

Assignment - 10

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

Objective: To be able to solve mathematical problems in ALP.

Outcome: Students will be efficient in handling and solving mathematical problems using ALP.

Theory:

The 80387 is the first intel coprocessor to be fully compliant with IEEE 754-1985 standard. Released in 1987, a full two years after the 386 chip, the i387 includes much improved speed over intel's previous 8087/80287 coprocessors and improved characteristics of its trigonometric functions.

~~Without~~ a coprocessor the 386 normally performs floating-point arithmetic through a software routines, implemented at runtime through a software exception handler. When a math coprocessor is paired with the 386 the coprocessor performs the floating point arithmetic in hardware, returning results much faster than an (emulating) software library call.

Registers:

Each of the eight numeric registers in the 80387's registers stack is 80 bits wide and is divided into fields corresponding to the NPX's extended real data types.

Numeric instructions address the data registers relative to the register on the top of the stack. At any point in time, this top-of-stack register is indicated by the TOP field in the NPX status word. Load or push operations decrement TOP by one and load a value into the new top register.

R0 R1 R2 R3 R4 R5 R6 R7	Tag Field's			80387 Data Registers
	Sign	Exponent	Significand	

Instructions.

- 1) **FLD** - loads (pushes) the source operand onto the top of the register stack. This is done by decrementing the stack pointer by one and then copying the content of the source to the new stack top.

Syntax: **FLD** source.

- 2) **FST** - Copies the NPX stack top to the destination, which may be another register on the stack or a single or double memory operand.

Syntax:

FST destination.

- 3) **FSTP** : Operates **FST** except that the NPX stack is popped following the transfer. This is done by tagging the top stack element as empty and then incrementing TOP.

~~Syntax:~~

FSTP destination.

- 4) **FSQRT** : Replaces the content of the top stack element with its square root.

Test Case :

Input	Expected	Actual	Test Result
$a = 1$ $b = 6$ $c = 3$	$x_1 = -0.55$ $x_2 = -5.45$	$x_1 = -0.55$ $x_2 = -5.45$	Pass
$a = 1$ $b = 2$ $c = 3$	$x_1 = -1 + 1.4i$ $x_2 = -1 - 1.4i$	$x_1 = -1 + 1.41i$ $x_2 = -1 - 1.41i$	Pass

Conclusion :- We implemented myprintf and myscanf functions and successfully calculated roots of a quadratic equation using coprocessor instructions.