

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Інститут комп'ютерних наук та інформаційних технологій
Кафедра програмного забезпечення**



ЗВІТ

**до лабораторної роботи №6
на тему: «Програмування арифметичного співпроцесора мікропроцесорів x86»
з дисципліни: «Архітектура комп'ютера»**

Лектор:

доц. кафедри ПЗ
Крук О. Г.

Виконав:

ст. гр. ПЗ-22
Чаус О. М.

Прийняв:

доц. кафедри ПЗ
Крук О. Г.

« ____ » _____ 2022 р.

Σ = ____

Тема роботи: Програмування арифметичного співпроцесора мікропроцесорів x86

Мета роботи: розвинути навички складання програми для арифметичного співпроцесора мовою асемблера для обчислення математичного виразу, відтранслявати і виконати в режимі відлагодження програму, складену відповідно до свого варіанту, обчислити заданий вираз в програмі мовою C та порівняти результати.

Варіант 27		
27	$\frac{4/a - c * a + \sqrt{23 * c - 4}}{8.4 * c - \frac{d}{9} + 13 * a}$	a = 8.3 c=7.1 d=1.2

Теоретичні відомості

Арифметичний процесор або співпроцесор - це цифровий пристрій, призначений для апаратного виконання арифметичних операцій над дійсними числами або числами з рухомою/плаваючою комою. Наявність співпроцесора дозволяє значно прискорити роботу програм, що виконують обчислення з високою точністю, тригонометричні розрахунки та опрацювання інформації, яка повинна бути подана у вигляді дійсних чисел. В перших моделях мікропроцесорів Intel співпроцесора не було, він виготовлявся у вигляді окремої інтегральної мікросхеми і входив в склад комп'ютерів як опція. Починаючи з моделі i486DX співпроцесор розміщується на тому ж кристалі, що і основний процесор.

Співпроцесор має вісім 80-розрядних регістрів даних R0 ... R7 для зберігання чисел з плаваючою комою, організованих у вигляді кільцевого стека. Номер регістра, який на даний момент перебуває на вершині стека, вказується в 3-бітовому полі TOP, що міститься в слові стану співпроцесора. При написанні програм, в яких використовуються команди з плаваючою комою, до вершини стека можна звернутися за допомогою операнда ST(0) (або просто ST). В командах можна також використовувати відносні до вершини стека операнди ST(1) ... ST(7). Абсолютні імена регістрів типу R0, R1, ... R7 використовувати не можна.

При виконанні команд з плаваючою комою їх операнди зберігаються в десятибайтових регістрах у розширеному форматі з подвійною точністю. При збереженні результату арифметичної операції в пам'яті співпроцесор автоматично перетворює його з розширеного формату в ціле або довге ціле число, а також в коротке або довге дійсне число.

Основний процесор і співпроцесор можуть обмінюватися значеннями з плаваючою комою тільки через оперативну пам'ять. Тому перед викликом команди співпроцесора її операнд завжди повинен міститися в пам'яті. При цьому співпроцесор завантажує число з пам'яті в свій стек регістрів, виконує над ним арифметичну операцію і результат зберігає в оперативну пам'ять.

Мнемоніки команд з плаваючою комою завжди починаються з літери F/f, щоб їх можна було відрізнити від інших команд основного процесора. Друга літера в мнемоніці (зазвичай це B/b або I/i) визначає спосіб інтерпретації операнда, що міститься в пам'яті. Літера B свідчить про те, що оператор поданий в двійково-десятковому коді (Binary-Coded Decimal, або BCD). Літера I говорить про те, що оператор поданий у вигляді цілочислового значення. Якщо ці літери не вказані, то вважається, що оператор міститься в пам'яті в одному з форматів чисел із плаваючою комою.

Хід роботи

1. Склад програму обчислення виразу за допомогою команд співпроцесора для WINDOWS.

Текст програми:

27	$\frac{4/a - c * a + \sqrt{23 * c - 4}}{8.4 * c - \frac{d}{9} + 13 * a}$	a = 8.3 c=7.1 d=1.2
----	--	---------------------

.686

.model flat,stdcall

.stack

.data

A REAL4 8.3

B REAL4 7.1

D REAL4 1.2

N1 REAL4 4.0

N2 REAL4 23.0

N3 REAL4 4.0

N4 REAL4 8.4

N5 REAL4 9.0

N6 REAL4 13.0

TTEST REAL4 8.52

TOP REAL4 ?

BOT REAL4 ?

RES REAL4 ?

.code

main:

 finit

 ;top

 fld N1

 fdiv A

 fld N2

 fmul B

 fsub N3

 fsqrt

 fld B

 fmul A

 fsubp

 faddp

 fst TOP

 ;bottom

 fld N4

 fmul B

 fld N6

 fmul A

 fld D

 fdiv N5

 fsubp

 faddp

 fst BOT

 fld TOP

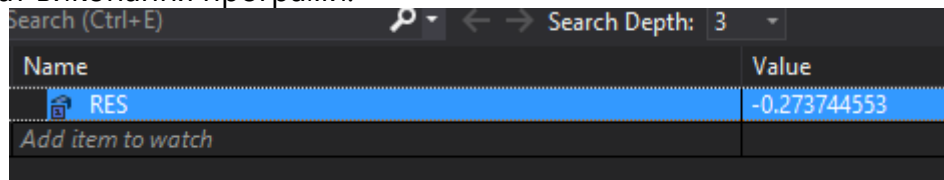
 fdiv BOT

 fst RES

 RET

END main

Результат виконання програми:



Name	Value
RES	-0.273744553
<i>Add item to watch</i>	

2. Для перевірки результати програми, створив програму для обчислення виразу на мові C.

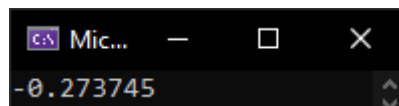
Текст програми:

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

float calculate(const float A, const float B, const float D) {
    const float N1 = 4;
    const float N2 = 23;
    const float N3 = 4;
    const float N4 = 8.4;
    const float N5 = 9;
    const float N6 = 13;
    return (N1 / A - B * A + sqrt(N2 * B - N3)) /
           (N4 * B - D / N5 + N6 * A);
}

int main() {
    printf("%f\n", calculate(8.3, 7.1, 1.2));
}
```

Результат виконання:



```
C:\> Mic...
-0.273745
```

Отже, програма на мові асемблера написана правильно.

Висновки: на цій лабораторній роботі я розвинув навички складання програми для арифметичного співпроцесора мовою асемблера для обчислення математичного виразу, відтранслявав і виконати в режимі відлагодження програму, складену відповідно до свого варіанту, обчислив заданий вираз в програмі мовою C та порівняв результати.