Учреждение Образования

«Белорусский Государственный Университет Информатики и Радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина: АВС

Отчет

По лабораторной работе № 9

ТЕХНОЛОГИЯ ММХ

Выполнил студент

Группы 253505

Волчецкий А.М.

Минск 2014

### *Лабораторная работа 7.*

Для выполнения данной работы необходимо выполнить следующее:

а) изучить программную модель ММХ;

б) изучить систему команд ММХ;

в) обработать массивы из 8 элементов по следующему выражению (в зависимости от варианта):

1. F[i]=-A[i] -B[i] +C[i] -D[i] , i=1...8.

Используются следующие массивы:

**A**, **B** и **С** – 8 разрядные целые знаковые числа (\_int8);

**D** – 16 разрядные целые знаковые числа (\_int16).

Полученный результат отобразить на форме с использованием соответствующих элементов. При распаковке знаковых чисел совместно с командами распаковки использовать команды сравнения (сравнивать с нулём перед распаковкой).

#include "stdafx.h"

#include <iostream>

#include <stdio.h>

#include <conio.h>

#include <clocale>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

\_\_int8 a[] = { 1, 2, 3, 4, 5, 6, 7, 8 };

\_\_int8 b[] = { 1, 2, 3, 4, 5, 6, 7, 8 };

\_\_int16 c[] = { 1, 2, 3, 4, 5, 6, 7, 8 };

\_\_int16 d[] = { 1, 2, 3, 4, 5, 6, 7, 8 };

\_\_int16 f[] = { 0, 0, 0, 0, 0, 0, 0, 0 };

// F[i] = C[i] -A[i] - B[i] - D[i]

\_\_asm

{

//Загружаем массивы из памяти в регистры MMX

//movq для 64-разрядных данных

//mm0-mm7 -- 64-разрядные регистры

//сначала работаем с младшими байтами

movq mm0, [c]

movq mm1, [a]

movq mm2, [b]

movq mm3, [d]

// преобразовываем младшие байты в слова

//это делаем для выполнения операций с массивом с и d (т.к. размерность его элементов--слово)

punpcklbw mm1, mm4

punpcklbw mm2, mm4

psubw mm0, mm1 // C[i] -A[i]

psubw mm0, mm2 //C[i] -A[i] - B[i]

psubw mm0, mm3 //C[i] -A[i] - B[i] - D[i]

movq [f], mm0

// теперь работаем со старшими байтами

movq mm0, [c + 8]

movq mm1, [a]

movq mm2, [b]

movq mm3, [d + 8]

//теперь преобразовываем старшие байты в слова

punpckhbw mm1, mm4

punpckhbw mm2, mm4

psubw mm0, mm1 // C[i] -A[i]

psubw mm0, mm2 //C[i] -A[i] - B[i]

psubw mm0, mm3 //C[i] -A[i] - B[i] - D[i]

movq[f + 8], mm0

}

for (int i = 0; i < 8; i++)

{

printf("%d ", f[i]);

}

\_getch();

return 0;

}