УО БГУИР

Кафедра ЭВМ

Отчет по лабораторной работе

Тема: "Команды MMX/XMM"

Выполнил:

студент группы 650503 Юревич А. С.

Проверил:

к.т.н., доцент Одинец Д. Н.

Минск 2018

1. Постановка задачи

Вариант 1: Выполнить вычитание матриц.

Цель работы: Изучить расширение системы команд MMX процессоров Intel.

Создать консольное приложение, которое выполняет вычитание матриц тремя способами:

1) с использованием команд MMX

2) на ассемблере, без использования команд MMX

3) на языке Си

После вычислений должны быть выведены время выполнения и результат для каждого случая.

Значения элементов матриц генерируются приложением (не вводятся с клавиатуры). Вычисления производятся несколько (от 1 млн) раз. Размер матриц кратен количеству элементов в регистре ММХ.

1. Алгоритм решения задачи

Для того, чтобы реализовать поставленные задачи необходимо:

1. Создать консольное меню взаимодействия с пользователем
2. Реализовать проверку введённых значений на соответствие допустимым
3. Реализовать вычисления на языках Си, Ассемблера, Ассемблера с использованием MMX
4. Произвести замер производительности вычислений с помощью реализованных функций
5. Листинг программы

#include <iostream>

#include <random>

#include <time.h>

const int COUNT\_CYCLE = 5000000; //Количество итераций цикла

const int SIZE = 4; //Размер матриц

const int RAND = 10; //Область рандома

using namespace std;

/\*Вывод матрицы на экран\*/

void output(int ms[SIZE][SIZE])

{

for (int i = 0; i < SIZE; i++)

{

for (int j = 0; j < SIZE; j++)

{

cout << ms[i][j] << ' ';

}

cout << endl;

}

}

void null(int ms[SIZE][SIZE])

{

for (int i = 0; i < SIZE; i++)

{

for (int j = 0; j < SIZE; j++)

{

ms[i][j]=0;

}

}

}

/\*Заполнение матрицы случайными числами\*/

void rand(int ms[SIZE][SIZE])

{

for (int i = 0; i < SIZE; i++)

{

for (int j = 0; j < SIZE; j++)

{

ms[i][j] = rand() % RAND;

}

}

}

int main()

{

srand(time(NULL));

clock\_t start;

clock\_t end;

int ms1[SIZE][SIZE];

int ms2[SIZE][SIZE];

int result\_ms[SIZE][SIZE] = { 0 };

rand(ms1);

rand(ms2);

cout << "Number of iterations: " << COUNT\_CYCLE << endl;

cout << "Random area: " << RAND << endl << endl;

/\*Pure C\*/

start = clock();

for (int i = 0; i < COUNT\_CYCLE; i++)

{

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

result\_ms[i][j] = ms1[i][j] - ms2[i][j];

}

}

}

end = clock();

//output(result\_ms);

cout << "Pure C: " << (double)(end - start) / CLOCKS\_PER\_SEC << endl;

null(result\_ms);

/\*Asm\*/

start = clock();

for (int i = 0; i < COUNT\_CYCLE; i++)

{

\_asm {

mov esi, 0 //зануляем индекс

mov eax, 0 //значение разности

mov ebx, 16 //Массив 4х4

cycleAsm:

mov eax, ms1[esi]

sub eax, ms2[esi]

mov result\_ms[esi], eax

add esi, 4

dec ebx //декрементим количество проходов

cmp ebx, 0 //сравниваем с нулём

jne cycleAsm

}

}

end = clock();

//output(result\_ms);

cout << "ASM: " << (double)(end - start) / CLOCKS\_PER\_SEC << endl;

/\*MMX\*/

null(result\_ms);

start = clock();

for (int i = 0; i < COUNT\_CYCLE; i++)

{

\_asm {

pusha

xor esi, esi //Зануляем индекс

mov ebx, 8 //количество проходов по циклу (матрица 4x4)

cycleMMX:

movq MM0, ms1[esi] //Заносим значение в регистр MM0

psubq MM0, ms2[esi] // вычитаем значение другого массива

movq result\_ms[esi], MM0 //записываем значение в массив разности

add esi, 8 //передвигаем указатель на 4 байта (int)

dec ebx //декрементим количества проходов

cmp ebx, 0 //сравниваем с 0

jne cycleMMX

emms

popa

}

}

//output(result\_ms);

end = clock();

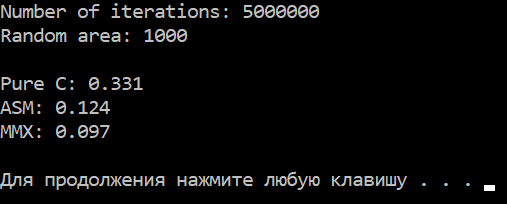
cout << "MMX: " << (double)(end - start) / CLOCKS\_PER\_SEC << endl << endl;

system("pause");

return 0;

}

1. Тестовые пример



1. Заключение

В ходе выполнения лабораторной работы были получены результаты выполнения вычитания матриц, реализованной разными способами: средствами языка Си, ассемблера и ассемблера с использованием MMX. Время, затраченное на выполнение вычислений функцией, реализованной с помощью команд MMX меньше аналогичных, реализованных на языке Си и ассемблере.