Experiment 10

Introduction to DEBUG Program

Objectives:

The main objective of this lab is to get you familiarized with DOS DEGUG program.

Procedure

PART I: Immediate Operands

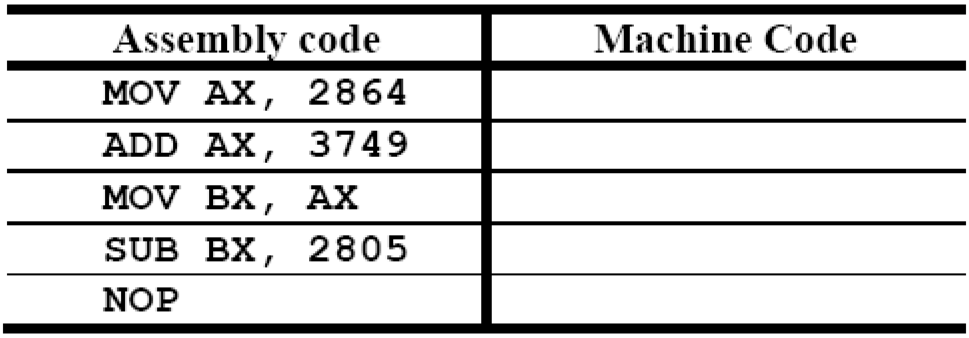
From Windows 95, 98 open a DOS window or you can use start→ run --type “command”→Enter.
Form winnt, 2000, xp use start→ run --type “cmd”→Enter.

Activate the DEBUG program by typing DEBUG at the DOS prompt. Perform the following Activities.

Activity 1.1: Enter the following program instructions in assembly code at the offset memory location 100h by typing **A 100** at the DEBUG program prompt then press Enter-key.

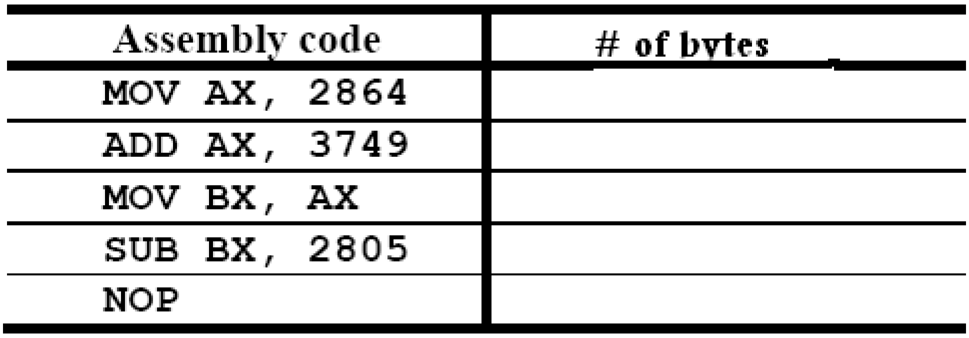
```
MOV AX, 2864
ADD AX, 3749
MOV BX, AX
SUB BX, 2805
NOP
```

Activity 1.2: Use DEBUG command **U** to unassembled the instructions in Activity 1.1.What is the machine code corresponding to each assembly code instruction?



Assembly code	Machine Code
MOV AX, 2864	
ADD AX, 3749	
MOV BX, AX	
SUB BX, 2805	
NOP	

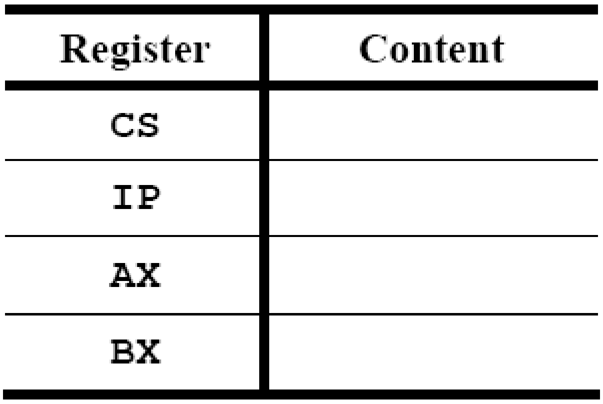
Activity 1.3: How many bytes does it need to represent each instruction in binary?



Assembly code	# of bytes
MOV AX, 2864	
ADD AX, 3749	
MOV BX, AX	
SUB BX, 2805	
NOP	

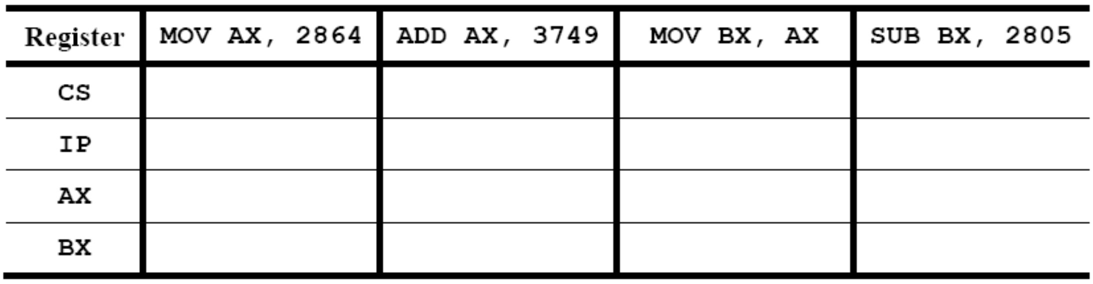
Activity 1.4: How is the (immediate) data **2864** stored at memory offset **101h**?

Activity 1.5: What are the contents of **CS**, **IP**, **AX**, and **BX**? Use DEBUG command **R** to display this information?



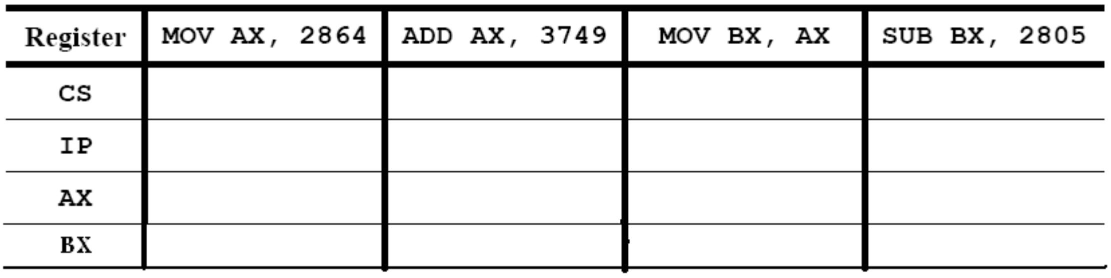
Register	Content
CS	
IP	
AX	
BX	

Activity 1.6: Predict the contents of the following registers after execution of each instruction: **CS**, **IP**, **AX**, and **BX**.



Register	MOV AX, 2864	ADD AX, 3749	MOV BX, AX	SUB BX, 2805
CS				
IP				
AX				
BX				

Activity 1.7: Use the T command to execute the program. Determine the content of the above registers after executing each instruction. Explain any discrepancies?



Register	MOV AX, 2864	ADD AX, 3749	MOV BX, AX	SUB BX, 2805
CS				
IP				
AX				
BX				

Activity 1.8: Explain why the content of IP changes after each instruction is executed.

Activity 1.9: What is the offset address of the second **MOV** instruction? What is its physical address?

PART 2: Memory Addressing

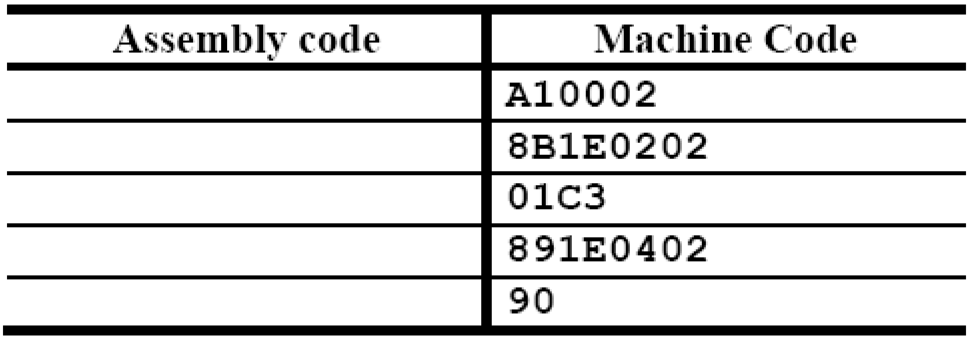
Activity 2.1: Enter the following data at the offset memory location **200h** using DEBUG command **E.**

E DS: 200 1B 9F
E DS: 202 36 4A 00 00
E DS: 206 2A 2A 2A

Activity 2.2: Enter the following program instructions in machine code at the offset memory location **100h** using DEBUG command **E.**

E CS: 100 A1 00 02
E CS: 103 8B 1E 02 02
E CS: 107 01 C3
E CS: 109 89 1E 04 02
E CS: 10D 90

Activity 2.3: What is the assembly code corresponding to each machine code Instructions?



Assembly code	Machine Code
	A10002
	8B1E0202
	01C3
	891E0402
	90

Activity 2.4: What is the 8-bit data value stored at DS: 0200 after the data in Activity 2.1has been entered? (Note that the data value is a byte).

Activity 2.5: What is the 16-bit data value stored at DS: 0200 after the data in Activity 2.1 has been entered?

Activity 2.6: Predict the data value stored at DS: 0204 AFTER the code in Activity 2.2 is executed? (Note that the data value is a byte).

Activity 2.7: Execute the program, and then determine the content of **AX** after the instruction **A10002** is executed?

AX: □

Activity 2.8: What is the content (data value) of memory at offset address 0204 after each instruction?

	A10002	8B1E0202	01C3	891E0402
DS:204				

PART 3: Entering assembly code in DEBUG

Activity 3.1: Enter the following assembly code using the DEBUG command **A** at **CS** offset address **100h**:

MOV CL, 42
MOV DL, 2A
ADD CL, DL
NOP

Activity 3.2: Execute the program using the **T** command. Determine the content of **CL**, **DL**, and **IP** after execution of each instruction?

	MOV CL,42	MOV DL,2A	ADD CL,DL
CL			
DL			
IP			