Министерство высшего образования и науки РФ

Вятский государственный университет

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчёт по предмету «Цифровые устройства и микропроцессоры» Лабораторная работа № 3

«Использование математического сопроцессора»

Вариант 10

Выполнил: студент группы ИНБб-3301-02-00 Кощеев Д.М.

Проверил: преподаватель Земцов М. А.

Киров 2025

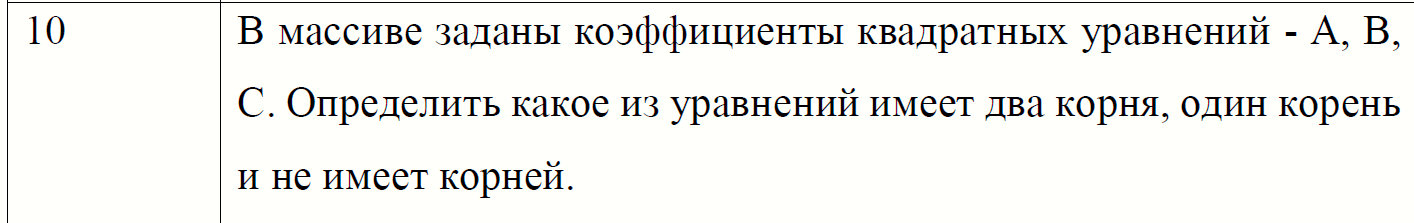
**Цель работы:** изучение принципов выполнения арифметических команд с помощью математического сопроцессора FPU микропроцессоров с архитектурой x86. 

Таблица 1 – Выражение, которое надо реализовать на Ассемблере

Код программы:

.686

.model flat, stdcall

.stack 100h

.data

; массивы A, B, C для трех уравнений

A dd 1.0, 2.0, 1.0

B dd 6.0, 4.0, 1.0

X dd 2.0, 2.0, 9.

Y dd 3244294144

n equ 3 ; Количество уравнений

discriminant dd 0.0 ; Переменная для хранения дискриминанта

; Счётчики для количества корней

two\_roots\_count dd 0

one\_root\_count dd 0

no\_roots\_count dd 0

.code

ExitProcess PROTO STDCALL :DWORD

Start:

mov ecx, n ; Количество уравнений

lea esi, [A] ; Указатель на массив A

lea edi, [B] ; Указатель на массив B

lea ebx, [X] ; Указатель на массив C

next\_equation:

; Вычисляем дискриминант D = B^2 - 4AC

fld dword ptr [edi] ; Загружаем B

fmul st(0), st(0) ; B^2

fld dword ptr [esi] ; Загружаем A

fld dword ptr [ebx] ; Загружаем C

fmul st(0), st(1) ; A \* C

fadd st(0), st(0) ; 2 \* A \* C

fadd st(0), st(0) ; 4 \* A \* C

fsubp st(1), st(0) ; B^2 - 4 \* A \* C

fstp dword ptr [discriminant] ; Сохраняем дискриминант

; Определяем количество корней

fld Y ; Загружаем 0

fld dword ptr [discriminant] ; Загружаем дискриминант

fcomip st(0), st(1) ; Сравниваем дискриминант с 0

ja two\_roots ; Если D > 0, два корня

je one\_root ; Если D = 0, один корень

jb no\_roots ; Если D < 0, нет корней

two\_roots:

; Увеличиваем счётчик для двух корней

inc dword ptr [two\_roots\_count]

jmp next\_iteration

one\_root:

; Увеличиваем счётчик для одного корня

inc dword ptr [one\_root\_count]

jmp next\_iteration

no\_roots:

; Увеличиваем счётчик для отсутствия корней

inc dword ptr [no\_roots\_count]

jmp next\_iteration

next\_iteration:

add esi, 4 ; Переход к элементу массива A

add edi, 4 ; Переход к элементу массива B

add ebx, 4 ; Переход к элементу массива C

loop next\_equation

exit:

Invoke ExitProcess, 0 ; Завершение программы

End Start

массивы A, B, C для трех уравнений

A dd 1.0, 2.0, 1.0

B dd 6.0, 4.0, 1.0

X dd 2.0, 2.0, 9.

В первом уравнении берём целые числа 1, 6, 2. Дискриминант этих трёх значений является положительным, следовательно имеет 2 корня. На рисунке 1 представлена переменная two\_roots\_count, предназначенная для определения дискриминанта, у которого получается два корня. Также видно, что она имеет значение 1 после первого цикла нахождения дискриминанта.

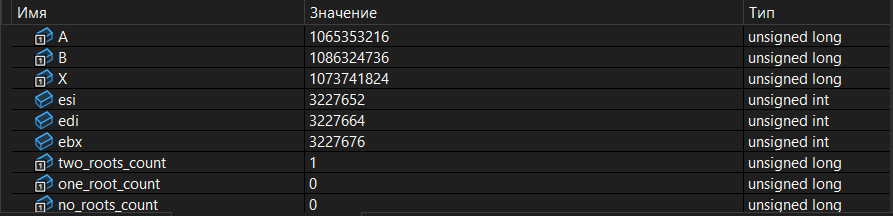


Рисунок 1 –Результаты после первого вычисления

Во втором уравнении берём целые числа 6, 4, 1. Дискриминант этих трёх значений является ноль, следовательно имеет 1 корень. На рисунке 2 представлена переменная one\_roots\_count, предназначенная для определения дискриминанта, у которого получается один корень. Также видно, что она имеет значение 1 после второго цикла нахождения дискриминанта.

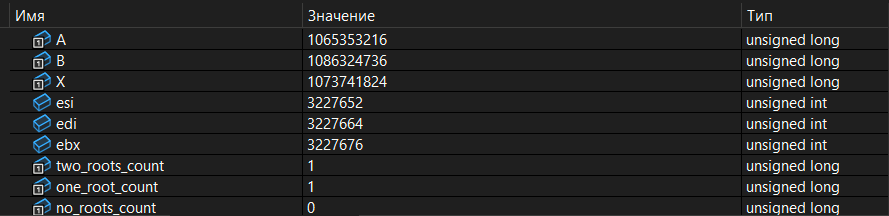


Рисунок 2 – Результаты после второго вычисления

В третьем уравнении берём целые числа 2, 2, 9. Дискриминант этих трёх значений является отрицательное число, следовательно не имеет корней. На рисунке 3 представлена переменная no\_roots\_count, предназначенная для определения дискриминанта, у которого отсутствуют корни. Также видно, что она имеет значение 1 после третьего цикла нахождения дискриминанта.

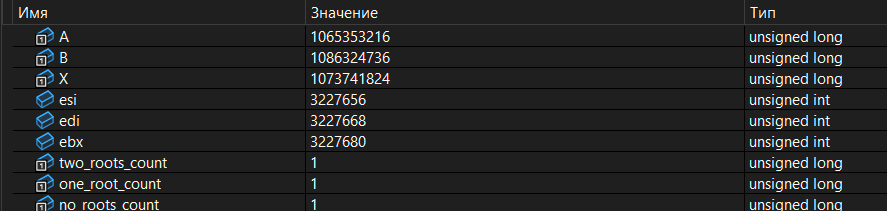


Рисунок 3 – Результаты после третьего вычисления

**Вывод:** изучили принципы выполнения арифметических команд с помощью математического сопроцессора FPU микропроцессоров с архитектурой x86.