Лабораторная работа №2

**"Команды MMX/XMM"**

К теме: Технология MMX. Технология SSE. Регистры MMX/XMM, типы данных и команды MMX/ХММ.

# Цель работы

Изучить расширение системы команд MMX процессоров Intel.

Продолжительность работы - 4 часа.

# ЗАДАНИЕ

Создать консольное приложение, которое выполняет заданные вычисления (в соответствии с вариантом) 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 (1 млн) раз. Размер матриц (векторов) кратен количеству элементов в регистре ММХ.

# Теоретические сведения

MMX (MultimediaExtensions- мультимедийное расширение) — коммерческое название дополнительного набора инструкций, выполняющих характерные для процессов кодирования/декодирования потоковых аудио/видео данных действия за одну машинную инструкцию. Впервые появился в процессорах Pentium MMX. Разработан в лаборатории Intel, в первой половине 1990-х.

SIMD (англ.SingleInstruction, MultipleData) - принцип компьютерных вычислений, позволяющий обеспечить параллелизм на уровне данных.

Для обработки данных и хранения промежуточных результатов в Pentium MMX используются восемь 64-разрядных регистров MM0..MM7, которые физически совмещены со стеком регистров математического сопроцессора. При выполнении любой из MMX-команд происходит установка «режима MMX» с отметкой этого в слове состояния сопроцессора (FPU TagWord). С этого момента стек регистров сопроцессора рассматривается как набор MMX-регистров; завершает работу в режиме MMX команда EMMS (EndMultiMediaState). С одной стороны, такая реализация позволила избежать проблем совместимости с механизмами переключения контекста в существующих операционных системах, поскольку число регистров процессора, и, следовательно, код, выполняющий их сохранение и восстановление, не изменились. С другой стороны, переход между режимами занимает значительное время, и совмещение, например, в одном цикле команд сопроцессора с командами MMX может не только не ускорить, а даже существенно замедлить выполнение программы. Поэтому при необходимости для достижения наилучших результатов рекомендуется группировать эти команды отдельно друг от друга.

Команды технологии MMX работают с 64-разрядными целочисленными данными, а также с данными, упакованными в группы (векторы) общей длиной 64 бита. Такие данные могут находиться в памяти или в восьми MMX-регистрах.

Команды технологии MMX работают со следующими типами данных:

* упакованные байты (восемь байтов в одном 64-разрядном регистре) (англ. *packedbyte*);
* упакованные слова (четыре 16-разрядных слова в 64-разрядном регистре) (*packedword*);
* упакованные двойные слова (два 32-разрядных слова в 64-разрядном регистре) (*packeddoubleword*);
* 64-разрядные слова (*quadword*).

MMX-команды имеют следующий синтаксис: instruction dest, src. Здесь instruction - имя команды, dest - выходной операнд, src - входной операнд.

Большинство команд имеют суффикс, который определяет тип данных:

* B, W, D, Q указывают тип данных. Если в суффиксе есть две из этих букв, первая соответствует входному операнду, а вторая — выходному.

SSE (англ.StreamingSIMDExtensions) - это SIMD набор инструкций, разработанный Intel. Технология SSE позволяла преодолеть проблему одновременного использования сопроцессора (его регистры использовались для MMX и работы с вещественными числами).

SSE включает в архитектуру процессора восемь 128-битных регистров (xmm0 до xmm7), каждый из которых трактуется как 4 последовательных значения с плавающей точкой одинарной точности. SSE включает в себя набор инструкций, который производит операции со скалярными и упакованными типами данных.

Преимущество в производительности достигается в том случае, когда необходимо произвести одну и ту же последовательность действий над разными данными.

# ВАРИАНТЫ заданий

1. Вычитание матриц.
2. Вычисление матричного выражения A = B + kC, где A, B, C – матрицы, k – скалярный множитель.
3. Нахождение суммы квадратов всех элементов матрицы.
4. Сформировать из двух массивов "максимальный" (результат[i] = max(массив1[i], массив2[i])). \*pmaxub, pmaxsb, pmaxuw, pmaxsw.
5. Копирование элементов одной матрицы в другую.
6. Нахождение суммы всех элементов матрицы.
7. Умножение матриц.
8. Выполнить операцию логического И для каждой пары соответствующих элементов.
9. Нахождение сумм по столбцам матрицы.
10. Нахождение сумм по строкам матрицы.
11. Сложение матриц.
12. Сформировать массив средних значений (результат[i] = avg(массив1[i], массив2[i]) ) \*pavgb, pavgw
13. Сформировать из двух массивов "минимальный" (результат[i] = min(массив1[i], массив2[i])). \*pminub,pminsb,pminuw,pminsw
14. Выполнить операцию логического ИЛИ для каждой пары соответствующих элементов.
15. Выполнить операцию логического исключающего ИЛИ для каждой пары соответствующих элементов.

# Вопросы к защите

1. На каком главном принципе основана технология ММХ?
2. С какими типами данных работают ММХ команды?
3. Какая команда обеспечивает переход процессора из режима исполнения MMX-команд в режим исполнения обычных команд с плавающей запятой?
4. Что делает команда movq mm0, mm1?
5. Назначение команды EMMS.
6. Что такое SIMD команды?
7. Назначение команд MMX (где они могут применяться).
8. \*Какое максимальное ускорение можно получить, используя MMX-команды?

# Список рекомендуемой литературы

1. http://www.codenet.ru/progr/optimize/mmx.php
2. http://www.club155.ru/x86cmdsimd-41
3. http://miu.by/rus/kaf\_ep/kaf\_download/19990\_102153768.pdf