МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА РАДИОЭЛЕКТРОННЫХ СРЕДСТВ

Отчет по дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №3

«ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКОГО СОПРОЦЕССОРА»

Вариант 3

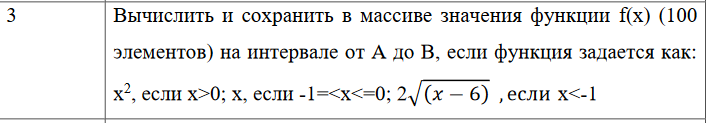
|  |  |
| --- | --- |
| Выполнил: студент группы ИНБс–3301 | Н.С. Юрьев |
|  |  |
| Проверил: доцент кафедры РЭС | М.А. Земцов |

Киров

2024

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

**Текст задания:**

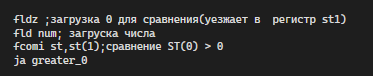


**Ход работы:**

Задаем начальное значение и значение шага для вычисления

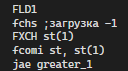


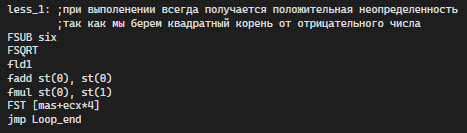
Загружаем значение числа в сопроцессор и сравниваем его с 0. Если число больше 0, то переходим к выполнению операции возведения числа в степень





Если число находится в диапазоне от -1 до 0, то переходим по к добавлению числа в массив


Если обе операции не были выполнены, то переходим в последнюю часть   


После каждого завершения каждого перехода вызывается в переход в последнюю часть программы, в которой из текущего значения вычитается шаг и итератор увеличивается на 1. Программа завершает свое выполнение после прохождения 100 циклов записи полученных данных в массив  
**Текст программы:**

.686

.model flat,stdcall

.stack 100h

.data

mas DD 100 dup(?)

num dd 5.0 ; стартовое значение

step dd 0.1 ; шаг для числа

six dd 6.0

.code

ExitProcess PROTO STDCALL :DWORD

Start:

xor eax, eax

xor ebx, ebx

xor ecx, ecx ; регистр ecx будет счетчиком до 100 для массива

Loop\_start:

finit

cmp ecx, 100

je exit ; если прошло 100 циклов, то завершаем программу

fldz ;загрузка 0 для сравнения(уезжает в регистр st1)

fld num; загруска числа

fcomi st,st(1);сравнение ST(0) > 0

ja greater\_0

FLD1

fchs ;загрузка -1

FXCH st(1)

fcomi st, st(1)

jae greater\_1

jmp less\_1

greater\_0:

FMUL ST(0),ST(0)

FST [mas+ecx\*4] ; сохраняем в массив число

jmp Loop\_end

greater\_1:

FST [mas+ecx\*4]

jmp Loop\_end

less\_1: ;при выполенении всегда получается положительная неопределенность

;так как мы берем квадратный корень от отрицательного числа

FSUB six

FSQRT

fld1

fadd st(0), st(0)

fmul st(0), st(1)

FST [mas+ecx\*4]

jmp Loop\_end

Loop\_end:

fld num

fsub step

fst num; num - 0.1

add ecx,1 ; увеличение итератора на 1

jmp Loop\_start

exit:

Invoke ExitProcess,1

End Start

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