Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

за 4 семестр

По дисциплине: «КCиС»

Тема: «Архитектура и программирование сопроцессора»

Выполнила:

Студентка 2 курса

Группы ПО-6 (2)

Данилюк Д. Б.

Проверил:

Бойко Д.О.

Брест, 2022

Лабораторная работа №4

Архитектура и программирование сопроцессора

**Вариант 11**

**Цель:** научиться работать с сопроцессором с использованием целочисленных команд.

**Содержание работы:**

Изучить теоретические сведения (ЗНАТЬ: архитектуру, форматы чисел, типовые команды, структуру программ). Составить и отладить программу на языке ассемблера для вычисления значения функции, используя регистры сопроцессора (таблица 2.2). Написать вариант программы с использованием **целочисленных** команд сопроцессора.

**Вариант задания:**

**Код программы:**

#include<iostream>

using namespace std;

int main() {

const int ONE = 1, FIVE = 5, FOUR = 4, MINUSFOUR = -4, ELEVEN = 11;

int x;

int y;

cout << "Print x and y (%d %d): ";

cin >> x >> y;

int temp;

int u;

\_asm {

mov eax, [x]

add eax, [y]

finit //инициализация сопроцессора: инициализация управляющих регистров сопроцессора определнными значениями

//приведение сопроцессора в начальное состояние

cmp eax, 11

jg met1 //если больше 11, то переход на метку p1

cmp eax, -4

jl met2 //если меньше -4, то переход на метку p2

jmp met3 //безусловный переход

met1:

//jmp endasm //безусловный переход

fild[y] //загрузка значения y в st(0)

fimul[y] // st(0)=y\*y

fistp [temp] // temp = y\*y

fild[y] //загрузка значения y в st(0)

fimul [y] // st(0)=y\*y

fiadd [temp] // st(0)=y\*y + y\*y

fistp[temp] // temp = 2\*y\*y

fild[y] //загрузка значения y в st(0)

fimul [y] // st(0)=y\*y

fiadd [temp] // st(0)=2y\*y + y\*y

fiadd [ONE] // st(0)=3\*y\*y+1

fistp [temp] // temp = 3\*y\*y+1

fild[x] //загрузка значения x в st(0)

fimul[x] // st(0)=x\*x

fiadd [temp] // st(0)=x\*x+3\*y\*y+1

fistp [temp] // temp = x\*x+3\*y\*y+1

fild[x] //загрузка значения x в st(0)

fiadd [x] // st(0)=x + x

fiadd [y] // st(0)=x + x + y

fidiv [temp] // st(0)=(x + x + y)/(x\*x+3\*y\*y+1)

fistp [u] //u=(x + x + y)/(x\*x+3\*y\*y+1)

jmp endasm //безусловный переход

met2:

fild [y] //загрузка значения y в st(0)

fimul [y] // st(0)=y\*y

fistp [temp] // temp = y\*y

fild[y] //загрузка значения y в st(0)

fimul [y] // st(0)=y\*y

fiadd [temp] // st(0)=y\*y + y\*y

fiadd [ONE] //st(0)=y\*y + y\*y+1

fistp[temp] // temp = 2\*y\*y+1

fild[x] //загрузка значения x в st(0)

fimul[x] // st(0)=x\*x

fiadd [temp] // st(0)=x\*x+2\*y\*y+1

fistp [temp] // temp = x\*x+2\*y\*y+1

fild[x] //загрузка значения x в st(0)

fimul [FOUR] // st(0)=x\*4

fimul [y] // st(0)=x\*4\*y

fidiv [temp] // st(0)=(x\*4\*y)/(x\*x+2\*y\*y+1)

fistp [u] //u=(x\*x+2\*y\*y+1)

jmp endasm //безусловный переход

met3:

fild [y] //загрузка значения y в st(0)

fimul [y] // st(0)=y\*y

fistp [temp] // temp = y\*y

fild[y] //загрузка значения y в st(0)

fimul [y] // st(0)=y\*y

fiadd [temp] // st(0)=y\*y + y\*y

fistp[temp] // temp = 2\*y\*y

fild[x] //загрузка значения x в st(0)

fimul[x] // st(0)=x\*x

fimul [FIVE] // st(0)=x\*x\*5

fiadd [temp] // st(0)=x\*x\*5+2\*y\*y

fistp [u] //u=x\*x\*5+2\*y\*y

jmp endasm //безусловный переход

endasm:

}

cout << u;

}

**Результат выполнения:**







**Вывод:** научилась работать с сопроцессором с использованием целочисленных команд.