Exp No: 9 **Floating point operations**

Date: 12/10/2020

Name: Gokhulnath T
Register Number: 185001051

Aim: To write assembly program to do following Floating point operations:

a) Floating point addition
b) Floating point subtraction

Procedure:

1. Install all the required file for executing MASM programs.(Masm, edit, link, debug etc..).
2. Write the assembly program in any editor before mounting the folder to the MASM.
3. Mount the folder that contains the assembly program with any name such as “d”.
    - mount d e:\masm
4. Create the object file of the assembly program using masm.
    - masm 16BITADD.asm
5. Use the link to create the executable file of the object file created from the above step.
    - Link 16BITADD.obj
6. Run the executable file using debug.
    - debug 16BITADD.exe
7. By un-assembling the program you can check the code segment of the program
    - u 076b:0100
8. To check the data memory segment, you can use the memory option to view the data stored.
    - d 076a:0000
9. To enter your own values, you can use the enter option which will prompt for new values.
    - e 076a:0000
10. To execute the program, you can use go option
    - G
11. After successful execution and termination of the program, you can check the result by checking the data memory segment
    - d 076a:0000
12. The result can be viewed in the respective address mentioned in the program.

9 a) Floating point addition

Algorithm:

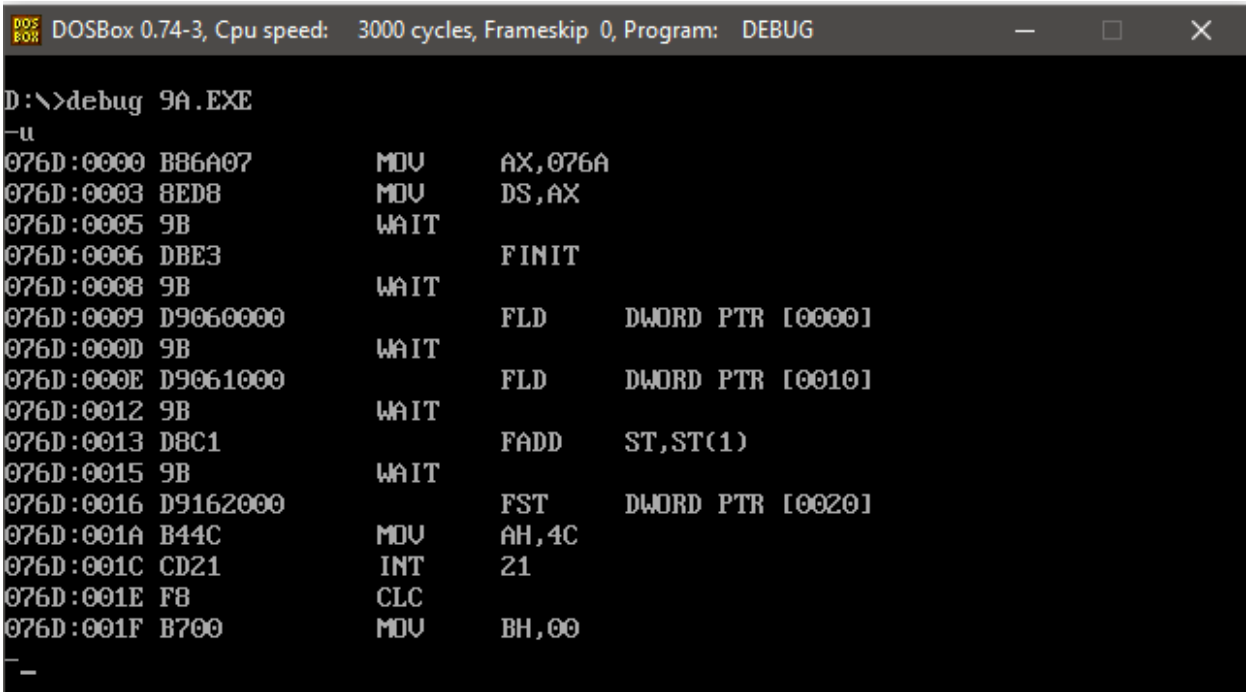
a) Assign data to ax register
b) Load contents of memory location ax in register ds
c) Intialize 8087 stack
d) Load the first operand to st(0)
e) Load the second operand to st(1)
f) Add st(0) and st(1)
g) Store sum in st(0)
h) Load content 4ch termination code to ah register (setup function-4C of the int21)
i) Call BIOS int21 to return to DOS

Program:

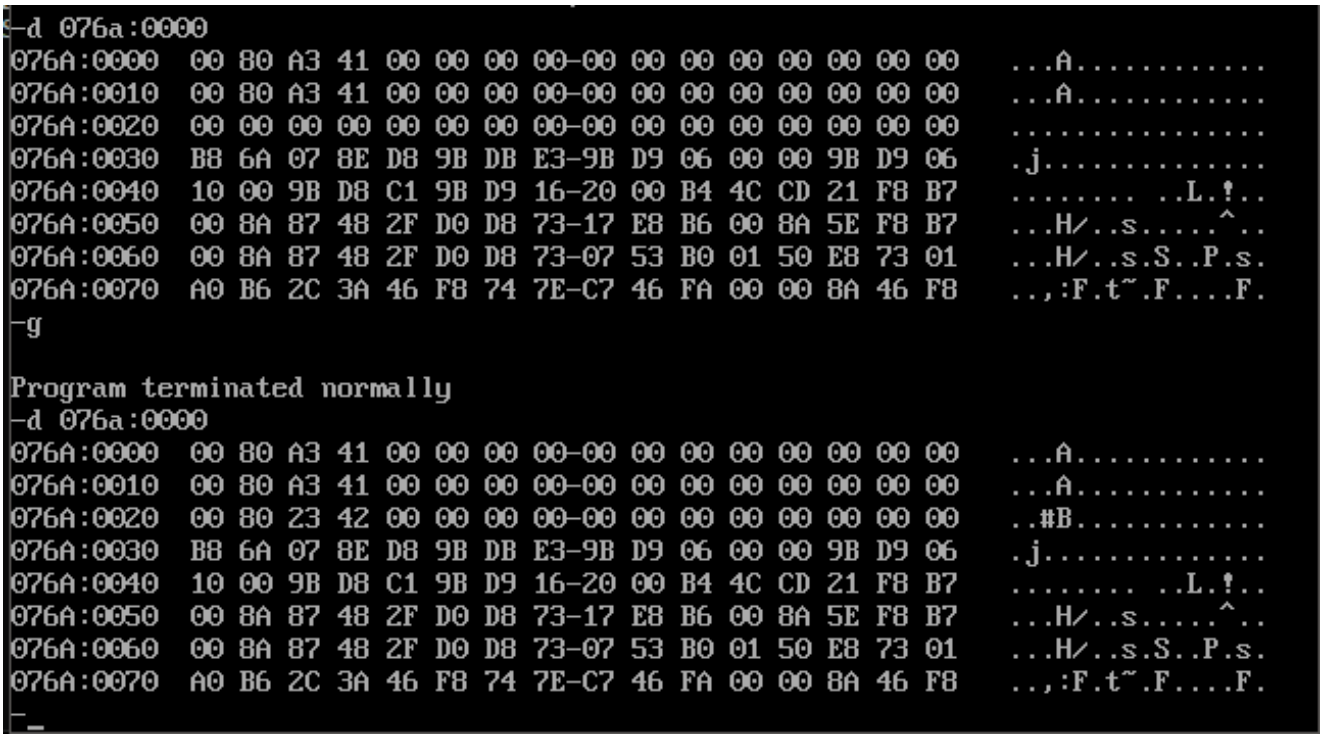
Program	**Comments**
ASSUME CS:CODESEG, DS:DATASEG	Initializing the code, data and extra segments to assembler
DATASEG SEGMENT	Data segment
ORG 00H	Indicating the x data segment to be in 00h
X DD 20.4375	X is declared and initialized to 20.4375
ORG 10H	Indicating the y data segment to be in 10h
Y DD 20.4375	y is declared and initialized to 20.4375
ORG 20H	Indicating the sum data segment to be in 20h
SUM DD ?	Sum is declared
DATASEG ENDS	
CODESEG SEGMENT	Code segment
start: MOV AX,DATASEG	Transferring the data from memory location data to ax
MOV DS,AX	Transferring the data from memory location ax to ds
FINIT	initialize 8087 stack
FLD X	load X into ST(0)
FLD Y	load Y into ST(1)
FADD ST(0),ST(1)	ST(0) = X+Y

FST SUM	store ST(0) in sum
MOV Ah,4CH	setup function-4C of the int21
INT 21H	call BIOS int21 to return to DOS
CODE ENDS	Code ends
end start	

Unassembled code:



Sample Input and output:



20.4375 + 20.4375 = 40.875 (decimal)

41A38000 + 41A38000 = 42238000 (hexadecimal)

Result:

Thus, the assembly program for Floating point addition is written and executed.

9 b) Floating point subtraction

Algorithm:

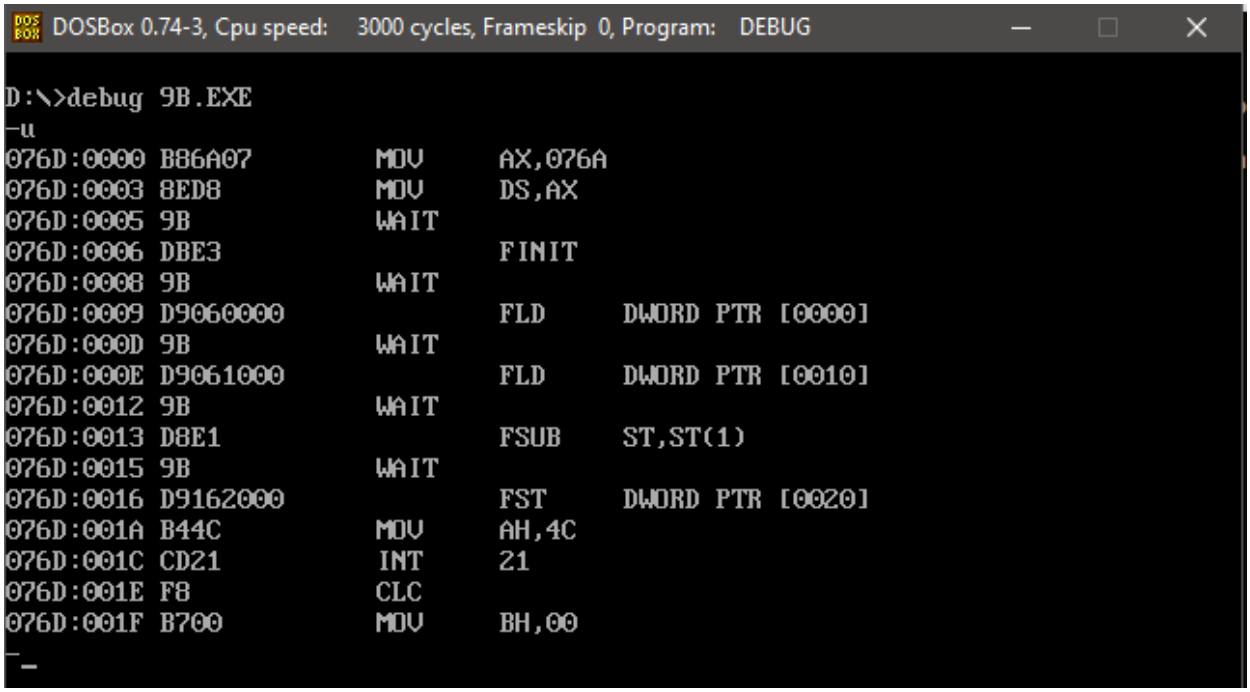
a) Assign data to ax register
b) Load contents of memory location ax in register ds
c) Intialize 8087 stack
d) Load the first operand to st(0)
e) Load the second operand to st(1)
f) Subtract st(0) and st(1)
g) Store sum in st(0)
h) Load content 4ch termination code to ah register (setup function-4C of the int21)
i) Call BIOS int21 to return to DOS

Program:

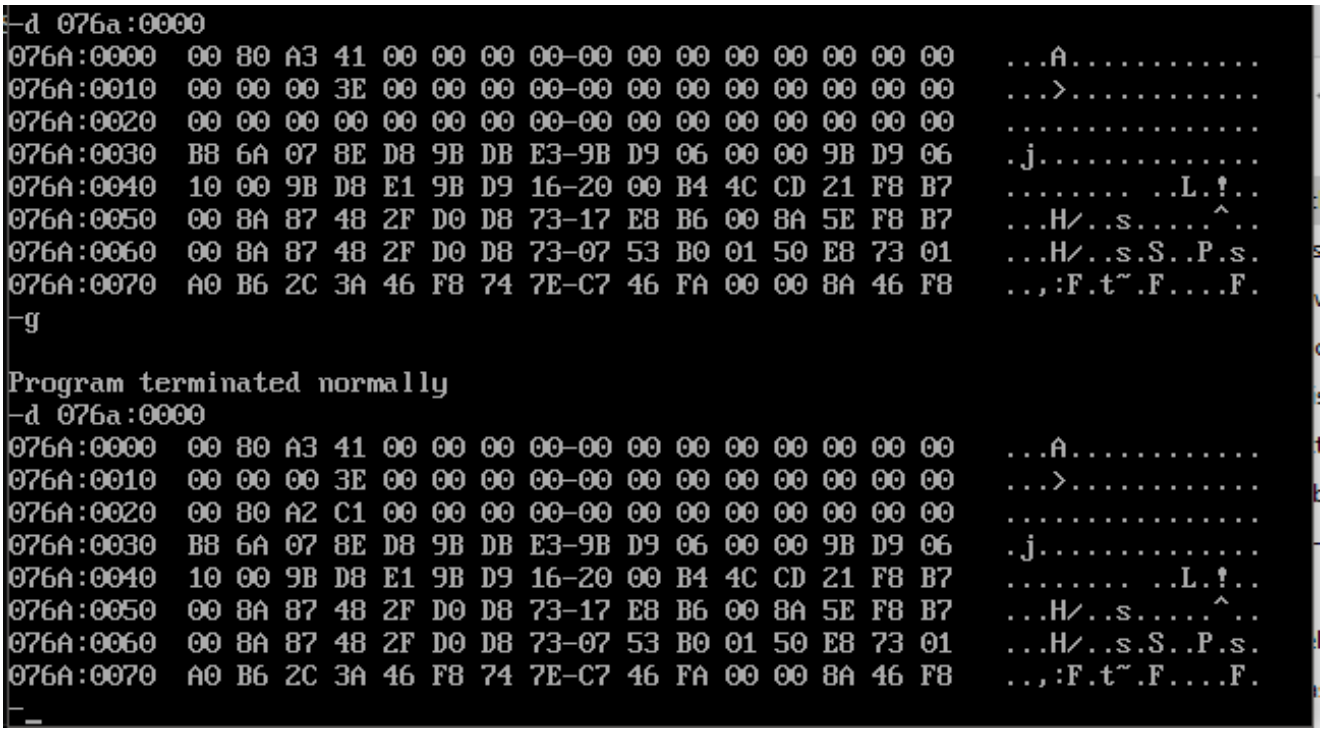
Program	**Comments**
ASSUME CS:CODESEG, DS:DATASEG	Initializing the code, data and extra segments to assembler
DATASEG SEGMENT	Data segment
ORG 00H	Indicating the x data segment to be in 00h
X DD 20.4375	X is declared and initialized to 20.4375
ORG 10H	Indicating the y data segment to be in 10h
Y DD 00.125	y is declared and initialized to 00.125
ORG 20H	Indicating the sum data segment to be in 20h
SUM DD ?	Sum is declared
DATASEG ENDS	
CODESEG SEGMENT	Code segment
start: MOV AX,DATASEG	Transferring the data from memory location data to ax
MOV DS,AX	Transferring the data from memory location ax to ds
FINIT	initialize 8087 stack
FLD X	load X into ST(0)
FLD Y	load Y into ST(1)
FSUB ST(0),ST(1)	ST(0) = X-Y

FST SUM	store ST(0) in sum
MOV Ah,4CH	setup function-4C of the int21
INT 21H	call BIOS int21 to return to DOS
CODE ENDS	Code ends
end start	

Unassembled code:



Sample Input and output:



20.4375 – 00.125= 20.3125(decimal)

41A38000 + 3E000000 = C1A28000 (hexadecimal)

Result:

Thus, the assembly program for Floating point subtraction is written and executed.