Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет (институт) Информационных технологий

Кафедра Прикладная математика

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_

А.И.Потупчик

(подпись преподавателя) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Отчет

по лабораторной (практической) работе №\_\_8\_\_

Использование SIMD-расширений для реализации векторных и матричных алгоритмов

(название лабораторной (практической) работы)

по дисциплине Архитектура ЭВМ

(наименование дисциплины)

ЛР 09.03.04.21.000 ОТ

(обозначение документа)

Студент группы ПИ-02 Р.А.Чередов

(инициалы, фамилия)

Преподаватель доцент, доцент А.И.Потупчик

(должность, ученое звание) (инициалы, фамилия)

Барнаул 2022

Задание:

Используя ММХ-расширение процессоров I80x86 реализовать заданный алгоритм.

Описние системы команд ММХ и задания находятся в личном кабинете.

**21. Дан двумерный массив b. Рассматривая его как матрицу, транспонировать b.**

Решение:

#include <iostream>

#include "stdio.h"

#define \_CRT\_SECURE\_NO\_WARNINGS

using namespace std;

// Вывод матрицы в консоль

void print(int\* arr, int row, int column) {

for (int i = 0; i < row; ++i) {

for (int j = 0; j < column; ++j) {

cout << arr[i \* column + j] << "\t";

}

cout << endl;

}

}

int main() {

const int row = 4; // Кол-во строк матрицы

const int column = 2; // Кол-во столбцов матрицы

int array\_A[row][column]; // Исходная матрица

int array\_temp[row][column];// Матрица для копирования

int row\_fun = row, column\_fun = column; // Значения количества столбцов и колонок для циклов по типу for

// Заполнение матрицы

for (int i = 0; i < row\_fun; ++i) {

for (int j = 0; j < column\_fun; ++j) {

array\_A[i][j] = i \* column + j + 1;

}

}

print(\*array\_A, row\_fun, column\_fun); // Печать исходной матрицы

cout << "Start matrix" << endl;

// Очистка данных для корректной работы

\_\_asm {

and ecx, 0FFFFh;

mov eax, 0

mov ebx, 0

}

// Транспонирование матрицы

for (int i = 0; i < column; ++i) {

for (int j = 0; j < row; ++j) {

\_\_asm {

// Достаём нужные элементы из матрицы

// В eax кладём отступ для нужного элемента матрицы

mov eax, j

mov ebx, column

mul ebx

mov ebx, 4

mul ebx

push eax

mov eax,i

mov ebx, 4

mul ebx

pop ebx

add eax,ebx

movd mm0, [array\_A + eax] // в mm0 кладём элемент

// Сохраняем нужные элементы в матрицу

// В eax кладём отступ на место куда будет сохранён нужный элемент матрицы

mov eax, i

mov ebx, row

mul ebx

mov ebx, j

add eax,ebx

mov ebx, 4

mul ebx

movd [array\_temp + eax], mm0 // из mm0 сохраняем элемент во вторую матрицу

}

}

}

// Очистка данных для корректной работы программы

\_\_asm {

and ecx, 0FFFFh;

mov ebx, 16 // В регистр ebx кладём отступ для команд ммх

}

// Из временной матрицы копируем данные в основную матрицу

for (int i = 0; i < column; ++i) {

\_\_asm {

// Получаем индекс строки из которой достаются элементы

mov eax, i

mul ebx

movups xmm0, [array\_temp+eax] // Кладём первые 4 элемента в xmm0

movups [array\_A+eax], xmm0 // Сохраняем эти 4 элемента в основную матрицу

}

}

// Меняем количество столбцов и колонок местами

row\_fun = column;

column\_fun = row;

cout << endl;

print(\*array\_A, row\_fun, column\_fun);

cout << "End matrix" << endl;

system("pause");

}

Тесты





