|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
| Расчетно-графическое задание № 4 | | |
| по дисциплине «Языки программирование и методы трансляции» | | |
| ПРОГРАММИРОВАНИЕ СОПРОЦЕССОРА | | |
|  | | |
|  | Бригада 9 | ПМ-13 исакин даниил |
|  | ПМ-13 вострецова екатерина |
| Вариант 9 |  |
|  |  |
|  |  |
|  |  |
| Преподаватель | дворецкая виктория константиновна |
|  |  |
| Новосибирск, 2024 | | |

1. **Цель работы**

Изучить архитектуру и средства программирования сопроцессора (модуля операций с плавающей точкой) на языке ассемблера и приобрести практические навыки работы с ними.

1. **Задание**

1. Изучить архитектуру и средства программирования сопроцессора на языке ассемблера.

2. Написать программу, реализующую вычисление функции в точке (вводится пользователем) в соответствии с заданным вариантом. Программа должна состоять из модулей на С++ и ассемблере, причем в модуле на С++ осуществляется ввод-вывод, а все вычисления – в модуле на ассемблере.

3. Отладить программу, убедиться в правильности ее работы на тестовых примерах.

4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, график функции, текст разработанной программы (на этапах вычислений в комментариях указывать состояние стека сопроцессора) и результаты тестирования.

5. Защитить лабораторную работу, ответив на вопросы преподавателя.

**3. Алгоритм**

1 Получить число при помощи std::cin

2) Передать число в сопроцессор

3) Вычислить tan(x), сравнить результат с машинным нулем и бесконечностью

4) Если ноль, то результат бесконечность

5) Если бесконечность, то результат 0

6) Сложить с выражением -2\*х

7) Вернуть результат

8) Вывести результат на экран

**3. Используемые средства**

**FLD** – загрузка данных

**FXCH** - Команда обмена вершины регистрового стека ST(0) с любым другим регистром стека сопроцессора ST(i)

**FLDZ** – загрузка нуля;

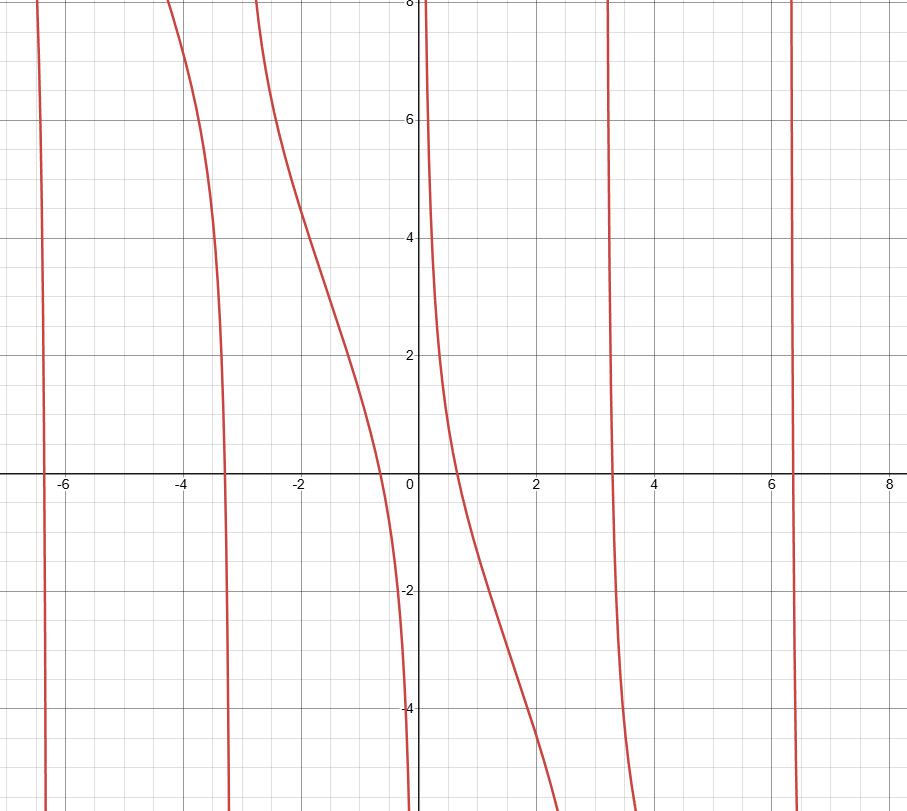
**FLD1** – загрузка единицы;

**FCOMI** - сравниваются ST(0) и ST(1)

**FPTAN** – команда вычисляет тангенс угла, находящегося в вершине стека сопроцессора. Результат возвращается в регистре ST(1), а в регистр ST(0) помещается единица. Аргументы всех этих команд должны задаваться в радианах и не могут быть больше 263 и меньше –263.

1. **Функция для реализации**

****

****

**6. Текст программы**

**Source.cpp**

#include <iostream>

#include <iomanip>

#include <cmath>

#include <stdio.h>

# define M\_PI 3.14159265358979323846 /\* pi \*/

extern "C" double f(double x);

double f\_check(double x)

{

double res = 0.0;

double arg1 = std::tan(x);

if (std::abs(arg1) <= 1e-15)

{

res = 1.0 / res;

}

else if (std::fabs(arg1) >= 1e+15)

{

res = 0.0;

}

else

{

res = 1.0/arg1;

}

return res - 2 \* x;

}

int main() {

double values[] = {0 ,M\_PI, M\_PI / 2, 3\*M\_PI/4.0, M\_PI};

for (double x : values) {

double result = f(x);

printf("f(%.5lf) = %8.5lf f control(%.5lf) = %8.5lf\n", x, f(x), x, f\_check(x));

}

return 0;

}

**Proc.asm**

.386

.model flat, C

.data

tmp\_val real8 0.0

.code

PUBLIC f

f PROC

finit

fld qword ptr [esp+4]

fptan ; ST(0) = 1.0, ST(1) = tan(x)

fxch st(1) ; ST(0) = tan(x), ST(1) = 1.0

fst tmp\_val

fld tmp\_val ; ST(0) = tan(x)

fabs ; ST(0) = |tan(x)|, ST(1) = tan(x)

fld qword ptr [epsilon\_small] ;(ST(0) = 1e-15, ST(1) = |tan(x)|, ST(2) = tan(x))

fcomi st(0), st(1)

fstp st(0)

jae tan\_is\_small

fld tmp\_val ; ST(0) = tan(x)

fabs ; ST(0) = |tan(x)|

fld qword ptr [epsilon\_large] ;(ST(0) = 1e+15, ST(1) = |tan(x)|, ST(2) = tan(x))

fcomi st(0), st(1)

fstp st(0)

jbe tan\_is\_large

fld1 ; ST(0) = 1.0

fld tmp\_val ; ST(0) = tan(x), ST(1) = 1.0

fdivp st(1), st(0) ; ST(0) = ctg(x)

jmp compute\_result

tan\_is\_small:

fld1 ; ST(0) = 1.0

fldz ; ST(0) = 0.0, ST(1) = 0.0

fdivp st(1), st(0) ; ST(0) = Infinity

jmp compute\_result

tan\_is\_large:

fldz ; ST(0) = 0.0

jmp compute\_result

compute\_result:

; Вычисляем 2 \* x:

fld qword ptr [esp+4] ; (ST(0) = x, ST(1) = ctg(x))

fld1 ;

fadd st(0), st(0) ; ST(0) = 2.0

fmulp st(1), st(0) ; ST(0) = 2 \* x, ST(1) = ctg(x)

; Вычисляем ctg(x) - 2 \* x:

fsubp st(1), st(0) ; ST(0) = ctg(x) - 2 \* x

; Возврат результата:

ret

epsilon\_small real8 1.0e-15 ; Константа для проверки малого значения

epsilon\_large real8 1.0e+15 ; Константа для проверки большого значения

f ENDP

END

**7. Тесты**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Результат | Описание |
| 1 | 0.00000 | inf | Проверка на обработку бесконечности |
| 2 | 0.78540 | -0.57080 | Корректная работа программы |
| 3 | 1.57080 | -3.14159 | Обработка нулевого значения тригонометрической функции. Должно быть в точности число пи |
| 4 | 2.35619 | -5.71239 | Стандартная работа программы |
| 5 | 3.14159 | inf | Обработка аргумента пи. Функция там обращается в бесконечность |

**8. Выводы**

1) В ходе работы научились программировать сопроцессор для работы с плавающей точкой

2) Применили на практике низкоуровневые операции для расчёта тригонометрических функций.

3) Разобрались с обработкой исключительных ситуаций. Поняли как происходит обработка краевых значений с применением сопроцессора для работы чисел с плавающей точкой.