МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №8
по дисциплине «Организация ЭВМ и систем»
Тема: Обработка вещественных чисел.
Программирование математического
сопроцессора.

Студентка гр. 9383 _____ Чебесова И.Д,

Преподаватель _____ Ефремов М.А.

Санкт-
Петербург 2020

Цель работы.

Познакомиться с обработкой вещественных чисел в ассемблере и командами сопроцессора. Написать программу на ЯВУ (C++) с ассемблерной вставкой для расчета определенной функции.

Задание.

Вариант 6.

Разработать подпрограмму на языке Ассемблера, обеспечивающую вычисление заданной математической функции с использованием математического сопроцессора. Подпрограмма должна вызываться из головной программы, разработанной на языке С. При этом должны быть обеспечены заданный способ вызова и обмен параметрами. Альтернативный вариант реализации: разработать на языке Ассемблера фрагмент программы, обеспечивающий вычисление заданной математической функции с использованием математического сопроцессора, который включается по принципу inline в программу, разработанную на языке С. Возможный пример требуемой разработки программы для случая вычисления функции fmod(x,y)=x mod y приведен ниже в приложении 1. Выполнить трансляцию программы с подготовкой ее ассемблерной версии и отладочной информации. Для выбранного контрольного набора исходных данных прогнать программу под управлением отладчика. При этом для каждой команды сопроцессора следует фиксировать содержимое используемых ячеек памяти, регистров ЦП и численных регистров сопроцессора до и после выполнения этой команды. Проверить корректность выполнения вычислений для нескольких наборов исходных данных. Примечание: При разработке программ вычисления заданной функции можно воспользоваться версиями соответствующих программ, взятыми из каталога FLOW_PNT.

ВАРИАНТ 6.

* function

Name Acos - compute acos

Usage double Acos (double *xP);

Prototype in math.h

Description Computes acos of the number pointed to by xP.

Arguments to acos must be in the range -1 to 1, acos returns a value in the range 0 to pi.

Use the trig identities acos (x) = atan (sqrt (1-x^2) / x) */

Ход работы.

В ходе работы была разработана программа на языке C++ с ассемблерной вставкой, которая считает арккосинус переданного числа x по формуле atan(sqrt (1-x^2) / x).

В основной функции происходит считывание входного x и проверка его на корректность: от [-1;1]. Далее происходит расчет библиотечной функции арккосинуса для проверки вычислений ассемблера и непосредственно сами вычисления.

Команды сопроцессора, которые используются в программе:

fld — загрузка вещественного операнда в вершину стека

fmul — вещественное умножение, если не введены операнды, то перемножаются два верхних операнда стека (st(0)*st(1))

fld1 — загрузка константы (1) в вершину стека

fxch — меняет между собой введенный операнд стека, например st(1) и вершину st(0)

fsub — вычитает из одного операнда дрогой, по умолчанию st(1)-st(0)

fsqrt — вычисление квадратного корня

fpatan - вычисляет арктангенс угла с аргументами в st(0)=знаменатель и st(1)=числитель

fstp - сохранение вершины стека в память с выталкиванием

Исходный код см. в приложении А.

Пример работы программы.

Входные данные:
Enter x in range [-1; 1] to calculate arccos(x): -1

Выходные данные:
C++ arccos = 3.14159
Assembly arccos = 3.14159

Входные данные:
Enter x in range [-1; 1] to calculate arccos(x): 1

Выходные данные:
C++ arccos = 0
Assembly arccos = 0

Входные данные:
Enter x in range [-1; 1] to calculate arccos(x): 0

Выходные данные:
C++ arccos = 1.5708
Assembly arccos = 1.5708

Входные данные:
Enter x in range [-1; 1] to calculate arccos(x): 0.5

Выходные данные:
C++ arccos = 1.0472
Assembly arccos = 1.0472

Выводы.

Было произведено знакомство с обработкой вещественных чисел в ассемблере и командами сопроцессора. Была написана программа на ЯВУ (C++) с ассемблерной вставкой для расчета определенной функции.

ПРИЛОЖЕНИЕ А.

ИСХОДНЫЙ КОД ПРОГРАММЫ.

Название файла: Source1.cpp

```
#include <iostream>
#include <math.h>

using namespace std;

double Acos(double* xP)
{
     double x = *xP;
     double result = 0;
     _asm
     {
           fld x;       //st(0) = x
           fld st(0);   //st(0) = x; st(1) = x
           fmul;        //st(0) = x^2;
           fld1;        //st(0) = 1; st(1) = x^2;
           fxch st(1);  //st(0) = x^2; st(1) = 1;
           fsub;        //st(0) = 1-x^2
           fsqrt;       //st(0) = sqrt(1-x^2)
           fld x;       //st(0) = x; st(1) = sqrt(1-x^2)
           fpatan;      //st(0) = atan(sqrt(1-x^2)/x)
           fstp result;
     }
     return result;
}



int main()
{
     double x;

     cout << "Enter x in range [-1; 1] to calculate arccos(x): ";
     cin >> x;
     if ((x < -1) || (x > 1))
     {
           cout << "x is out of range!" << '\n';
     }

     cout << "C++ arccos = " << acos(x) << '\n';
     cout << "Assembly arccos = " << Acos(&x) << '\n';

     return 0;
}
```