

Ватутин Э.И.

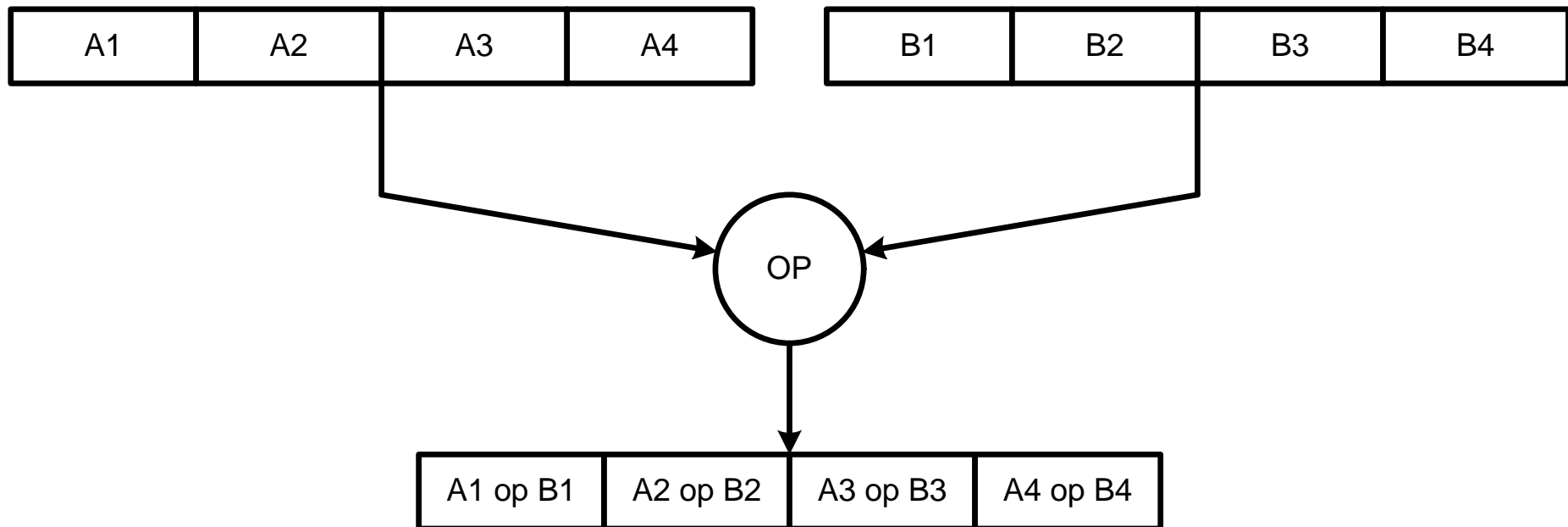
Курский государственный технический университет

**SIMD-ОПТИМИЗАЦИЯ
МАТЕМАТИЧЕСКИХ РАСЧЕТОВ
НА ПРИМЕРЕ ЗАДАЧИ
ВЫЧИСЛЕНИЯ ПЛОЩАДИ
ТРЕУГОЛЬНИКА**

Курск 2006

SIMD-принцип вычислений

SIMD (Single Instruction Multiple Data) принцип заключается в выполнении однотипных действий одной командой над несколькими операндами





Область применение SIMD

Обработка больших массивов однородной информации:

- операции с векторами, матрицами (3D-графика, математические расчеты);
- обработка изображений;
- цифровая фильтрация;
- многое другое.

Обычно используются приемы *векторизации* (поддерживаются современными продвинутыми оптимизирующими компиляторами – ICL, MSVC++, GCC)



Расширение области применения SIMD-вычислений

В отсутствии явных признаков векторизации (массивы однородной информации) или при наличии ограничений (например, зависимости по данным) SIMD-принцип зачастую не применяется. Сферу его применения можно расширить...



Математические расчеты

Как правило характеризуются наличием некоторого количества однотипных независимых по данным действий, которые можно эффективно реализовать с использованием SIMD-расширений (признаки векторизации отсутствуют).

Пример: вычисление площади треугольника по известным координатам его вершин.



Математическая модель

1. Вычисление длин сторон треугольника

$$l_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}, \quad i = \overline{1, 3}$$

2. Вычисление полупериметра

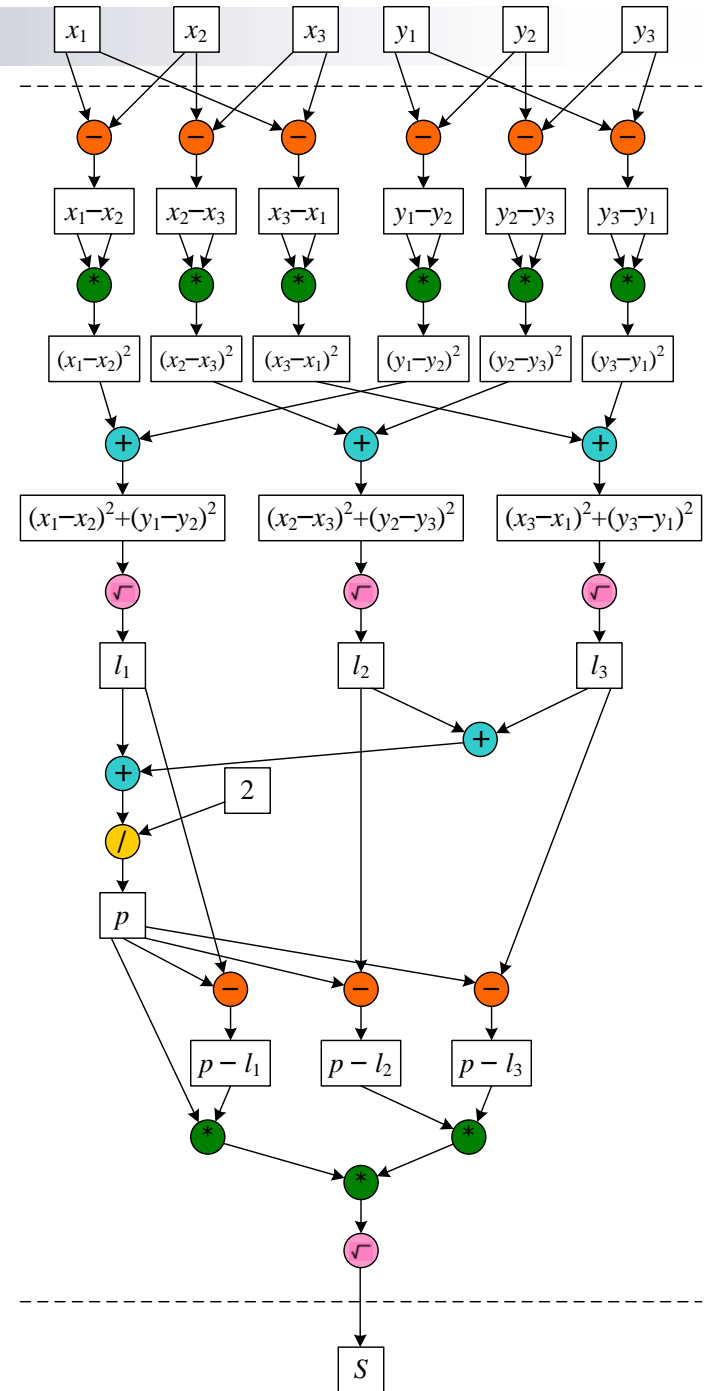
$$p = \frac{l_1 + l_2 + l_3}{2}$$

3. Вычисление искомой площади

$$S = \sqrt{p(p - l_1)(p - l_2)(p - l_3)}$$

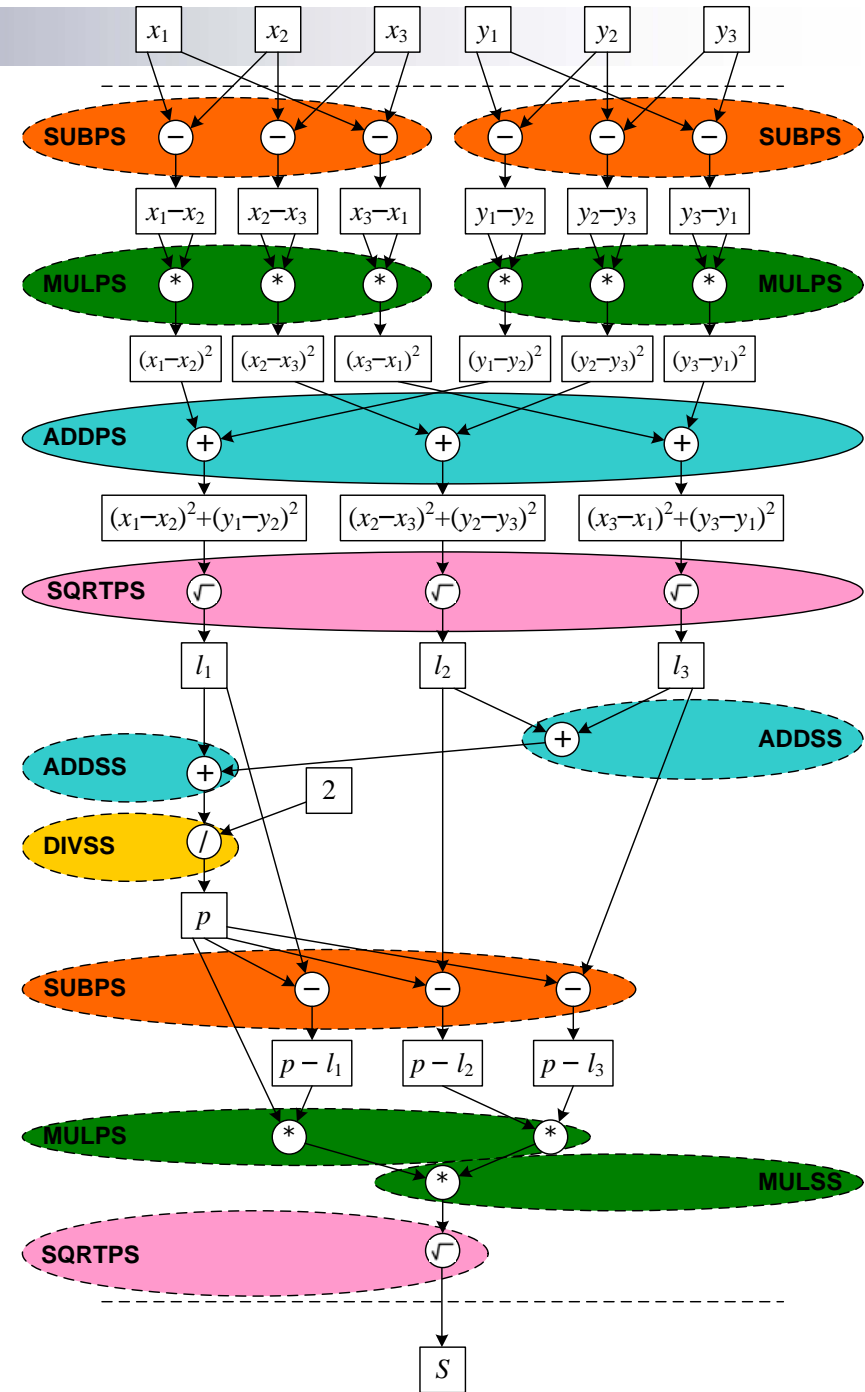
Схема вычисления

На схеме в виде ярусов изображены группы операций, выполнение которых можно производить параллельно

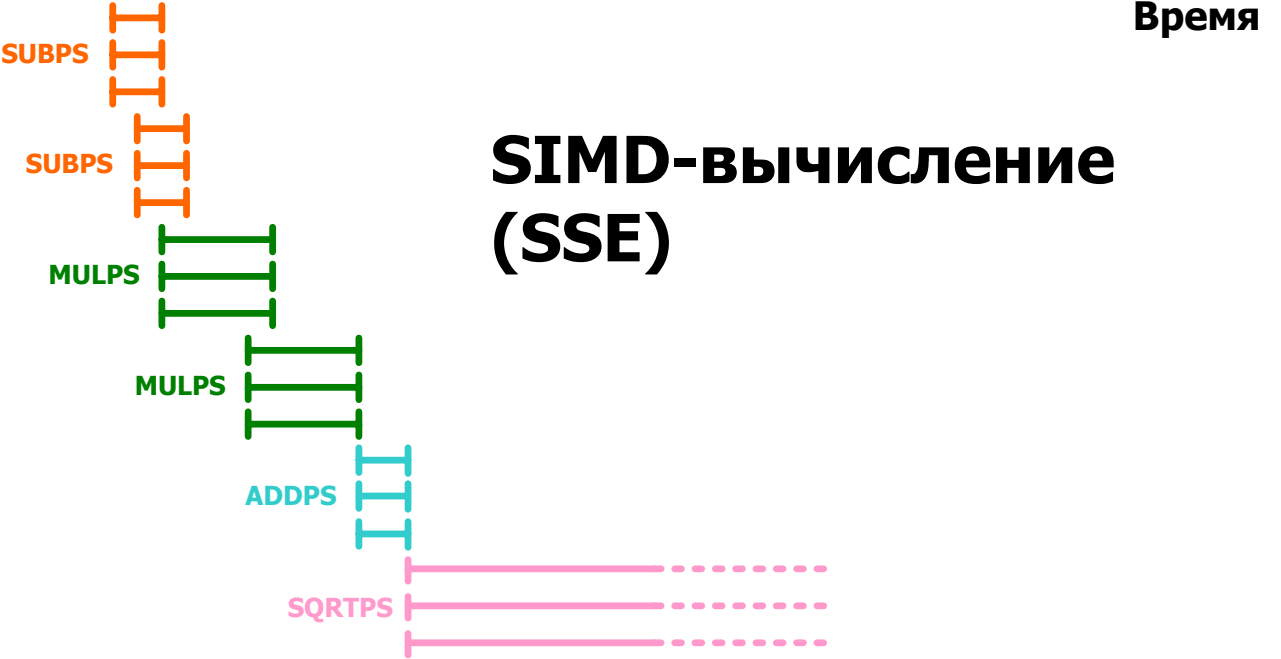
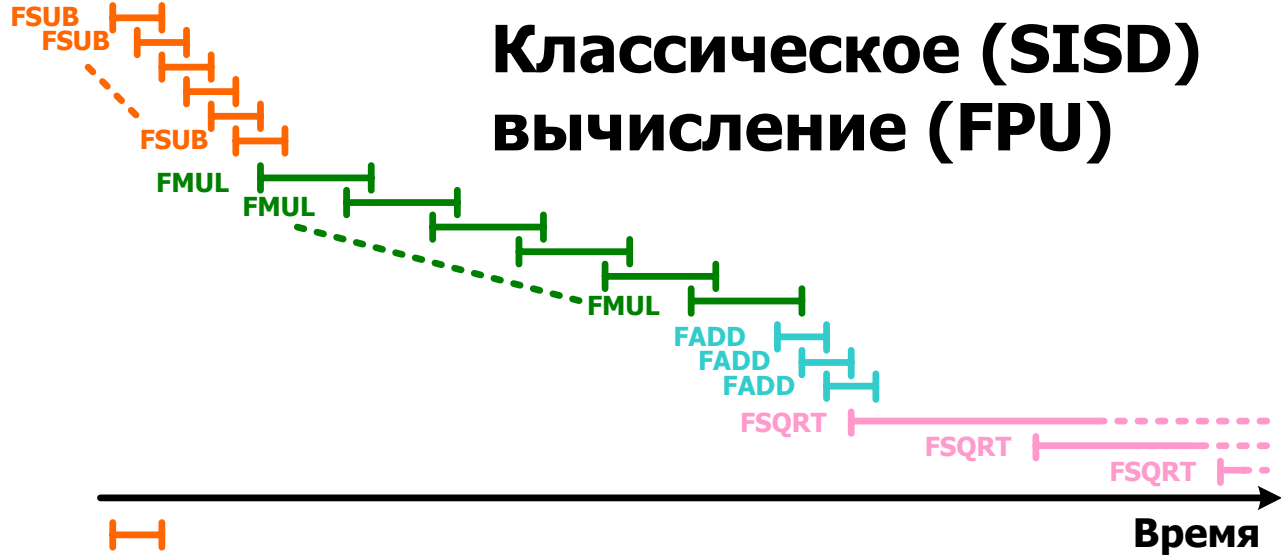


Реализация с помощью SSE-расширения

(команды перепаковки аргументов **SHUFPS** для упрощения не показаны)



Временная диаграмма





Результаты тестирования


Delphi – 412 тактов

Intel C++ Compiler 8.0 – 367 тактов

FPU (вручную) – 343 такта

SIMD (вручную) – 195 тактов

Меньше – лучше!
Delphi/SIMD = 2,11 выигрыш



Сложности ручного кодирования

- многообразии способов группировки элементарных операций в группы;
- учет влияния операций перепакетки данных;
- учет влияния зависимостей по данным;
- учет микроархитектуры процессора (занятость исполнительных устройств, портов запуска, регистров переименования и т.д.);
- влияние порядка расположения команд (декодирование);
- появление асимметричных команд (например, **ADDSUBPD** из SSE3)
- человеческий фактор (затраты времени на кодирование/отладку!).



**Доклад окончен.
Спасибо за внимание!**