Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4, 5

за 2 семестр

По дисциплине: «КСиС»

Выполнил:

Студент 2 курса

Группы ПО-7(1)

Борисюк Р.С.

Проверил:

Бойко Д.О.

2022

**Лабораторная работа #4, 5**

**АРХИТЕКТУРА И ПРОГРАММИРОВАНИЕ СОПРОЦЕССОРА**

**Цель работы:** изучить архитектуру и программирование сопроцессора.

**Лабораторная работа #4** - с использованием целочисленных команд сопроцессора;

**Лабораторная работа #5 -** с использованием вещественных команд сопроцессора

**Вариант 3**

**Вариант задания:**

**Код программы:**

**4.asm:**

section .text

global \_start

p1: ;вычисляем значение при x + y > 7

fild dword [x] ;st(0) = x

fimul dword [three] ;st(0) = 3 \* x

fild dword [y] ;st(0) = y, st(1) = 3 \* x

fimul dword [four] ;st(0) = 4 \* y, st(1) = 3 \* x

fadd st1 ;st(0) = 3 \* x + 4 \* y

fiadd dword [three] ;st(0) = 3 \* x + 4 \* y + 3

fist dword [u] ;u = st(0)

ret

p2: ;вычисляем значение при x + y < -5

fild dword [x] ;st(0) = x

fimul dword [x] ;st(0) = x \* x

fimul dword [two] ;st(0) = 2 \* x \* x

fild dword [y] ;st(0) = y, st(1) = 2 \* x \* x

fimul dword [three] ;st(0) = 3 \* y, st(1) = 2 \* x \* x

fsub st1 ;st(0) = 3 \* y - 2 \* x \* x

fistp dword [u] ;u = st(0), st(0) = st (1)

fstp st0 ;очистка

fild dword [y] ;st(0) = y

fimul dword [y] ;st(0) = y \* y

fiadd dword [four] ;st(0) = y \* y + 4

fild dword [u] ;st(0) = 3 \* y - 2 \* x \* x, st(1) = y \* y + 4

fdiv st1 ;st(0) = (3 \* y - 2 \* x \* x) / (y \* y + 4)

fistp dword [u] ;u = st(0)

ret

p3: ;вычисляем значение при -5 <= x + y <= 7

fild dword [x] ;st(0) = x

fimul dword [x] ;st(0) = x \* x

fimul dword [three] ;st(0) = 3 \* x \* x

fild dword [y] ;st(0) = y, st(1) = 3 \* x \* x

fmul st0 ;st(0) = y \* y, st(1) = 3 \* x \* x

fimul dword [four] ;st(0) = 4 \* y \* y, st(1) = 3 \* x \* x

fxch st1 ;st(0) = 3 \* x \* x, st(1) = 4 \* y \* y

fsub st1 ;st(0) = 3 \* x \* x - 4 \* y \* y

fisub dword [three] ;st(0) = 3 \* x \* x - 4 \* y \* y - 3

fist dword [u] ;u = st(0)

ret

\_start:

finit ;приведение сопроцессора в начальное состояние

fild dword [x] ;st(0) = x

fiadd dword [y] ;st(0) = x + y

ficom dword [seven] ;сравниваем st(0) c 7

fstsw ax ;сохранение swr в регистре ax

sahf ;запись swr->ax-> регистр флагов

jnc met1 ;если x + y > 7 переход на метку1

ficom dword [sFive] ;сравниваем st(0) c 7

ftst

fstsw ax ;сохранение swr в регистре ax

sahf ;запись swr->ax-> регистр флагов

jc met2 ;если x + y < -5 переход на метку2

call p3 ;вычисляем значение при -5 <= x + y <=7

jmp exit

met1: call p1 ;вычисляем значение при a+b<5

jmp exit

met2: call p2 ;вычисляем значение при a+b=5

exit: ;вызов ядра

mov eax,1 ;номер системного вызова (sys\_exit)

int 0x80 ;вызов ядра

section .data

x dd -2

y dd 7

u dd 1

seven dd 7

sFive dd -5

three dd 3

four dd 4

two dd 2

**5.asm:**

section .text

global \_start

p1: ;вычисляем значение при x + y > 7

fld dword [x] ;st(0) = x

fmul dword [three] ;st(0) = 3 \* x

fld dword [y] ;st(0) = y, st(1) = 3 \* x

fmul dword [four] ;st(0) = 4 \* y, st(1) = 3 \* x

fadd st1 ;st(0) = 3 \* x + 4 \* y

fadd dword [three] ;st(0) = 3 \* x + 4 \* y + 3

fstp dword [u] ;u = st(0)

ret

p2: ;вычисляем значение при x + y < -5

fld dword [x] ;st(0) = x

fmul st0 ;st(0) = x \* x

fmul dword [two] ;st(0) = 2 \* x \* x

fld dword [y] ;st(0) = y, st(1) = 2 \* x \* x

fmul dword [three] ;st(0) = 3 \* y, st(1) = 2 \* x \* x

fsub st1 ;st(0) = 3 \* y - 2 \* x \* x

fstp dword [u] ;u = st(0), st(0) = st (1)

fstp st0 ;очистка

fld dword [y] ;st(0) = y

fmul st0 ;st(0) = y \* y

fadd dword [four] ;st(0) = y \* y + 4

fld dword [u] ;st(0) = 3 \* y - 2 \* x \* x, st(1) = y \* y + 4

fdiv st1 ;st(0) = (3 \* y - 2 \* x \* x) / (y \* y + 4)

fstp dword [u] ;u = st(0)

ret

p3: ;вычисляем значение при -5 <= x + y <= 7

fld dword [x] ;st(0) = x

fmul st0 ;st(0) = x \* x

fmul dword[three] ;st(0) = 3 \* x \* x

fld dword [y] ;st(0) = y, st(1) = 3 \* x \* x

fmul st0 ;st(0) = y \* y, st(1) = 3 \* x \* x

fmul dword [four] ;st(0) = 4 \* y \* y, st(1) = 3 \* x \* x

fxch st1 ;st(0) = 3 \* x \* x, st(1) = 4 \* y \* y

fsub st1 ;st(0) = 3 \* x \* x - 4 \* y \* y

fsub dword [three] ;st(0) = 3 \* x \* x - 4 \* y \* y - 3

fstp dword [u] ;u = st(0)

ret

\_start:

finit ;приведение сопроцессора в начальное состояние

fld dword [x] ;st(0) = x

fadd dword [y] ;st(0) = x + y

fcom dword [seven] ;сравниваем st(0) c 7

fstsw ax ;сохранение swr в регистре ax

sahf ;запись swr->ax-> регистр флагов

jnc met1 ;если x + y > 7 переход на метку1

fcom dword [sFive] ;сравниваем st(0) c -5

ftst

fstsw ax ;сохранение swr в регистре ax

sahf ;запись swr->ax-> регистр флагов

jc met2 ;если x + y < -5 переход на метку2

call p3 ;вычисляем значение при -5 <= x + y <=7

jmp exit

met1: call p1 ;вычисляем значение при a+b<5

jmp exit

met2: call p2 ;вычисляем значение при a+b=5

exit: ;вызов ядра

mov eax,1 ;номер системного вызова (sys\_exit)

int 0x80 ;вызов ядра

section .data

x dd -2.0

y dd 7.0

u dd 1.0

seven dd 7.0

sFive dd -5.0

three dd 3.0

four dd 4.0

two dd 2.0

**Тесты:**

**Исходная данные:**

1. *x* = 5, *y* = 6

**Ожидаемый результат:** 4: 42, 5: 42.0

1. *x* = -2, *y* = -3

**Ожидаемый результат:** 4: -1, 5: 1.3

1. *x* = -2, *y* = 7

**Ожидаемый результат:** 4: -187, 5: -187.0

**Результат программы:**

1. 4: , 5: 
2. 4: , 5: 
3. 4: , 5: 

**Вывод:** изучил архитектуру и программирование сопрацессора.