Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4, 5

за 2 семестр

По дисциплине: «КCиС»

Тема: «Архитектура и программирование сопроцессора»

Выполнил:

Студент 2 курса

Группы ПО-6(2)

Осьмушников А.Ю.

Проверил:

Бойко Д.О.

Брест, 2022

Лабораторная работа №4, 5

Архитектура и программирование сопроцессора

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

**Содержание работы:**

Изучить теоретические сведения. Составить и отладить программу на языке ассемблера для вычисления значения функции, используя регистры сопроцессора (таблица 2.2). Написать вариант программы с использованием целочисленных и вещественных команд команд сопроцессора.

**Вариант задания:**

**Целочисленное:**

**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;

}

**Результаты тестирования:**







**Вещественное:**

**Код программы:**

#include<iostream>

using namespace std;

int main() {

const float ONE = 1, TWO=2, TREE=3, FIVE = 5, FOUR = 4, MINUSFOUR = -4, ELEVEN = 11;

float x, y, u;

cout << "Print x and y: ";

cin >> x >> y;

\_asm {

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

fld[x] //загрузка значение x в st(0)

fadd[y]

fcom[MINUSFOUR] //сравниваем st(0) c -4

fstsw ax // сохранение swr в регистре ax

sahf //запись swr->ax-> регистр флагов

jb met2 //если меньше, то переход met2

fcom[ELEVEN] //сравниваем st(0) c 11

fstsw ax // сохранение swr в регистре ax

sahf //запись swr->ax-> регистр флагов

jbe met3 //если меньше или равно, то переход в met3

jmp met1 //безусловный переход в met1

met1:

fld[y] //загрузка значения y в st(0)

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

fmul[TREE] //3\*y\*y

fadd[ONE]//3\*y\*y+1

fld[x] //загрузка значения x в st(0)

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

fadd //x\*x+3\*y\*y+1

fld[x] //загрузка значения x в st(0)

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

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

fdivr // st(0)=(x+x+y)/(x\*x+3\*y\*y+1)

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

jmp endasm //безусловный переход

met2 :

fld[y] //загрузка значения y в st(0)

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

fmul[TWO] //y\*y\*2

fadd[ONE] //st(0)=2\*y\*y+1

fld[x] //загрузка значения x в st(0)

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

fadd //x\*x+2\*y\*y+1

fld[x] //загрузка значения x в st(0)

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

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

fdivr // st(0)=(x\*4\*y)/(x\*x+2\*y\*y+1)

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

jmp endasm //безусловный переход

met3 :

fld[y] //загрузка значения y в st(0)

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

fmul[TWO] // st(0)=2\*y\*y

fld[x] //загрузка значения x в st(0)

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

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

fadd // st(0)=x\*x\*5+2\*y\*y

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

jmp endasm //безусловный переход

endasm :

}

cout << u;

}

**Результаты тестирования:**







**Вывод:** в ходе данной лабораторной работы научилась работать с сопроцессором с использованием целочисленных и вещественных команд.