Floating Point Operations

Expt No:	9	**Name:**	Mahesh Bharadwaj K
Date :	15/10/2020	**Reg No:**	185001089

Aim:

To write and execute 8086 programs for floating point addition and subtraction.

Procedure:

- Mount masm folder to a drive on DOSBOX.
- Navigate to mounted drive using 'dir' .
- Save 8086 program with the extension **'.asm'** in the same folder using the command **'edit'**.
- Assemble the **.asm** file using the command **'masm filename.asm'**.
- Link the assmebled **.obj** file using the command **'link filename.obj'**.
- Debug the executable file **.exe** with the **'debug filename.exe'** command.

   i. **U:** To view the un-assembled code.
   ii. **D:** Used as 'D segment:offset' to see the content of memory locations starting from segment:offset address.
   iii. **E:** To change the values in memory.
   iv. **G:** Execute the program using command.
   v. **Q** exits from the debug session.

Algorithm:

1. Addition

∗ START: Move the starting address of data segment to AX register and move the data from AX register to DS register.

∗ Initialize the 8087-stack using FINIT command. This stack will be used for floating point operations.

∗ Load the floating-point number from variable X to the top of the stack i.e. ST (0) using FLD command.

∗ Now again load the floating-point number from variable Y to the top of the stack i.e. ST (0) using FLD command. The previous stack top contents will be pushed into the stack.

∗ Using FADD add ST(0) and ST(1) which stores the result in ST(0).

∗ Using FST store the resulting floating-point number from the top of the stack to the variable SUM.

2. Subtraction

∗ START: Move the starting address of data segment to AX register and move the data from AX register to DS register.

∗ Initialize the 8087-stack using FINIT command. This stack will be used for floating point operations.

∗ Load the floating-point number from variable X to the top of the stack i.e. ST (0) using FLD command.

∗ Now again load the floating-point number from variable Y to the top of the stack i.e. ST (0) using FLD command. The previous stack top contents will be pushed into the stack.

∗ Using FSUB sub ST(0) from ST(1) and store the result in ST(0).

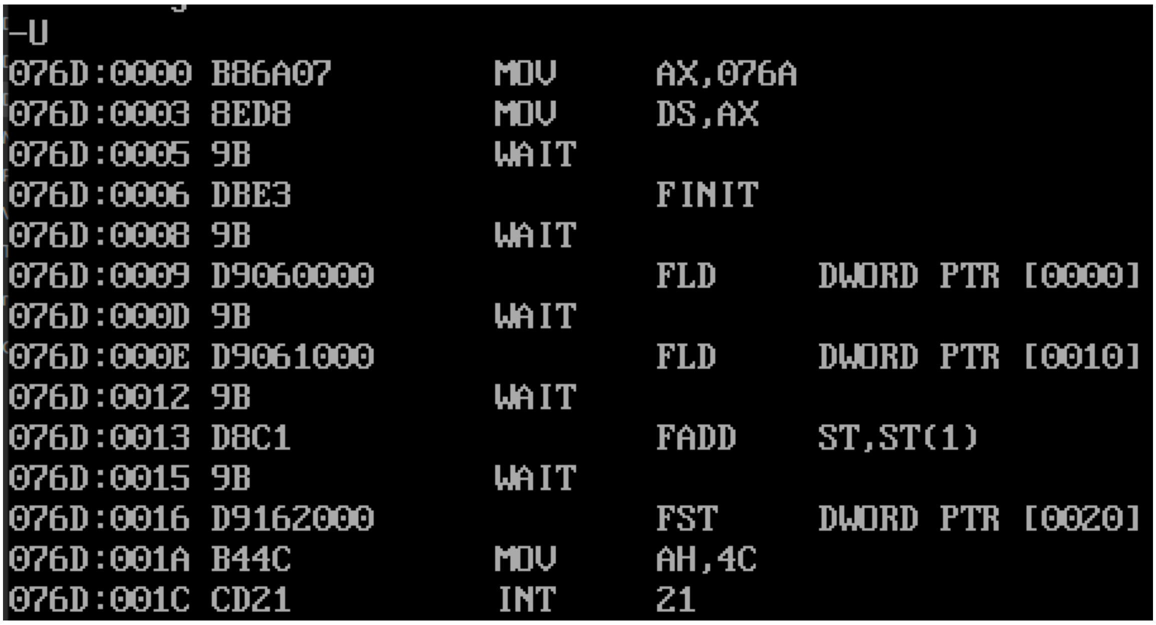
∗ . Using FST store the resulting floating-point number from the top of the stack to the variable SUM.

Program:

1. Addition

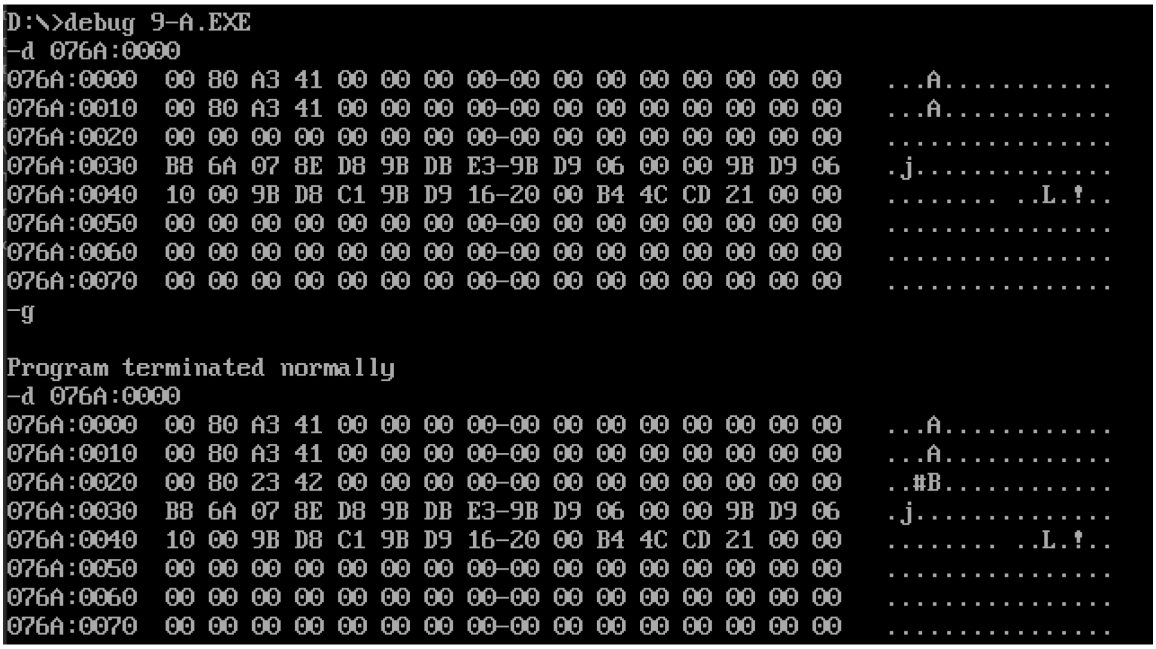
Program	Comments
start: MOV AX,data	Move data segment address contents to AX register
MOV ds,AX	Move data in AX register to DS register
FINIT	Initialize 8087 stack.
FLD X	load X into ST(0)
FLD Y	load Y into ST(0)
FADD ST(0), ST(1)	ST(0) = X + Y
FST SUM	Store ST(0) in sum.
MOV ah,4ch	
INT 21h	Request interrupt routine

Unassembled Code:



```
-U
076D:0000 B86A07        MOV     AX,076A
076D:0003 8ED8          MOV     DS,AX
076D:0005 9B            WAIT
076D:0006 DBE3                  FINIT
076D:0008 9B            WAIT
076D:0009 D9060000              FLD     DWORD PTR [0000]
076D:000D 9B            WAIT
076D:000E D9061000              FLD     DWORD PTR [0010]
076D:0012 9B            WAIT
076D:0013 D8C1                  FADD    ST,ST(1)
076D:0015 9B            WAIT
076D:0016 D9162000              FST     DWORD PTR [0020]
076D:001A B44C          MOV     AH,4C
076D:001C CD21          INT     21
```

Input and Output:



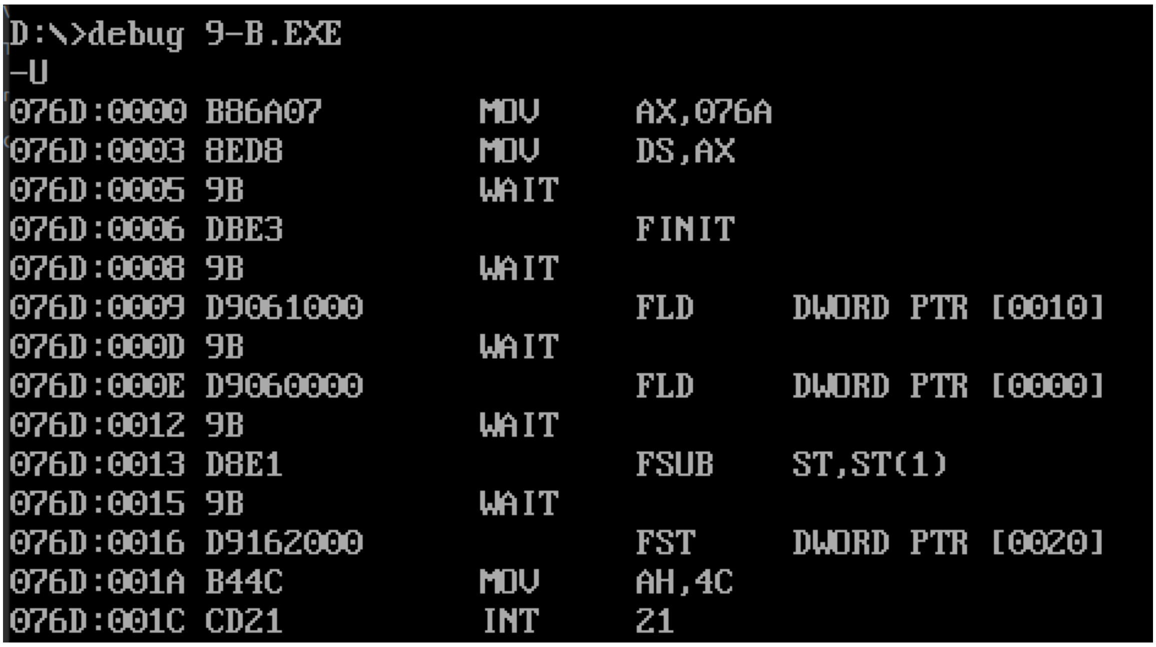
```
D:\>debug 9-A.EXE
-d 076A:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 C1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-g

Program terminated normally
-d 076A:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0020  00 80 23 42 00 00 00 00-00 00 00 00 00 00 00 00   ..#B............
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .j..............
076A:0040  10 00 9B D8 C1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

2. Subtraction

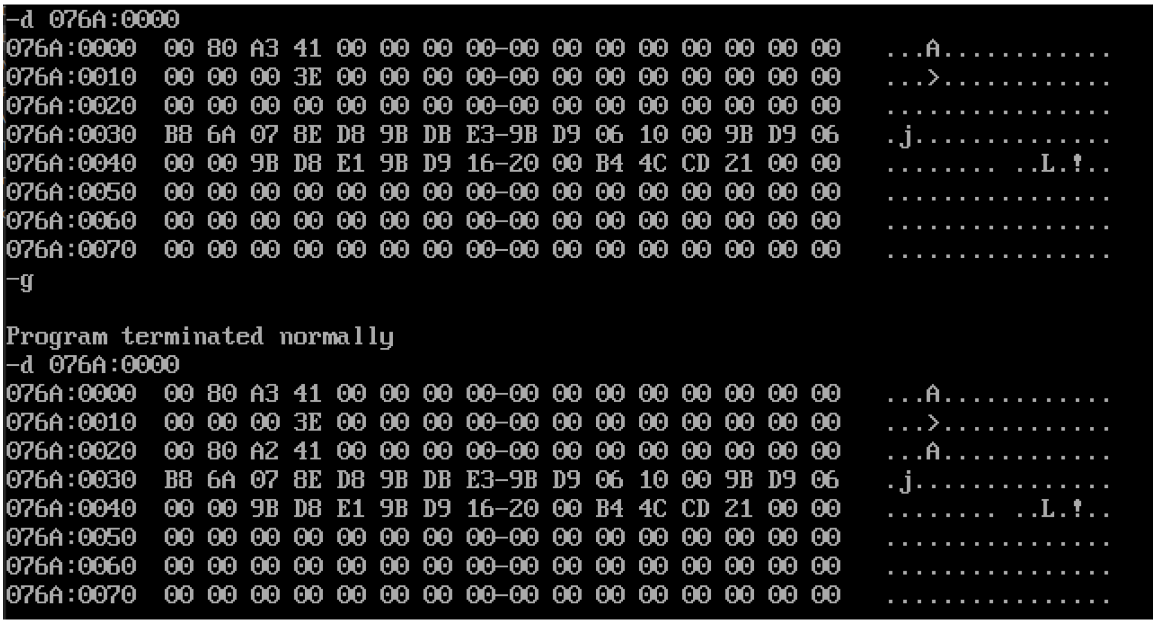
	Program	Comments
start:	MOV AX,data	Move data segment address contents to AX register
	MOV ds,AX	Move data in AX register to DS register
	FINIT	Initialize 8087 stack.
	FLD X	load X into ST(0)
	FLD Y	load Y into ST(0)
	FSUB ST(0), ST(1)	ST(0) = X - Y
	FST DIFF	Store ST(0) in diff.
	MOV ah,4ch	
	INT 21h	Request interrupt routine

Unassembled Code:



```
D:\>debug 9-B.EXE
-U
076D:0000 B86A07        MOV     AX,076A
076D:0003 8ED8          MOV     DS,AX
076D:0005 9B            WAIT
076D:0006 DBE3                  FINIT
076D:0008 9B            WAIT
076D:0009 D9061000              FLD     DWORD PTR [0010]
076D:000D 9B            WAIT
076D:000E D9060000              FLD     DWORD PTR [0000]
076D:0012 9B            WAIT
076D:0013 D8E1                  FSUB    ST,ST(1)
076D:0015 9B            WAIT
076D:0016 D9162000              FST     DWORD PTR [0020]
076D:001A B44C          MOV     AH,4C
076D:001C CD21          INT     21
```

Input and Output:



```
-d 076A:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 00 00 3E 00 00 00 00-00 00 00 00 00 00 00 00   ...>............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 10 00 9B D9 06   .j..............
076A:0040  00 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-g

Program terminated normally
-d 076A:0000
076A:0000  00 80 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0010  00 00 00 3E 00 00 00 00-00 00 00 00 00 00 00 00   ...>............
076A:0020  00 80 A2 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
076A:0030  B8 6A 07 8E D8 9B DB E3-9B D9 06 10 00 9B D9 06   .j..............
076A:0040  00 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 00 00   ........ ..L.!..
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

Result:

8086 ASL programs for floating point operations have been executed successfully using MS - DOSBox.