

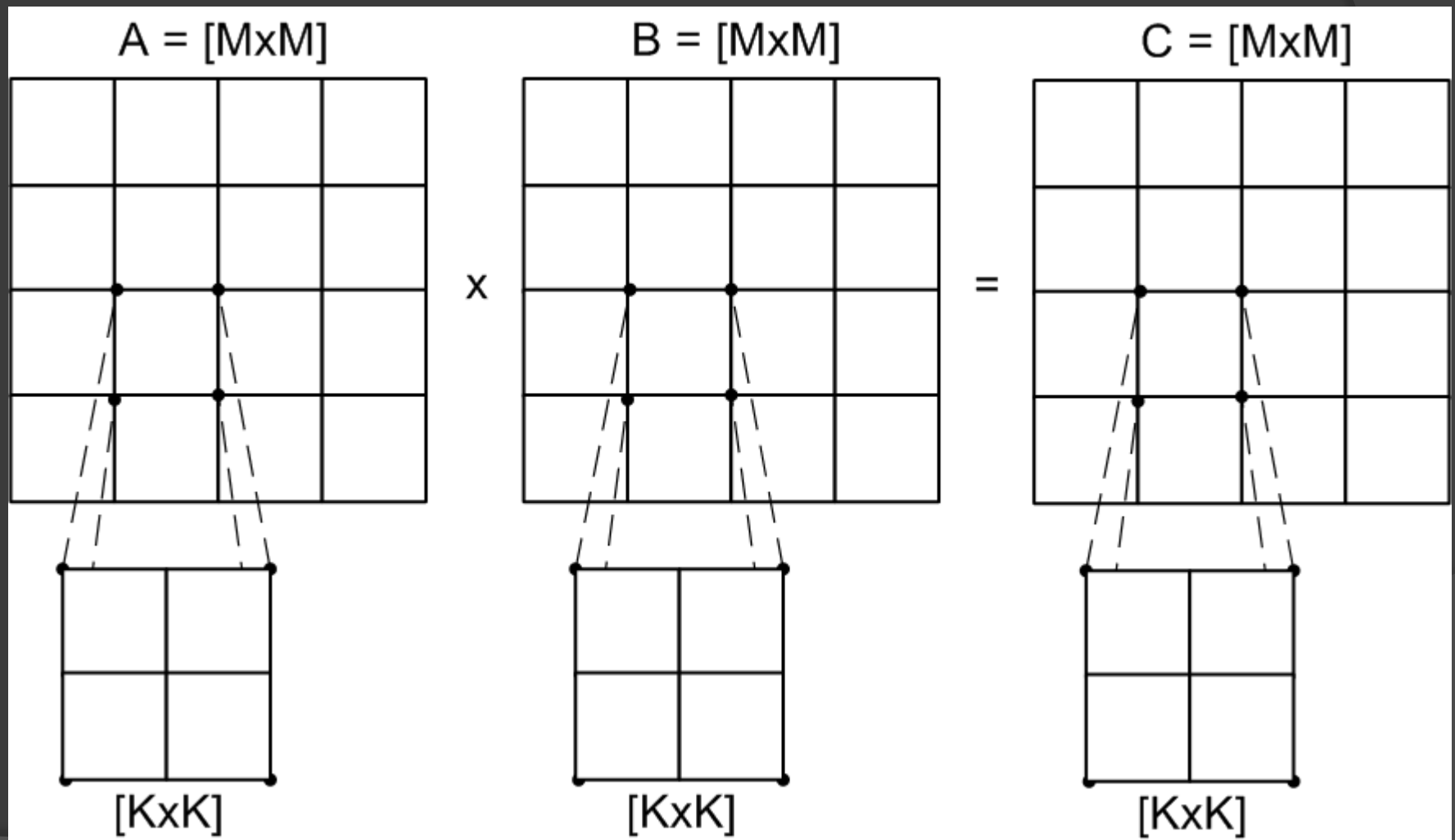
ЛАБОРАТОРНАЯ №1

«ВЕКТОРИЗАЦИЯ»

Задание

- ⦿ Перемножить 2 матрицы следующими способами:
 - с включенной векторизацией
 - с выключенной векторизацией
 - ручная векторизация с использованием SSE2-инструкций (либо новее)
 - ассемблер
 - intrinsics (альтернатива ассемблеру)
- ⦿ Элементы матрицы – своя матрица меньшего размера
- ⦿ Размер матрицы / тип входных данных:
 - определяется преподавателем

Задание



Задание

◎ Обязательные условия:

- в одном проекте минимум (!) 2 функции:
 - с автоматической векторизацией
 - с ручной векторизацией
- бонусные задачи – отдельные функции
- результаты работы всех функций СРАВНИВАЮТСЯ между собой
- время работы SSE2-версии *не* медленнее версии с автоматической векторизацией
- обосновать преимущества бонусных заданий

Замечания по векторизации

⦿ Зависит от среды разработки:

- Visual Studio 2008 / 2010
 - устанавливается Intel C++ Compiler
 - /Qvec – включение векторизации (включена по умолчанию)
 - /QxSSE2 либо /QaxSSE2 – SSE2 набор инструкций
 - /Qvec-report3 – выводить данные об удачной / неудачной векторизации



```
c:\Program Files (x86)\Intel\ComposerXE-2011\bin\ia32\Control.cpp(5) (col. 2): remark: LOOP WAS VECTORIZED.  
Microsoft (R) Incremental Linker Version 10.00.40219.01  
Copyright (C) Microsoft Corporation. All rights reserved.
```

Замечания по векторизации

⦿ Зависит от среды разработки:

- Visual Studio 2012 и новее
- GCC (версия неизвестна, точно с 2009 года)
 - ПРОБЛЕМА:
 - как отключить векторизацию ???
 - подсказки:
 - проверяйте в дизассемблере
 - неправильный способ: есть на MSDN
 - но он принимается в определенных условиях ☺
 - правильный способ: настройка компилятора

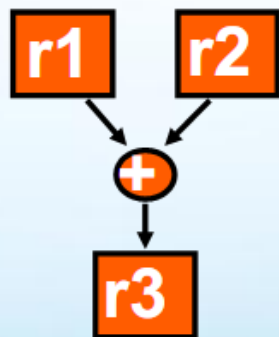
⦿ Ускорение векторизованной версии:

- float: ~3-3,5 раза
- double: ~1,2-1,5 раза

Векторизация

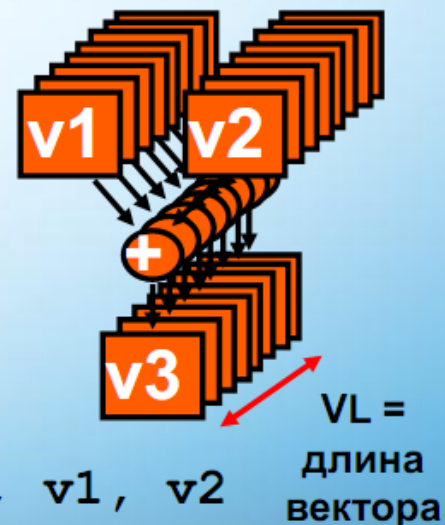
- выполнение одной инструкции над вектором данных
- в лаб.работе транспонировать матрицы запрещено

Скалярная
операция



`add.d r3, r1, r2`

Векторная
операция



`addvec.d v3, v1, v2`

Векторизация

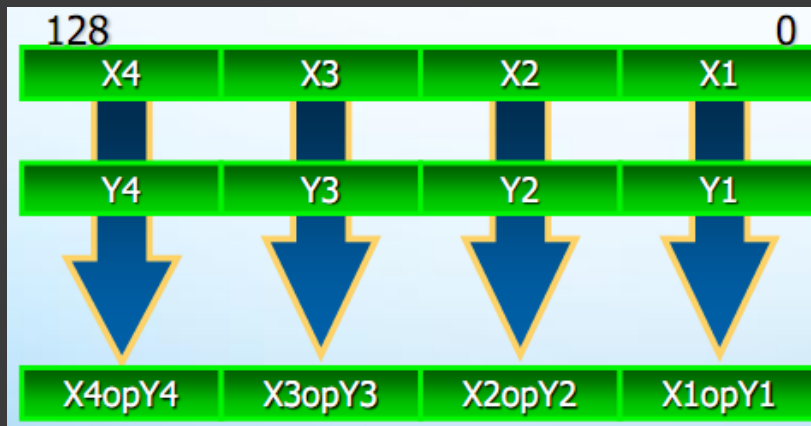
⦿ типы данных:

- int (не во всех SIMD-наборах)
- double, float (кроме MMX)

⦿ наборы SIMD-инструкции:

- MMX – Multi-media extension
 - 64 бит
- SSE – Streaming SIMD Extension
 - 128 бит
- AVX – Advanced Vector Extension
 - 256 бит
 - 512 бит (для AVX2)

Наборы инструкций SSE и AVX



Intel® SSE

Вектор: 128bit

Типы данных:

8,16,32,64 bit integers

32 and 64bit floats

VL: 2,4,8,16



Intel® AVX

Вектор: 256bit

Типы данных: 32 and 64 bit floats

VL: 4, 8, 16

Нет арифметических операций над целыми числами – но есть логические: OR, XOR, AND ...

Требования к коду

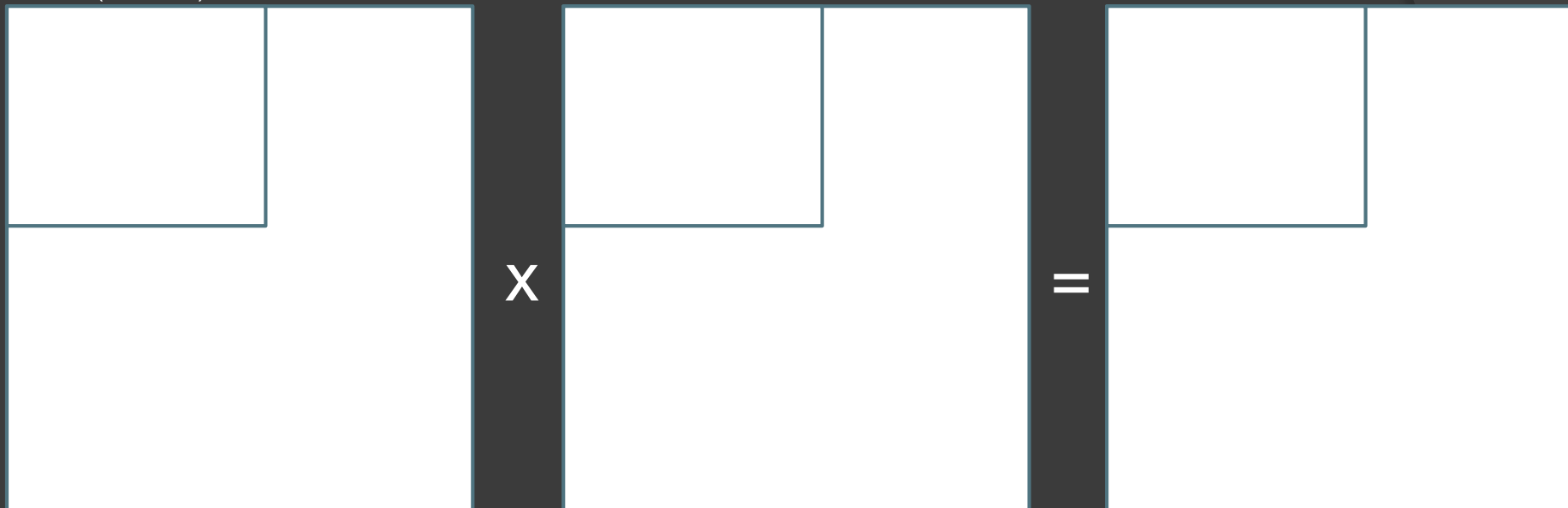
- ⦿ итерации цикла должны быть независимыми
- ⦿ код в цикле «просчитываемый»
- ⦿ нарушение потока команд
- ⦿ внутренний цикл

Система баллов

- ◎ -0,5 балла
 - списано и не поняли, что сделали
- ◎ 0 баллов (любой из критериев):
 - списано, но поняли, что сделали
- ◎ 0,1 балла
 - внутренняя матрица произвольного размера
- ◎ 0,2 балла
 - ручная развертка циклов и их оптимизация под заданную микроархитектуру
- ◎ 0,2 балла
 - оптимизация под кэши всех уровней
 - при выполнении задания матрица матриц заменяется на обычную матрицу

Оптимизация под кэш

$\text{mod}(\text{size}, 64)=0$



- умножение выполняется поблочно
- блок одной из матриц остается без изменений до тех пор, пока не будет обработаны все блоки второй матрицы

Рекомендуемое чтение

- ◎ [24504501.pdf](#) — то, как рекомендуется перемножать матрицы от Intel
 - P.S. Версия для тех, кому скучно ☺
- ◎ [matrixmult.pdf](#) — описание оптимизации под кэш, которая использовалась в GotoBLAS
- ◎ [crmemory.pdf](#) — информация к размышлению
 - и ее [перевод](#)