Floating Point Operations

Expt No:	9	**Name:**	Shivanirudh S G
Date :	16/10/2020	**Reg No:**	185001146

Aim:

To perform floating point operations in 8086.

Floating point Addition

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Initialise 8087 microprocessor using command FINIT.
- Load num1 and num2 onto the 8087 stack using FLD num1 and FLD num2 commands.
- Add stack elements 0 and 1 using FADD ST(0), ST(1).
- Store the result in sum using FST sum.

Program:

Program	Comments
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
org 00H	Store at offset 00
num1 dd 20.4325	Define decimal word num1 with value 20.4325
org 10H	Store at offset 10
num2 dd 20.4575	Define decimal word num2 with value 20.4575
org 20H	Store at offset 20
sum dd ?	Define decimal word sum to store result
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
finit	Initialise 8087 microprocessor
fld num1	Load num1 into stack of 8087
fld num2	Load num2 into stack of 8087
fadd st(0), st(1)	ST(0) = ST(0) + ST(1)
fst sum	Store value fo ST(0) in sum
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:

```
-u
0E27:0000 B8240E        MOV     AX,0E24
0E27:0003 8ED8          MOV     DS,AX
0E27:0005 9B            WAIT
0E27:0006 DBE3                  FINIT
0E27:0008 9B            WAIT
0E27:0009 D9060000              FLD     DWORD PTR [00
0E27:000D 9B            WAIT
0E27:000E D9061000              FLD     DWORD PTR [00
0E27:0012 9B            WAIT
0E27:0013 D8C1                  FADD    ST,ST(1)
0E27:0015 9B            WAIT
0E27:0016 D9162000              FST     DWORD PTR [00
0E27:001A B44C          MOV     AH,4C
0E27:001C CD21          INT     21
0E27:001E D0D8          RCR     AL,1
-_
```

Input and Output:

```
-d 0e24:0000
0E24:0000  C3 75 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   .u.A............
0E24:0010  F6 A8 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
0E24:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0030  B8 24 0E 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .$..............
0E24:0040  10 00 9B D8 C1 9B D9 16-20 00 B4 4C CD 21 D0 D8   ........ ..L.!..
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  C3 75 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   .u.A............
0E24:0010  F6 A8 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
0E24:0020  5C 8F 23 42 00 00 00 00-00 00 00 00 00 00 00 00   \.#B............
0E24:0030  B8 24 0E 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .$..............
0E24:0040  10 00 9B D8 C1 9B D9 16-20 00 B4 4C CD 21 D0 D8   ........ ..L.!..
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-_
```

Figure 1: **Input:** num1: 20.4325, num2: 20.4575; **Output:** sum: 40.69

Floating point Subtraction

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Initialise 8087 microprocessor using command FINIT.
- Load num1 and num2 onto the 8087 stack using FLD num1 and FLD num2 commands.
- Add stack elements 0 and 1 using FSUB ST(0), ST(1).
- Store the result in diff using FST diff.

Program:

Program	Comments
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
org 00H	Store at offset 00
num1 dd 20.4325	Define decimal word num1 with value 20.4575
org 10H	Store at offset 10
num2 dd 20.4575	Define decimal word num2 with value 20.4325
org 20H	Store at offset 20
diff dd ?	Define decimal word diff to store result
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
finit	Initialise 8087 microprocessor
fld num1	Load num1 into stack of 8087
fld num2	Load num2 into stack of 8087
fsub st(0), st(1)	ST(0) = ST(0) - ST(1)
fst diff	Store value fo ST(0) in diff
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:

```
-u
0E27:0000 B8240E        MOV     AX,0E24
0E27:0003 8ED8          MOV     DS,AX
0E27:0005 9B            WAIT
0E27:0006 DBE3                  FINIT
0E27:0008 9B            WAIT
0E27:0009 D9060000              FLD     DWORD PTR [00
0E27:000D 9B            WAIT
0E27:000E D9061000              FLD     DWORD PTR [00
0E27:0012 9B            WAIT
0E27:0013 D8E1                  FSUB    ST,ST(1)
0E27:0015 9B            WAIT
0E27:0016 D9162000              FST     DWORD PTR [00
0E27:001A B44C          MOV     AH,4C
0E27:001C CD21          INT     21
0E27:001E D0D8          RCR     AL,1
-_
```

Input and Output:

```
-d 0e24:0000
0E24:0000  F6 A8 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
0E24:0010  C3 75 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   .u.A............
0E24:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0030  B8 24 0E 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .$..............
0E24:0040  10 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 D0 D8   ........ ..L.!..
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  F6 A8 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   ...A............
0E24:0010  C3 75 A3 41 00 00 00 00-00 00 00 00 00 00 00 00   .u.A............
0E24:0020  00 CC CC BC 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0030  B8 24 0E 8E D8 9B DB E3-9B D9 06 00 00 9B D9 06   .$..............
0E24:0040  10 00 9B D8 E1 9B D9 16-20 00 B4 4C CD 21 D0 D8   ........ ..L.!..
0E24:0050  73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00   s........v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-
```

Figure 2: **Input:** num1: 20.4575, num2: 20.4325;
Output: difference: 0.025

Result:

The 8086 programs were written to perform Floating Point operations, and the results observed.