Assignment - 10

GoodLuck | Page No. | Date

Title :- Roots of equation.

Problem defination :-

Write 80386 ALP to find roots of quadratic equation, All possible cases must be considered.

Objective :-

i) To understand 80387 instruction set ii) To understand i/p & o/p of floating point numbers.

S/w requirement :-

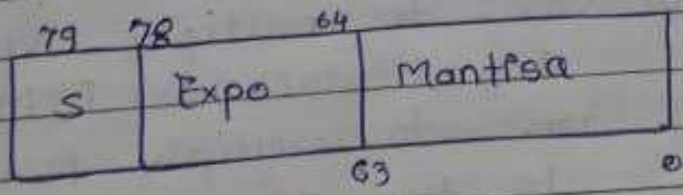
Text editor, Assembler (NASM), compiler (GCC), Debuger.

Theory :-

i) 80387 is the math coprocessor of 80386.

ii) It can be used for floating point operation in assembly.

iii) It consists of an 80 bit register stack which uses extended precision format as follows :-



1 bit - Sign.
15 bit - Exponent.
64 bit - Mantisa.

- The top of the stack can be accessed as 'st0'
Similarly, lower element as 'st1', 'st2',....

3.69	← st0
1.34	← st1
13.45	← st2

and output
Input floating point numbers is done by making the help of printf & scanf functions of c as global variables.

The program can be compiled as
gcc -o outputfile objectfile.

Instruction used
final calculation = $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

1) FINIT - To initialize 32't stack
2) FSUB var - to substract value from top of stack ex. FSUB quad[2]
3) FADD var - to add value 'var' to top of stack, ex. FADD quad[4]
4) FIMUL var - to multiply integer var to top of stack. ex. FIMUL quad[four
5) FMUL var - to multiply float value var to top of stack. ex. FMUL quad

6) FSQRT - calculate sqrt & stores it back at the top.

1) FDIV var - divides var from top of stack ex. FDIV divvar[20],
2) FLD var - Loads value var at top of stack.
3) FST var - Stores value of top to var
4) FSTP var - Stores value of top to var and pops value from stack.

Algorithm :-

calculate roots.

1) Initialize stack.
2) Load a on top of stack
3) Multiply a to the top of stack
4) Multiply 4 to the top of stack
5) pop this value to variable 4ac
6) Load b on top of stack
7) Multiply it with itself
8) Store value of variable say bb
9) substract 4ac from top
10) pop this value to dd
11) check sign of dd, if negative set flag=1 & chang the bit value
12) load dd & perform squareroot, pop this value to 'd'
13) load a on top, multiply 2 to it.
14) pop value to 2a
15) Load d' on top of stack

If flag=a

16) Sub b from top
17) add d
18) divide 2a
19) Display value

& perform similarly for other roots.

If flag=1 (imaginary)

16). sub -b from top

17). divide 2a, this is real part, pop value

18) Add d to top, divide 2a

19) pop this value, it is imaginary part.

20). Display both real & imaginary parts & perform operation for other root.

Print values of both roots.

Conclusion :-

Thus with the help of printf, scanf & 387 math coprocessor, we were able to calculate the roots of an equation.