**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний технічний університет України**

**«Київський Політехнічний Інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №11**

*з дисципліни «Системне програмування – 1»*

*на тему: «Програмування команд SSE у модулях на асемблері»*

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-44

*Барабаш Т.А.*

**Перевірив:**

Старший викладач

*Порєв В. М.*

**Київ – 2016**

**ЛАБОРАТОРНА РОБОТА №11**

*Програмування команд SSE у модулях на асемблері*

**Мета:** Навчитися програмувати модулі на асемблері, у яких містяться команди SSE, команди x87 FPU, а також використовувати такі модулі у проектах C++.

1. **Завдання**
2. Створити проект Visual C++ Win32 з ім'ям Lab11.
3. Написати на асемблері процедуру обчислення скалярного добутку двох векторів із використанням команд SSE. Ім'я процедури: MyDotProduct\_SSE. Процедуру оформити у окремому модулі і записати файли vectsse.asm, vectsse.h. Додати файл vectsse.asm у проект.
4. Запрограмувати на асемблері процедуру обчислення скалярного добутку двох векторів на основі команд x87 FPU без використання команд SSE. Ім'я процедури: MyDotProduct\_FPU. Процедуру оформити у окремому модулі і записати файли vectfpu.asm, vectfpu.h. Додати файл vectfpu.asm у проект.
5. Запрограмувати на C++ обчислення скалярного добутку тих самих векторів як звичайну функцію C++ з ім'ям MyDotProduct, яка приймає значення двох масивів і записує результат у числову перемінну (будь-яка оптимізація при компіляції повинна бути відсутня).
6. Зробити меню для вікна програми так, щоб користувач програми мав можливість викликати процедури на асемблері MyDotProduct\_SSE, MyDotProduct\_FPU з модулів vectsse, vectfpu, а також функцію MyDotProduct.
7. Запрограмувати вивід результатів обчислень та виміри часу виконання скалярного добутку для трьох варіантів реалізації.
8. Отримати дизасемблерний текст функції C++ MyDotProduct. Проаналізувати код дизасемблеру, порівняти з кодом на асемблері процедури MyDotProduct\_FPU.
9. Зробити висновки щодо використання модулів на асемблері у програмах на мові C++ .

**Кількість елементів векторів A та B має бути N =40 \* 3 = 120.**

**ІІ. Код програми**

.586

.model flat, c

.data

.code

MyDotProduct\_FPU proc dest:DWORD, A:DWORD, B:DWORD, N:DWORD

mov eax, A

mov ebx, B

mov edx, dest

mov ecx, N

dec ecx

fldz

@cycle:

fld dword ptr[eax+4\*ecx]

fmul dword ptr[ebx+4\*ecx]

faddp st(1), st(0)

dec ecx

cmp ecx, 0

jge @cycle

fstp dword ptr[edx]

ret

MyDotProduct\_FPU endp

End

.686

.xmm

.model flat, C

include longop.inc

.data

.code

MyDotProduct\_SSE proc dest:DWORD, A:DWORD, B:DWORD, N:DWORD

mov eax, A

mov ebx, B

mov edx, dest

mov ecx, N

sub ecx, 4

xorps xmm7, xmm7

@cycle:

movaps xmm0, [eax+4\*ecx]

movaps xmm1, [ebx+4\*ecx]

mulps xmm0, xmm1

haddps xmm0, xmm0

haddps xmm0, xmm0

addps xmm7, xmm0

sub ecx, 4

cmp ecx, 0

jge @cycle

movss dword ptr [edx], xmm7

ret

MyDotProduct\_SSE endp

end

\_\_declspec(align(16)) float oA[120];

\_\_declspec(align(16)) float oB[120];

\_\_declspec(align(16)) float oC[120];

\_\_declspec(align(16)) float oD[120];

\_\_declspec(align(16)) char TextBuf[120];

void prepare() {

for (long i = 0; i<120; i++)

{

oA[i] = 0.555 + i;

oB[i] = 0.555\*pow(-1.0, i);

oC[i] = 0.0;

oD[i] = 0.0;

TextBuf[i] = 0;

}

}

void vectorSSE(HWND hWnd) {

prepare();

SYSTEMTIME st;

long tst, ten;

GetLocalTime(&st);

tst = 60000 \* (long)st.wMinute

+ 1000 \* (long)st.wSecond

+ (long)st.wMilliseconds;

for (long i = 0; i<1000000; i++)

{

MyDotProduct\_SSE(oD, oA, oB, 120);

}

GetLocalTime(&st);

ten = 60000 \* (long)st.wMinute

+ 1000 \* (long)st.wSecond

+ (long)st.wMilliseconds - tst;

sprintf\_s(TextBuf, "Скалярний добуток = %f\nЧас виконання = %ld мс", oD[0], ten);

MessageBox(hWnd, TextBuf, "SSE", MB\_OK);

}

void vectorFPU(HWND hWnd) {

prepare();

SYSTEMTIME st;

long tst, ten;

GetLocalTime(&st);

tst = 60000 \* (long)st.wMinute

+ 1000 \* (long)st.wSecond

+ (long)st.wMilliseconds;

for (long i = 0; i<1000000; i++)

{

MyDotProduct\_FPU(oD, oA, oB, 120);

}

GetLocalTime(&st);

ten = 60000 \* (long)st.wMinute

+ 1000 \* (long)st.wSecond

+ (long)st.wMilliseconds - tst;

sprintf\_s(TextBuf, "Скалярний добуток = %f\nЧас виконання = %ld мс", oD[0], ten);

MessageBox(hWnd, TextBuf, "FPU", MB\_OK);

}

void vectorCPP(HWND hWnd) {

prepare();

SYSTEMTIME st;

long tst, ten;

GetLocalTime(&st);

tst = 60000 \* (long)st.wMinute

+ 1000 \* (long)st.wSecond

+ (long)st.wMilliseconds;

for (long i = 0; i<1000000; i++)

{

MyDotProduct(oD, oA, oB, 120);

}

GetLocalTime(&st);

ten = 60000 \* (long)st.wMinute

+ 1000 \* (long)st.wSecond

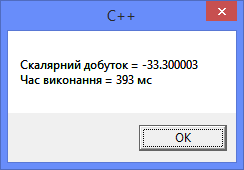
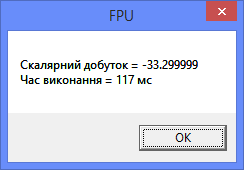
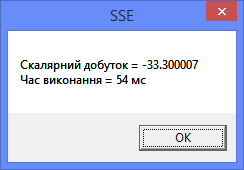
+ (long)st.wMilliseconds - tst;

sprintf\_s(TextBuf, "Скалярний добуток = %f\nЧас виконання = %ld мс", oD[0], ten);

MessageBox(hWnd, TextBuf, "C++", MB\_OK);

}

**III. Результат**



**ІV. Висновок**

У ході виконання лабораторної роботи було закріплено основні навички в операціях роботи із форматом SSE та доведено, що швидкість виконання операцій на мові асемблер є найбільшою. Програмування на мові високого рівня (C++) виявилося найбільш повільним, найшвидшим виявилося застосування команд формату SSE, що у приблизно 2 рази швидше за формат FPU та у 6 разів програмування на C++.