMU8086 Emulator

**Introduction:**

To comprehend the basic operation of Jump and Control, Loop and Call instructions. We can use these for reusability of code and loops for different purposes

**Lab Tasks:**

**Lab Task 03:** Write a program, which takes input from port address 3060h then calls a subroutine having label ‘ADDNOS’. ADDNOS subroutine adds the inputted values from the specified input port. The program takes input from port 10 times. Produce a certain delay by using a subroutine between two consecutive inputs from port. Program stores the final result as a word on memory location 0200.

**Code:**

org 100h

main proc

mov bl,10h

mov al,0h

mov dl,1h

call ADDNOS

jmp exit

ADDNOS proc near

L1:

add al,dl

dec bl

cmp bl,0

jg L1

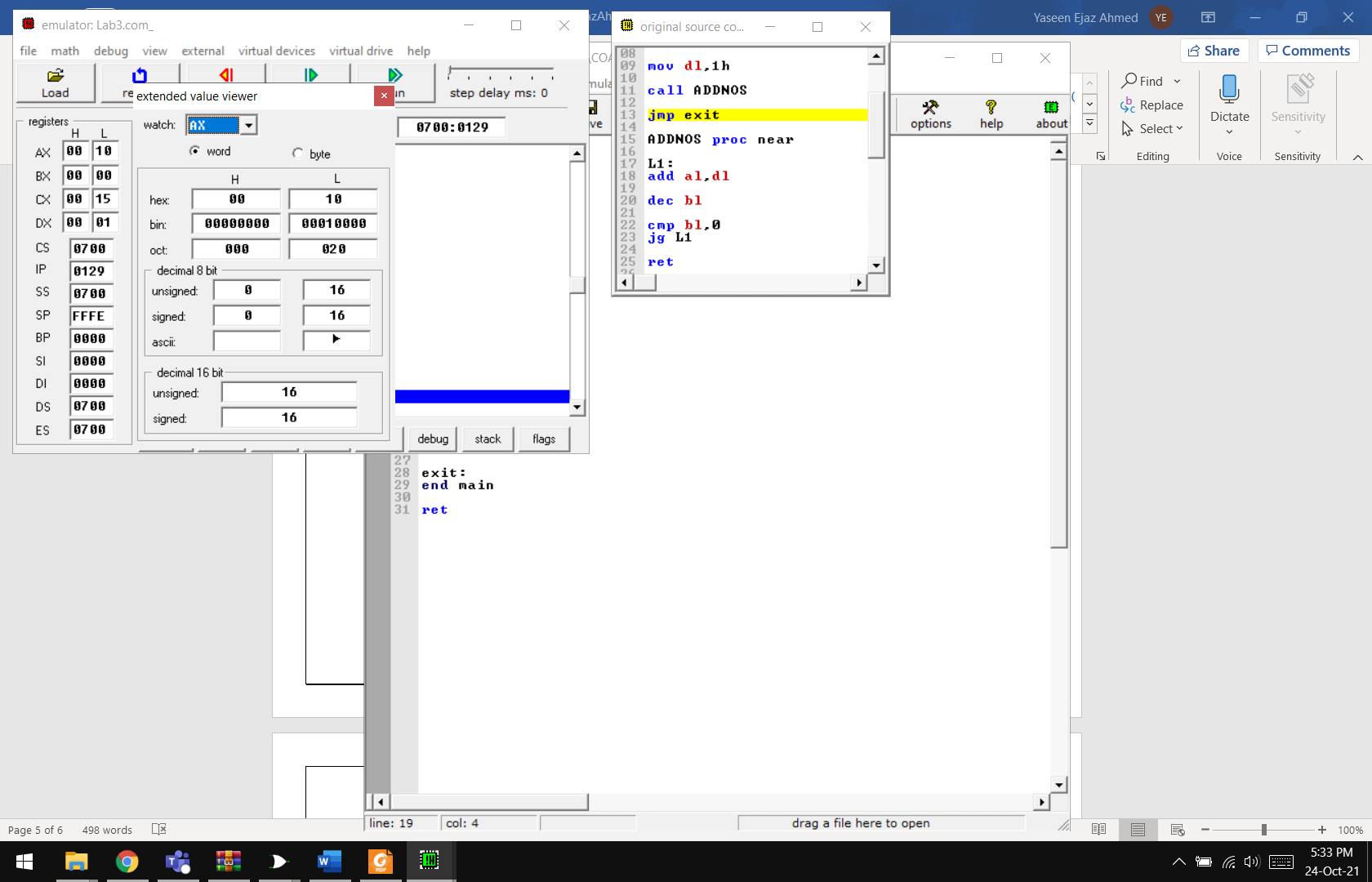
ret

exit:

end main

ret

**Output:**



**In Lab Task 01:** Write an assembly language program that uses a subroutine names ‘AVG’ to calculate the average of two 8-bit numbers. Numbers are stored in AL and AH registers and their average value should be left in BH.

**Code:**

org 100h

main proc

mov al,8h

mov ah,4h

mov dl,al

mov dh,ah

add dl,dh

mov ax,0

mov al,dl

call AVG

ret

AVG proc

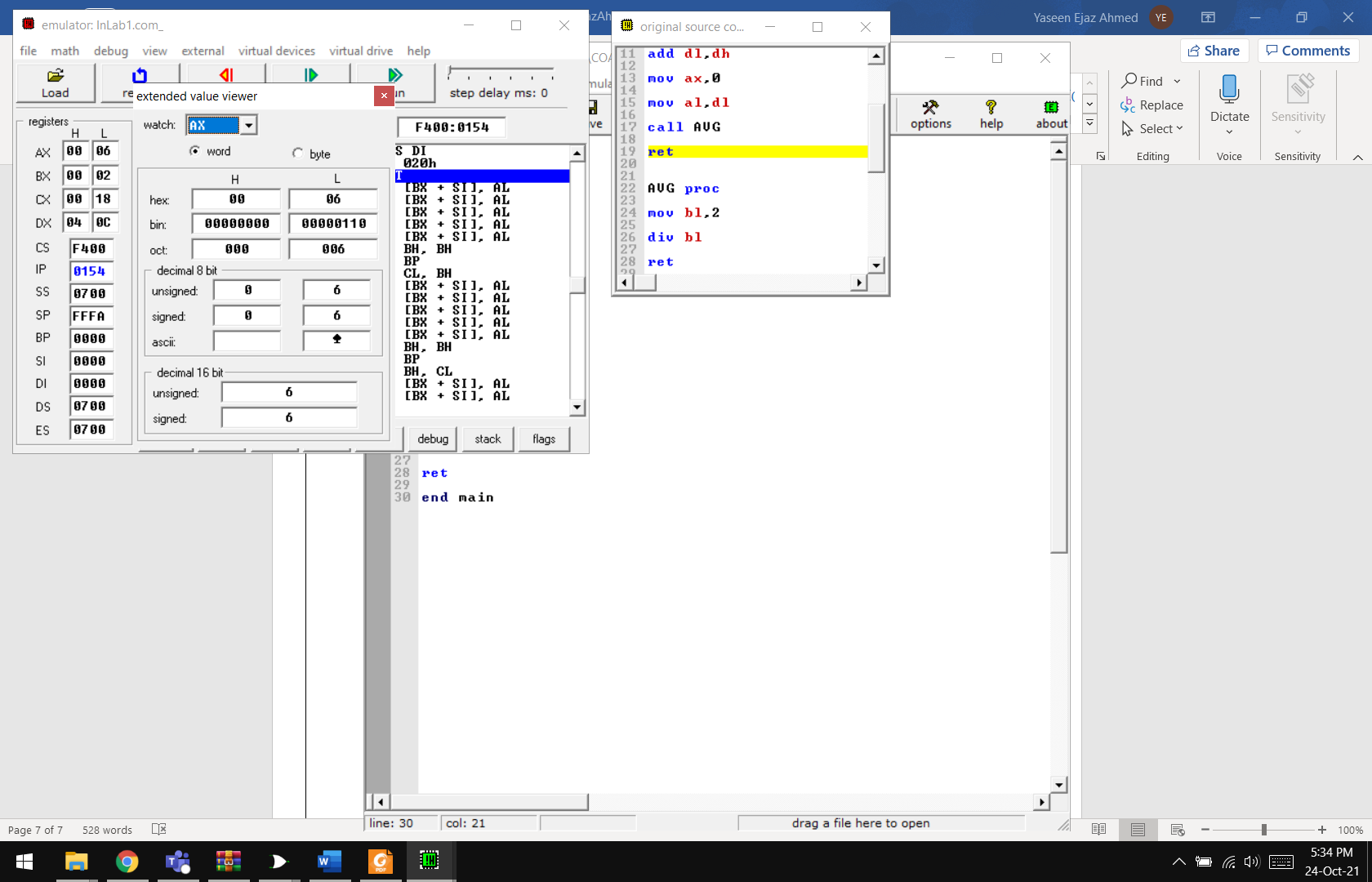
mov bl,2

div bl

ret

end main

**Output:**



**In-Lab Task 02**

Write an assembly language program that copies entire segment (source segment) with segment address 2345H to the segment (destination segment) with segment address 89ABH. Copying process should be such that the byte at the last offset of source segment must be copied to the first offset of destination segment, second last offset of source segment must be copied to the second offset of the destination segment and so on.

**Code**

org 100h

main proc

MOV [2345h],0000000011111111b

MOV AX,[2345h]

MOV [89ABH],0b

MOV BX,[89ABH]

MOV CX,16

HERE:

SHL AX,1

RCR BX,1

LOOP HERE

ret

**Output**

Graphical user interface

Description automatically generated**.**

**It’s the opposite of the value in AX**

**Results & Observations:**

In this lab we have learnt that we can use jump instructions to change the control of flow of the program. This allows us to use conditions and loops for reusability of code which is much more efficient than a code written many times. We can use these for different purposes.