

**Don Bosco Institute of
Technology, Kurla Academic**

EXPERIMENT NO: 1

Title: Assembly Language Programming -WAP to perform Addition & Subtraction of numbers

**Don Bosco Institute of
Technology, Kurla Academic**

**EXPERIMENT NO: 1
Addition & Subtraction of numbers**

AIM	WAP to perform Addition & Subtraction of numbers (8-bits,16 bits and 32 bits)
LEARNING OBJECTIVE	To perform some arithmetic operations
LEARNING OUTCOME	Ability of the student to perform arithmetic operation by using assembly language programming.
LAB OUTCOME	CSL404.1 Ability to explain and identify different instructions of 8086 microprocessor. CSL404.3 Ability to perform arithmetic operations using assembly language programming.
PROGRAM OUTCOME	PO11, PO52, PO83, PO93, PO122, PSO12
BLOOM'S TAXONOMY LEVEL	Remember, Applying
THEORY	<p>Data Transfer Instructions</p> <p>Data transfer instructions in the 8086 microprocessor are used to move data between registers, memory, and I/O ports. These instructions do not affect the flags.</p> <p>Common Instructions:</p> <ol style="list-style-type: none"> MOV (Move) <ul style="list-style-type: none"> Transfers data from source to destination. Syntax: MOV destination, source <p>Example: MOV AX, BX ; Copies the value of BX into AX MOV [5000H], AL ; Copies the value in AL to memory location 5000H</p> PUSH and POP <ul style="list-style-type: none"> PUSH: Pushes data onto the stack. Syntax: PUSH source Example: PUSH AX POP: Pops data from the stack into the specified destination. Syntax: POP destination Example: POP AX XCHG (Exchange) <ul style="list-style-type: none"> Exchanges the values of the two operands. Syntax: XCHG destination, source Example: XCHG AX, BX IN and OUT <ul style="list-style-type: none"> IN: Reads data from an I/O port into the accumulator.

**Don Bosco Institute of
Technology, Kurla Academic**

Syntax: **IN AL, port**

Example: **IN AL, 02H**

- **OUT**: Sends data from the accumulator to an I/O port.

Syntax: **OUT port, AL**

Example: **OUT 02H, AL**

5. LEA (Load Effective Address)

- Loads the offset address of the source into the destination register.
- Syntax: **LEA destination, source**
- Example: **LEA BX, [SI+10H]**

Addition Instructions

Addition instructions perform arithmetic operations on data. They affect the flags (e.g., Carry Flag, Zero Flag, Overflow Flag).

Common Instructions:

1. ADD

- Adds the source to the destination.
- Syntax: **ADD destination, source**

Example:

ADD AX, BX ; AX = AX + BX

ADD AL, 05H ; Adds 5 to AL

○

2. ADC (Add with Carry)

- Adds the source, destination, and the Carry Flag (CF).
- Syntax: **ADC destination, source**

Example:

ADC AX, BX ; AX = AX + BX + CF

○

3. INC (Increment)

- Increments the value of the operand by 1.
- Syntax: **INC destination**

Example:

INC AX ; AX = AX + 1

○

Subtraction Instructions

Subtraction instructions also affect the flags. They are used to subtract data.

**Don Bosco Institute of
Technology, Kurla Academic**

	<p>Common Instructions:</p> <p>1. SUB</p> <ul style="list-style-type: none"> ○ Subtracts the source from the destination. ○ Syntax: SUB destination, source <p>Example: SUB AX, BX ; AX = AX - BX SUB AL, 05H ; Subtracts 5 from AL</p> <p>○</p> <p>2. SBB (Subtract with Borrow)</p> <ul style="list-style-type: none"> ○ Subtracts the source and borrow (Carry Flag) from the destination. ○ Syntax: SBB destination, source <p>Example: SBB AX, BX ; AX = AX - BX - CF</p> <p>○</p> <p>3. DEC (Decrement)</p> <ul style="list-style-type: none"> ○ Decrements the value of the operand by 1. ○ Syntax: DEC destination <p>Example: DEC AX ; AX = AX - 1</p>
SOFTWARE USED	Emu8086 Emulation Software
PROBLEM STATEMENTS	<p>Addition Programs:</p> <ol style="list-style-type: none"> 1. Add two numbers: 8 bits numbers using 8 bit registers 2. Add two numbers:16 bits numbers using 16 bit register 3. Add two numbers:16 bits numbers using 8 bit registers 4. Add two numbers:32 bits numbers using 16 bit registers <p>Subtraction Programs:</p> <ol style="list-style-type: none"> 1. Subtract two numbers: 8 bits numbers using 8 bit registers 2. Subtract two numbers:16 bits numbers using 8 bit registers 3. Subtract two numbers:16 bits numbers using 16 bit registers 4. Subtract two numbers:32 bits numbers using 16 bit registers <p>Content beyond syllabus: WAP to add two 8 bits numbers stored on memory location 3000:0800h and 3000:0801h and store result on 3000:0700h and carry on 3000:0701h</p>

Don Bosco Institute of Technology, Kurla Academic

OUTPUT

The image displays two screenshots of the emu8086 - assembler and microprocessor emulator 4.08 interface. The top screenshot shows the assembly code in the 'original source code' window and the 'registers' and 'memory' windows. The bottom screenshot shows the assembly code in the 'new' window and the 'registers' and 'memory' windows.

Top Screenshot:

original source code:

```
01  
02 ; You may customize this  
03 ; The location of this  
04  
05 org 100h  
06  
07 ; add your code here  
08  
09 MOV AH,0110B  
10 MOV AL,0110B  
11 ADD AH,AL  
12  
13 ret  
14  
15  
16  
17  
18
```

registers:

	H	L
AX	0C	06
BX	00	00
CX	00	07
DX	00	00
CS	0700	
IP	0106	
SP	0700	
BP	FFFE	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

memory:

Address	Value	Instruction
07100	B4 180	MOV AH, 06h
07101	06 006	MOV AL, 06h
07102	80 176	ADD AH, AL
07103	06 006	RET
07104	02 002	NOP
07105	E0 224	NOP
07106	C3 195	NOP
07107	90 144	NOP
07108	90 144	NOP
07109	90 144	NOP
0710A	90 144	NOP
0710B	90 144	NOP
0710C	90 144	NOP
0710D	90 144	NOP
0710E	90 144	NOP
0710F	90 144	NOP
07110	90 144	NOP

Bottom Screenshot:

new:

```
01  
02 ; You may customize this  
03 ; The location of this te  
04  
05 org 100h  
06  
07 ; add your code here  
08  
09 MOV ax,4564H  
10 MOV bx,4564H  
11 ADD ax,bx  
12  
13 ret  
14  
15  
16  
17  
18
```

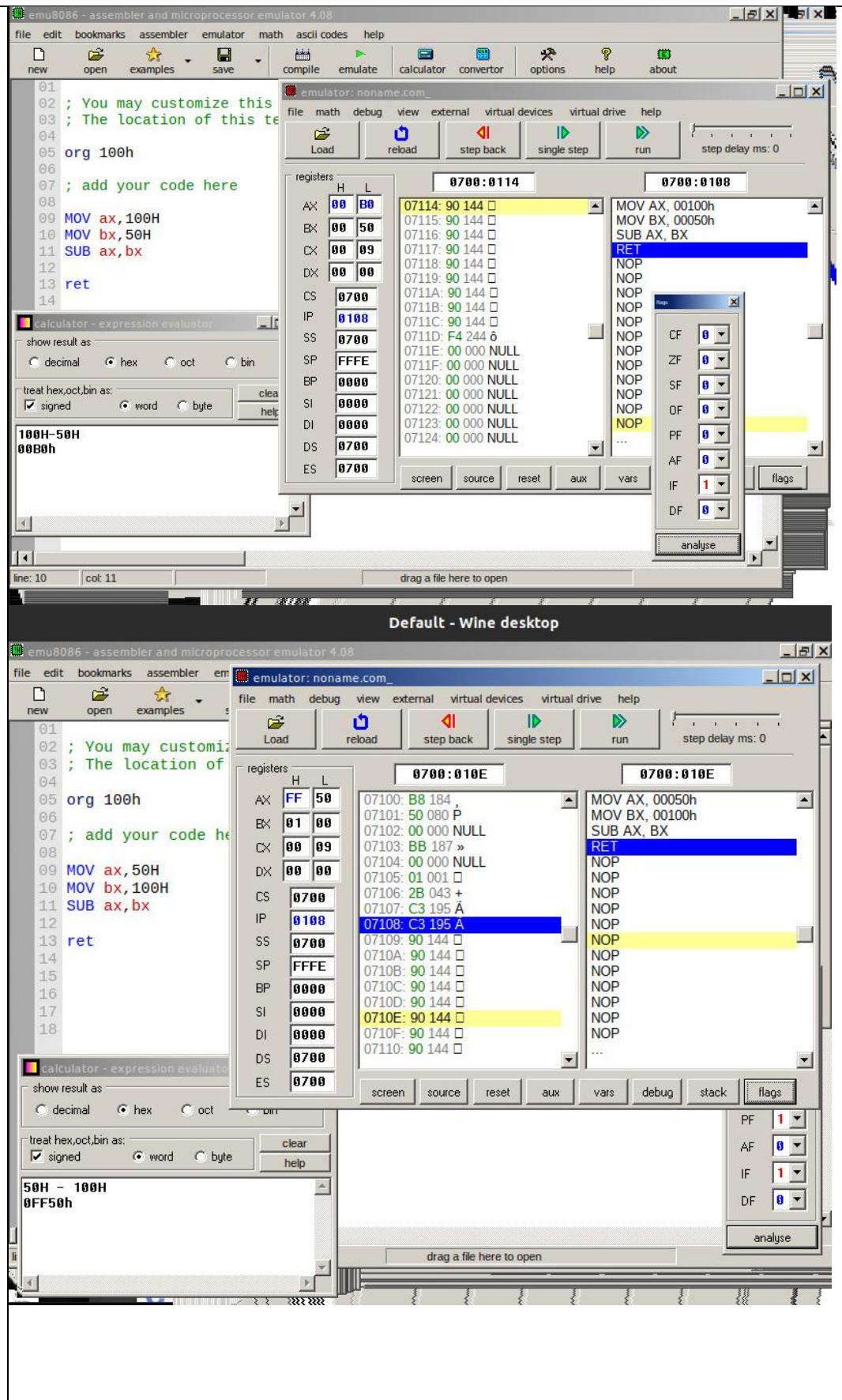
registers:

	H	L
AX	8A	C8
BX	45	64
CX	00	09
DX	00	00
CS	F400	
IP	0154	
SP	0700	
BP	FFFA	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

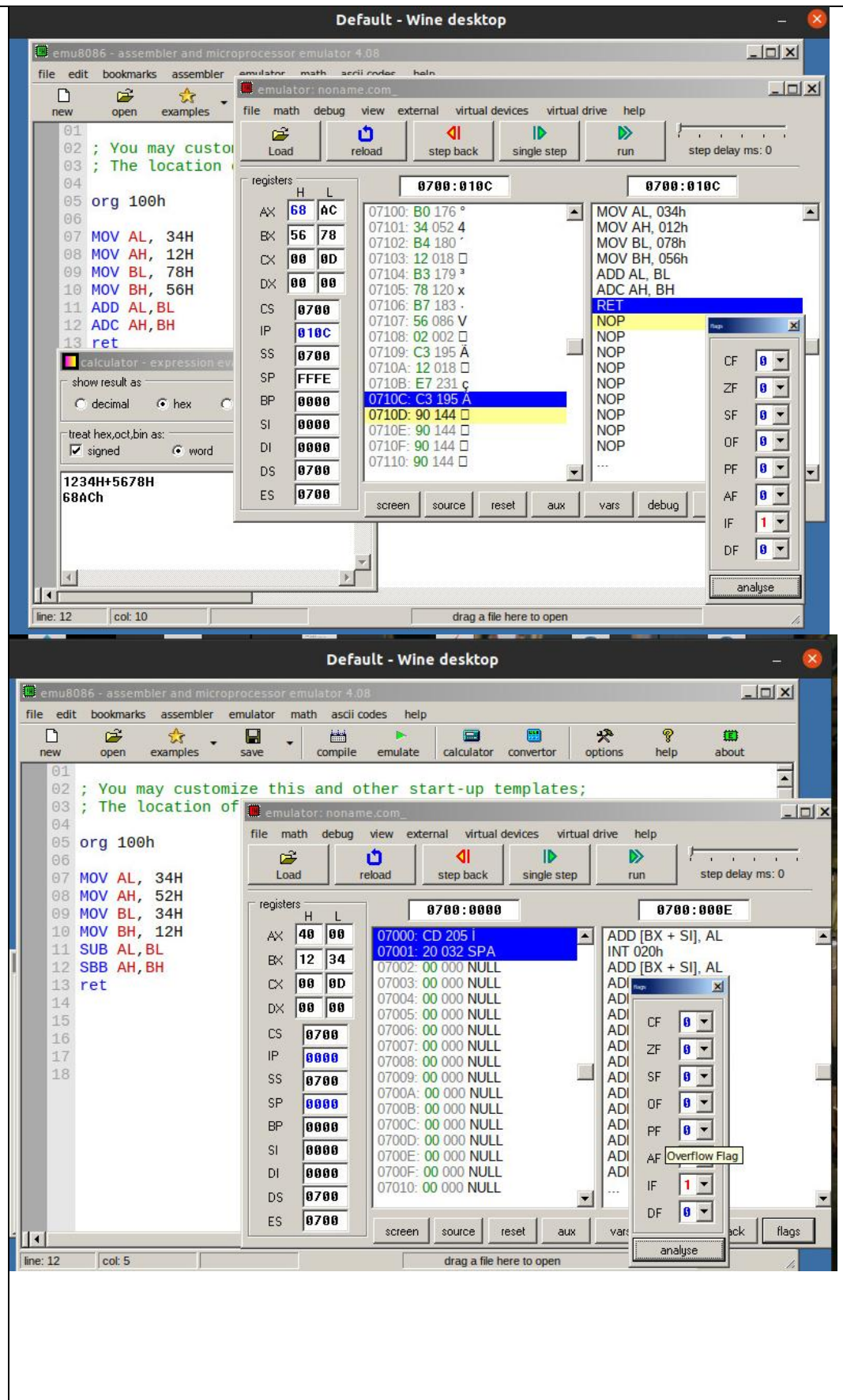
memory:

Address	Value	Instruction
F4150	FF 255	RES
F4151	FF 255	RES
F4152	CD 205	I
F4153	20 032	SPA
F4154	CF 207	I
F4155	00 000	NULL
F4156	00 000	NULL
F4157	00 000	NULL
F4158	00 000	NULL
F4159	00 000	NULL
F415A	00 000	NULL
F415B	00 000	NULL
F415C	00 000	NULL
F415D	00 000	NULL
F415E	00 000	NULL
F415F	00 000	NULL
F4160	FF 255	RES

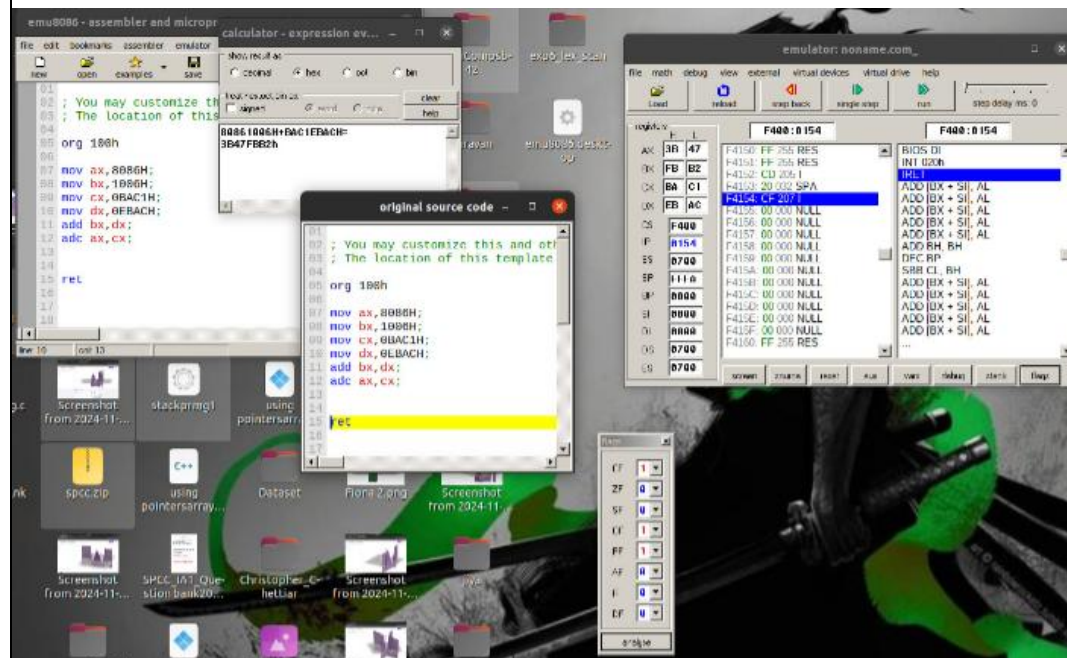
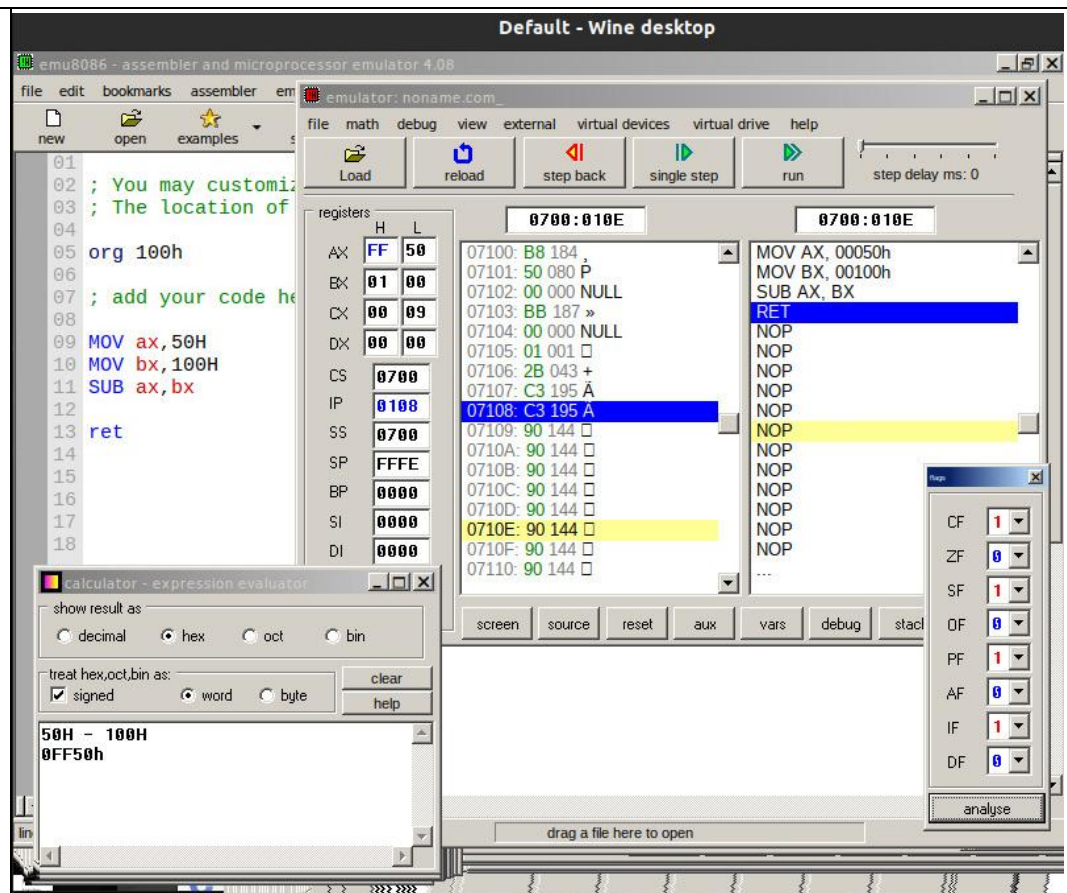
Don Bosco Institute of Technology, Kurla Academic



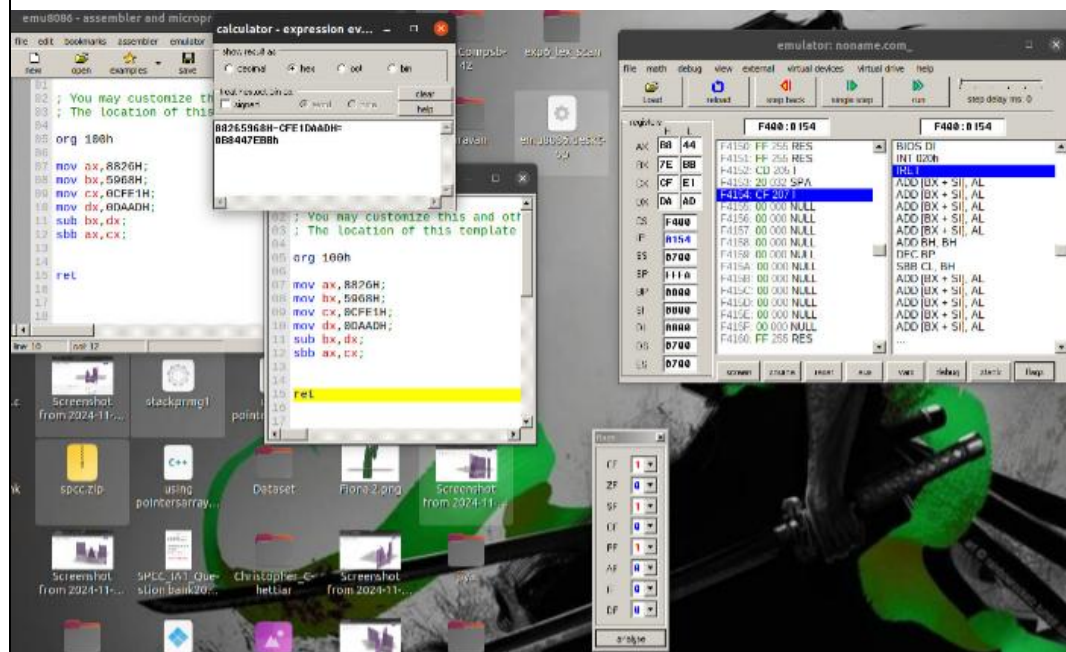
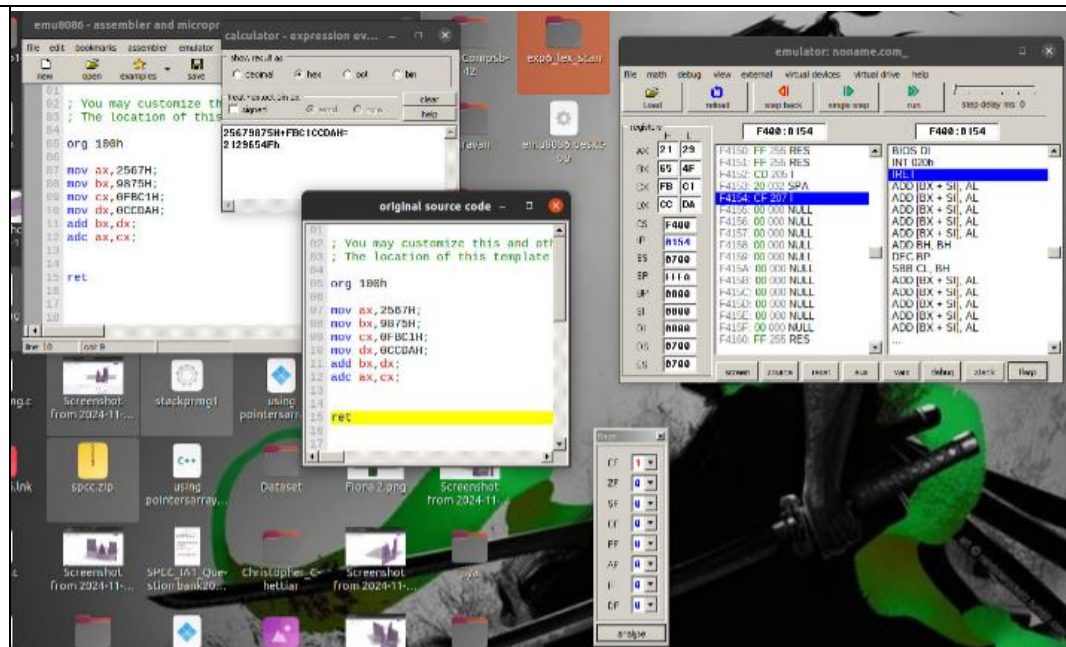
Don Bosco Institute of Technology, Kurla Academic



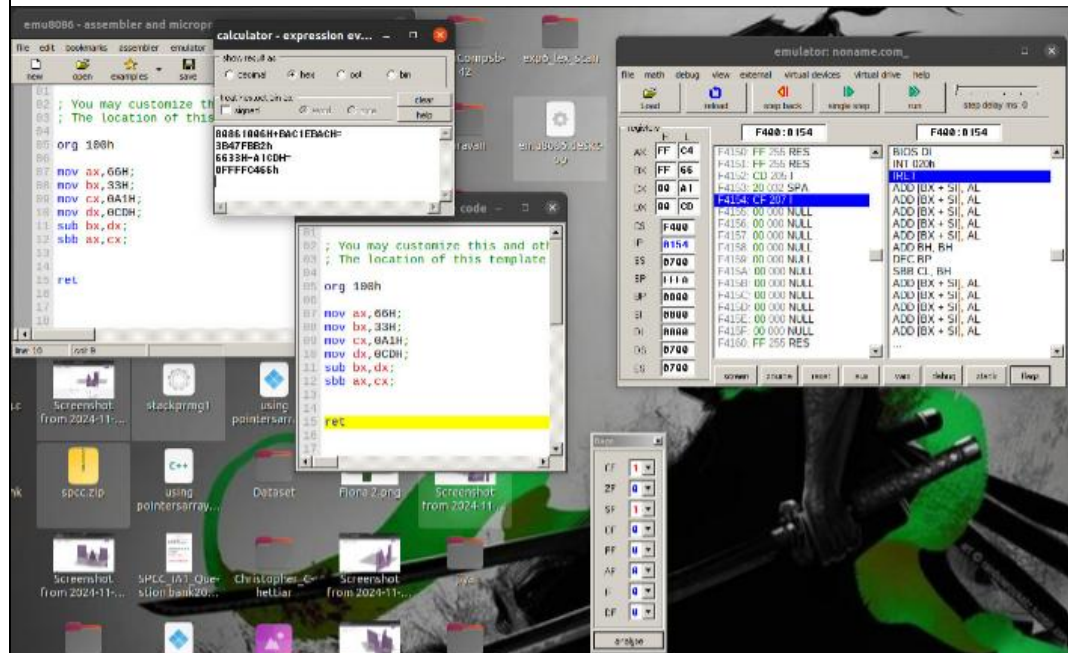
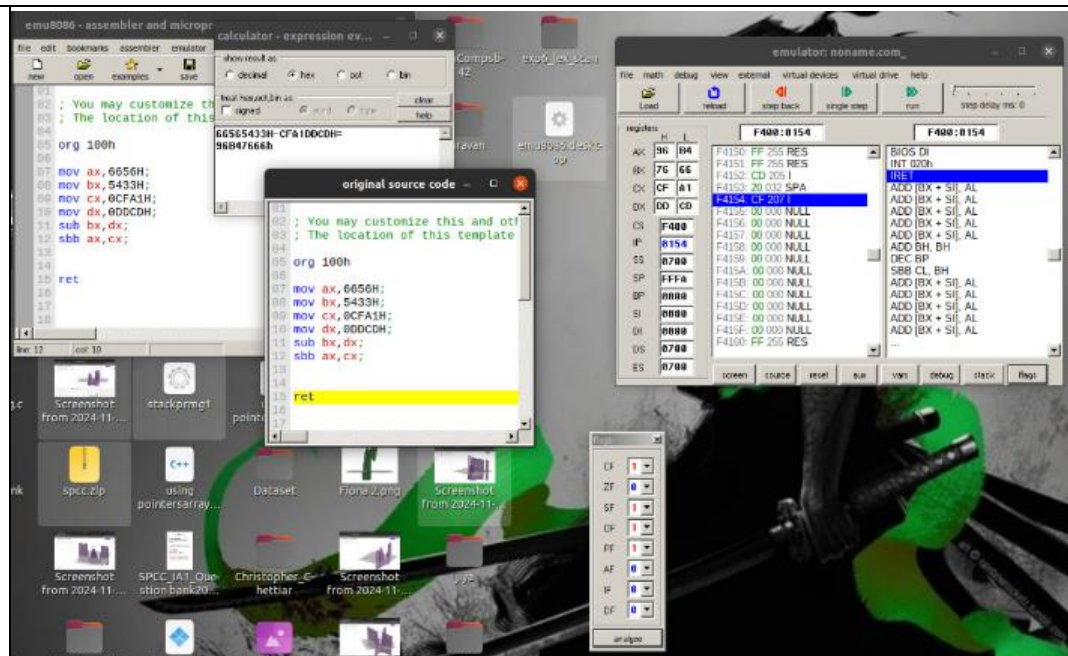
Don Bosco Institute of Technology, Kurla Academic



Don Bosco Institute of Technology, Kurla Academic



Don Bosco Institute of Technology, Kurla Academic



**Don Bosco Institute of
Technology, Kurla Academic**

CONCLUSION	Data transfer instructions move data between registers, memory, and I/O ports without affecting flags. Addition and subtraction instructions perform arithmetic operations while updating flags like Carry, Zero, and Overflow. These instructions form the foundation of assembly language programming on the 8086 microprocessor, enabling efficient data manipulation and arithmetic tasks.
REFERENCES	1. Intel Microprocessors: Barry B. Brey, 8th Edition, Pearson Education India 2. 8086/8088 family: Design Programming and Interfacing: John Uffenbeck, PHI. 3. Advanced Microprocessors and Peripherals: K M Bhurchandani, A K Ray McGraw Hill