ASSIGNMENT NO. 10

AIM :- Write 80387 ALP to find the roots of the quadratic equation. All the possible cases must be considered in calculating the roots.

APPARATUS :

- Core 2 duo/i3/i5/i7 - 64bit processor
- OS – ubuntu 32bit/64bit OS
- Assembler used –nasm (the netwide assembler)
- Editor Used – gedit

THEORY :

A) Features of 80387:

- High performance 80-Bit Internal Architecture
- Implements ANSI/IEEE standard 754-1985 for Binary floating-point arithmetic
- Expands Intel386DX CPU data types to include 32-, 64-, 80-bit floating point, 32-, 64-bit integers and 18-bit BCD operands
- Extends Intel386DX CPU instruction set to include Trigonometric, Logarithmic, Exponential and Arithmetic instructions for all data types
- Upward object code compatible
- Full-range transcendental operations for SINE, COSINE, TANGENT, ARCTANGENT and LOGARITHM
- Built-in Exception handling
- Operates independently in all modes of 80386

- Eight 80-bit Numeric registers
- Available in 68-pin PGA package
- One version supports 16MHz-33MHz

B) Instruction of co-processor used in the assignment:

FINIT: Initialise Co-processor

FLDZ: Load zero on stack top

FILD: Load Integer on stack

FIDIV: Divide stack top by an integer value

FIMUL: Multiply stack top by an integer value

FST: Store stack top

FADD: Add in stack top

FBSTP: Store integer part of stack top in 10 byte packed BCD format

FMUL: Multiply stack top

FSQRT: Square Root of Stack Top

FSTSW: Stores the coprocessor status word

FTS: compares ST0 and 0

C) Concept of Quadratic Equation:

The quadratic formula computes the solutions to the quadratic equation:

ax 2 + bx + c = 0

The formula itself gives two solutions for x: x1 and x2.

The expression inside the square root (b 2 − 4ac) is called the discriminant. Its value is useful in determining which of the following three possibilities are true for the solutions.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The expression inside the square root ($b^2 - 4ac$) is called the discriminant. Its value is useful in determining which of the following three possibilities are true for the solutions.

1. There is only one real degenerate solution. $b^2 - 4ac = 0$
2. There are two real solutions. $b^2 - 4ac > 0$
3. There are two complex solutions. $b^2 - 4ac < 0$

Main Algorithm:

1. Start
2. Initialize a,b,c,four,two in data section
3. Initialize the co-processor
4. Load b on stack top
5. multiply top of stack with b
6. Load a on stack
7. Multiply stack top with c
8. Multiply stack top with four
9. Subtract operand 1 from operand2 of stack
10. Test stack top
11. Store co-processor status word in AX
12. Store AH value in Flag register
13. If top of stack is less than zero then jump to step xxi else continue
14. Take square root of top of stack
15. Store it in one variable let D (D=sqrt(b*b-4ac))
16. Subtract b from stacktop
17. Divide stack top by a and two

18. Call Disp_result procedure to display root1
19. Load zero on stack top
20. Subtract D from Stacktop (i.e 0-D=-D)
21. Subtract b from stacktop
22. Divide stack top by a and two
23. Call Disp_result procedure to display root2
24. Display Message “No Real Solution” using Display macro
25. Stop

Procedure: 1

1. Name : Disp_result
2. Purpose : to convert 10 bit packed BCD result in Ascii
3. I/P : value of stack top
4. Algorithm for Procedures
   i. Start
   ii. Multiply stack top by 100. (Our numbers are having two digits after
   iii. decimal point)
   iv. Store integer part of stack top in 10 byte packed BCD format using Instruction FBSTP.
   v. Store stack top in resbuff of 10 byte
   vi. RSI point to the resbuff+9
   vii. Mov value pointed by RSI in BL
   viii. Rotate BL by 4
   ix. Move BL in AL
   x. And AL with 0FH
   xi. Compare AL with 0

xii. If not equal then go to step xiv else continue

xiii. Print “+” using display Macro

xiv. Jump to step xv

xv. Print “-” using display Macro

xvi. Rotate BL by 4

xvii. Push RSI

xviii. Call disp8num procedure (Print first byte after sign)

xix. Pop RSI

xx. Dec RSI

xxi. Initialize counter RCX by 8 (8byte before decimal point)

xxii. Save RCX and RSI on stack

xxiii. Move value pointed by RSI in BL register

xxiv. Call disp8num procedure

xxv. Pop RSI and RCX

xxvi. Decrement RSI

xxvii. If RCX is not equal to zero Repeat step from xxi else continue

xxviii. Print decimal point using Display macro

xxix. Move value pointed by RSI in BL register

xxx. Call disp8num procedure

xxxi. Ret

xxxii. Stop

CONCLUSION:

_____

_____

_____