Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Архитектура вычислительных систем

Отчёт по лабораторной работе №7-8

ПРОГРАММИРОВАНИЕ АРИФМЕТИЧЕСКОГО

СОПРОЦЕССОРА

Выполнил:

студент гр. 153503

Кончик Д. С.

Проверил:

Калиновская А.А.

Минск 2023

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc118909172)

[2 Задание 3](#_Toc118909173)

[3 Результат выполнения 5](#_Toc118909174)

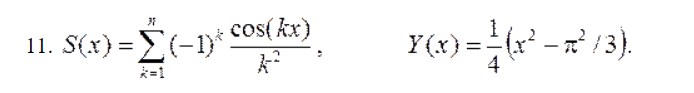
[4 Вывод 8](#_Toc118909175)

**1 ЦЕЛЬ РАБОТЫ**

Изучение архитектуры и возможностей арифметического сопроцессора, разработка программ для выполнения арифметических операций, реализация этих программ с использованием ассемблерных вставок в языках программирования, обеспечение взаимодействия программы и арифметического сопроцессора.

2 ЗАДАНИЕ

*Лабораторная работа №7.* Написать программу, находящую решение квадратного уравнения с помощью сопроцессора.

*Лабораторная работа №8.* Значение аргумента изменяется от до с шагом . Для каждого найти значения функции , суммы и число итераций , при котором достигается требуемая точность . Результат вывести в виде таблицы. Значения и вводятся с клавиатуры.

Листинг 1 ­– Файл «lab\_7.cpp»

#include <iostream>

#include <cmath>

#include <windows.h>

using namespace std;

int main()

{

double a, b, c;

cin >> a;

cin >> b;

cin >> c;

if (a == 0) {

double r = 0;

\_\_asm {

FLD b

FLD c

FDIV

FCHS

FST r

}

std::cout << "Уравнение имеет один корень: " << r << endl;

} else {

double four = 4;

double D = 0;

\_\_asm {

FLD b

FLD b

FMUL

FLD four

FLD a

FLD c

FMUL

FMUL

FSUB

FST D

}

if (D < 0) {

std::cout << "Уравнение не имеет действительных корней." << endl;

} else if (fabs(D) < 1e-15) {

double r = 0;

\_\_asm {

FLD b

FCHS

FLD a

FLD a

FADD

FDIV

FST r

}

std::cout << "Уравнение имеет один корень: " << r << endl;

} else {

double r1 = 0;

\_\_asm

{

FLD b

FCHS

FLD D

FSQRT

FSUB

FLD a

FLD a

FADD

FDIV

FST r1

}

double r2 = 0;

\_\_asm

{

FLD b

FCHS

FLD D

FSQRT

FADD

FLD a

FLD a

FADD

FDIV

FSTP r2

}

cout << "Уравнение имеет два корня: " << r1 << " и " << r2 << endl;

}

}

return 0;

}

Листинг 2 ­– Файл «lab\_8.cpp»

#include <iostream>

#include <cmath>

#include <Windows.h>

using namespace std;

double PI = acos(-1);

double Y(double x) {

double result = 0;

double three = 3;

double four = 0.25;

\_\_asm {

FLD x

FLD x

FMUL

FLD PI

FLD PI

FMUL

FLD three

FDIV

FSUB

FLD four

FMUL

FSTP result

}

return result;

}

double S(double x, int n) {

double sum = 0;

for (int k = 1; k <= n; k++) {

double item = 0;

double dk = (double)k;

\_\_asm {

FLD x

FLD dk

FMUL

FCOS

FLD dk

FLD dk

FMUL

FDIV

FSTP item

}

if (k % 2 == 0) {

\_asm {

FLD sum

FLD item

FADD

FSTP sum

}

}

else {

\_asm {

FLD sum

FLD item

FSUB

FSTP sum

}

}

}

return sum;

}

int get\_n\_from\_sub(double x, double eps) {

int n = 1;

while (fabs(Y(x) - S(x, n++)) > eps);

return n;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

double a, b, h, eps;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите h: ";

cin >> h;

cout << "Введите eps: ";

cin >> eps;

cout << endl << endl;

int iter = 1;

cout << "|------|---|------|------|---|" << endl;

cout << "| iter | x | S(x) | Y(x) | n |" << endl;

cout << "|------|---|------|------|---|" << endl;

for (double x = a; a < b; x += h) {

int n = get\_n\_from\_sub(x, eps);

cout << "| " << iter++ << " | " << x << " | " << Y(x) << " | " << S(x, n) << " | " << n << " | " << endl;

}

return 0;

}

3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

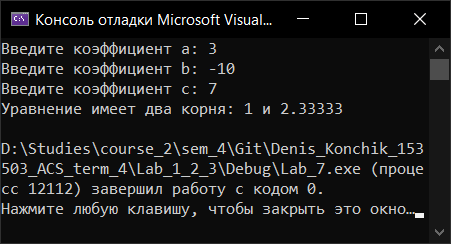


Рисунок 1 – Результат выполнения программы (работа №7)

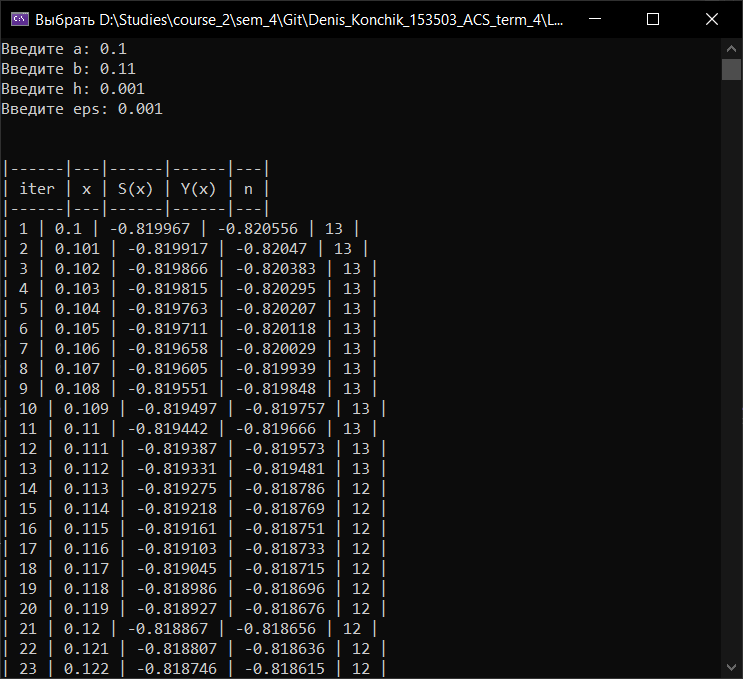


Рисунок 2 – Результат выполнения программы (работа №8)

4 ВЫВОД

В результате выполнения лабораторных работ по программированию арифметического сопроцессора FPU были получены навыки работы с ассемблерными вставками в языке C++, а также понимание принципов работы FPU.

Для выполнения заданий были использованы команды FPU, такие как FLD, FSTP, FADD, FMUL, FCOS и др. Они были применены для решения различных математических задач, включая вычисление корней квадратного уравнения, нахождение значения косинуса и др.