

Московский Государственный Университет им. Ломоносова
Факультет ВМК кафедра СКИ

Использование сопроцессоров для высокопроизводительных вычислений

Бурцев Александр
apburtsev@gmail.com

Москва 2013

Содержание

1. Несколько слов о параллелизме (введение)
2. Архитектура CPU
3. Архитектура GPU
 1. CUDA
 2. OpenCL
 3. OpenACC
4. Архитектура MIC
 1. OpenMP

Несколько слов о параллелизме (введение)

Виды параллелизма:

1. По виду работ:

1. Параллелизм по данным

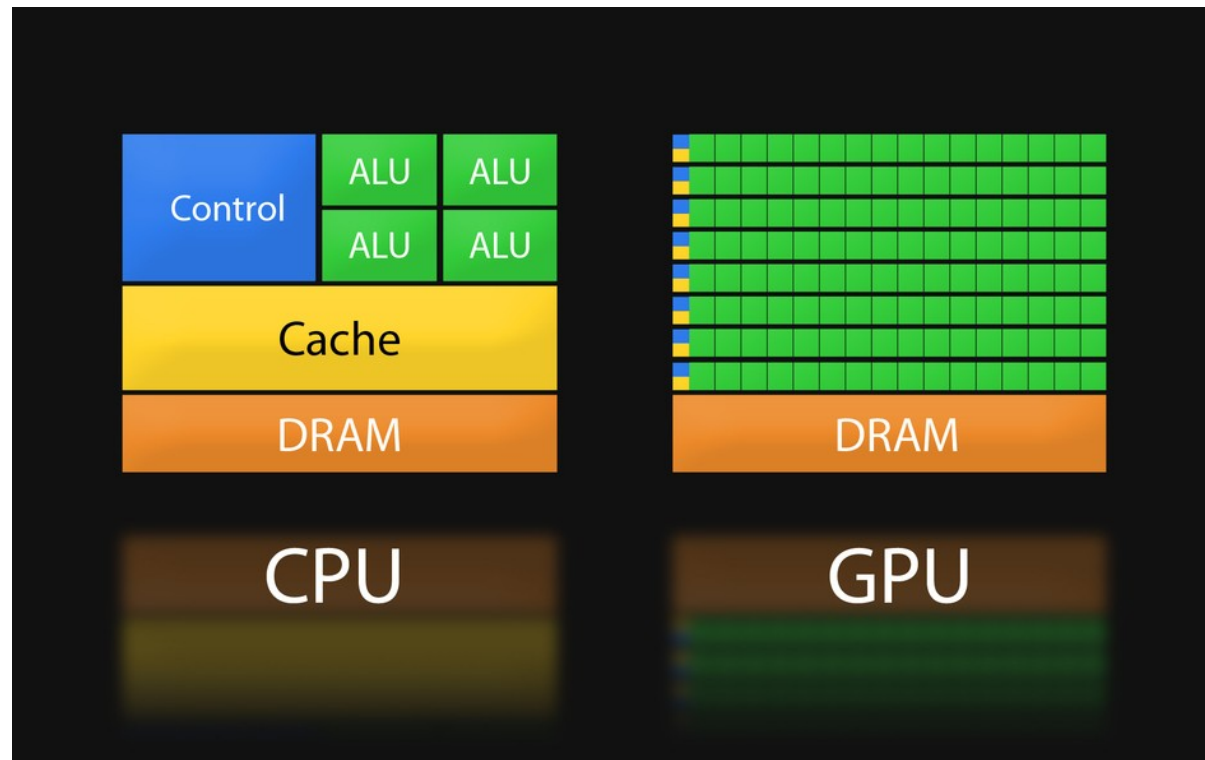
2. Параллелизм по заданиям

2. По объёму работ:

1. Крупнозернистый параллелизм

2. Мелкозернистый параллелизм

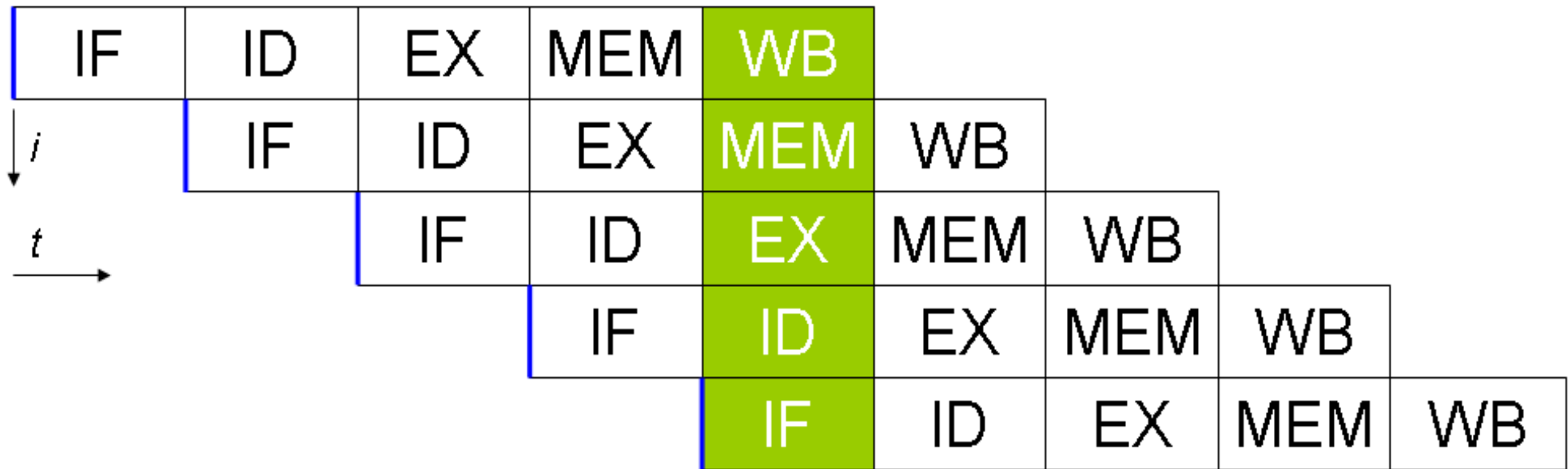
Архитектура CPU



Особенности CPU:

- Большой КЭШ (многоуровневый)
- Мощная система выборки и дешифровки инструкций Control Unit
- Мощные ALU

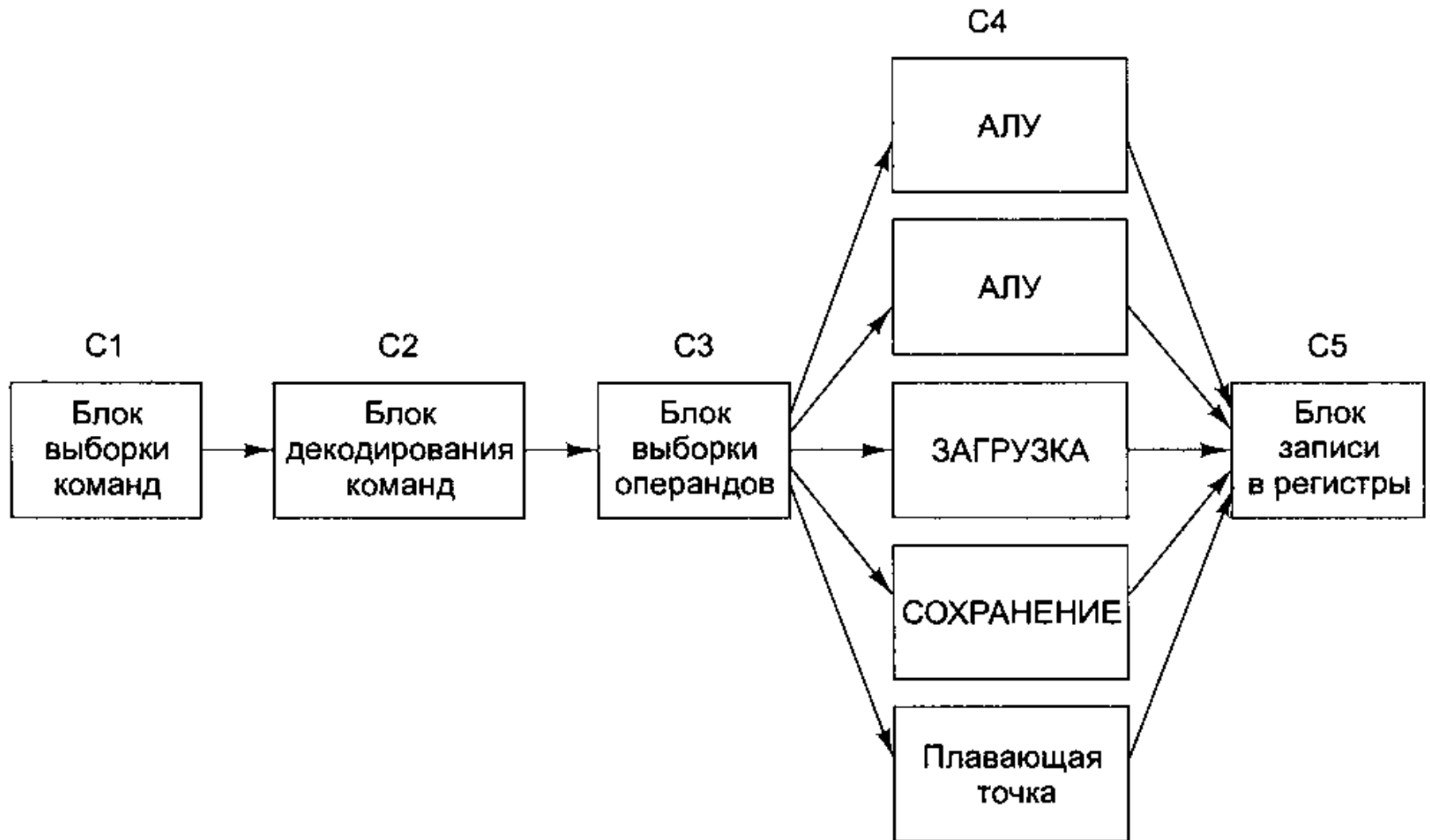
Архитектура CPU



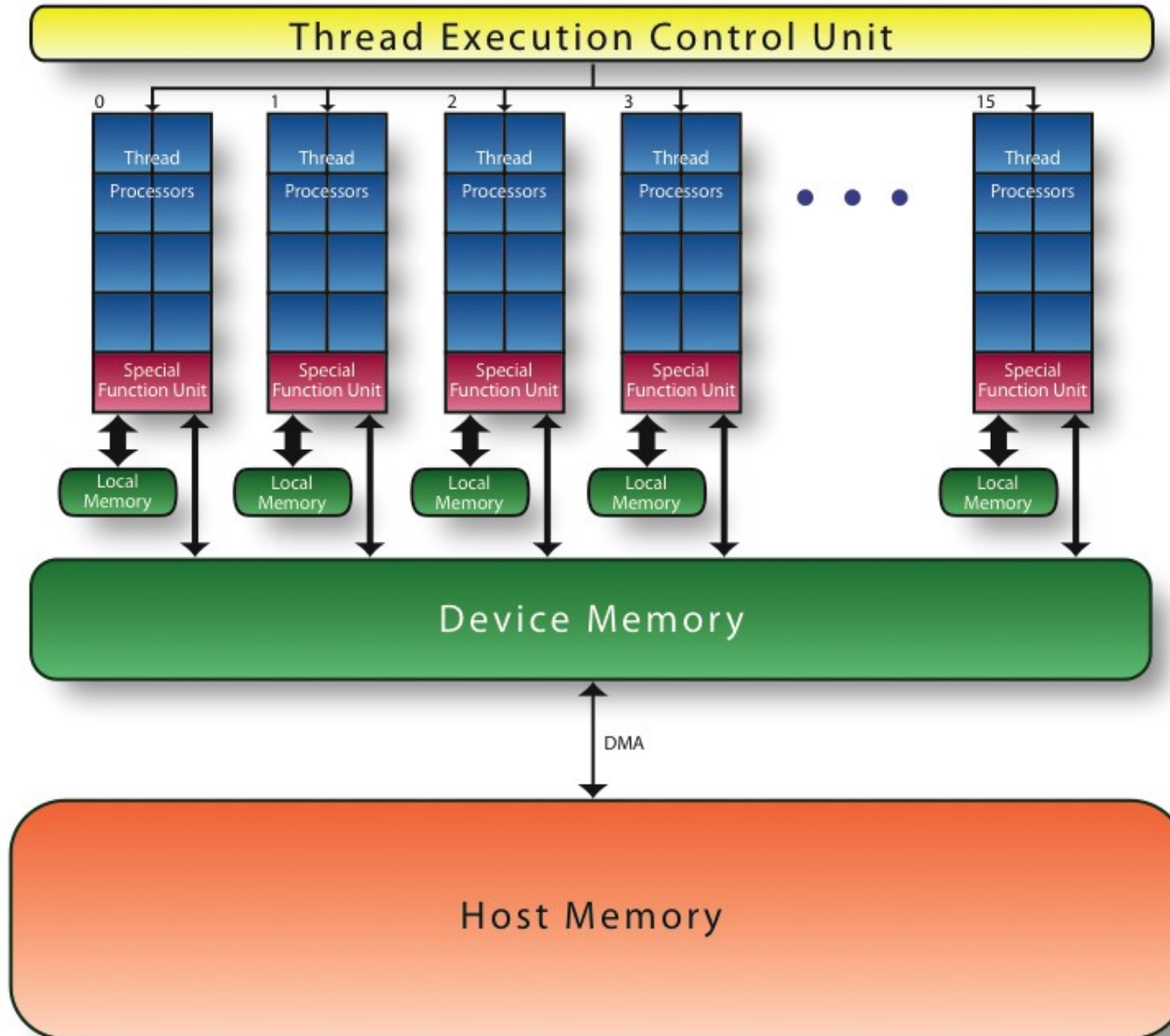
Классический пример пятиступенчатого конвейера на RISC-машине (IF = выборка инструкции, ID = декодирование инструкции, EX = выполнение инструкции, MEM = доступ к памяти, WB = запись результата в регистры).

Материал из Википедии — свободной энциклопедии. Статья: Параллельные вычислительные системы.

Архитектура CPU



Архитектура GPU



- Особенности GPU:
- Маленький КЭШ
 - Слабые ALU объединенные в группы
 - Отсутствие развитой системы выборки команд
 - Модуль спец. функций

CUDA

Возможности:

1. Использование текстурной и общей памяти
2. Полная аппаратная поддержка целочисленных и побитовых операций
3. Поддержка компиляции GPU кода средствами открытого LLVM в ассемблерные инструкции
4. Асинхронные операции

Порядок работы:

1. Выделение памяти на устройстве
2. Копирование памяти (CPU=>GPU)
3. Запуск ядра
4. Копирование памяти (GPU=>CPU)
5. Очистка памяти

OpenCL

Особенности:

1. Открытый стандарт, поддержка GPU всех производителей а так же многопроцессорных систем
2. Компиляция кода на этапе выполнения
3. Векторные типы, синхронизация, функции для Work-items/Work-Groups

Порядок работы:

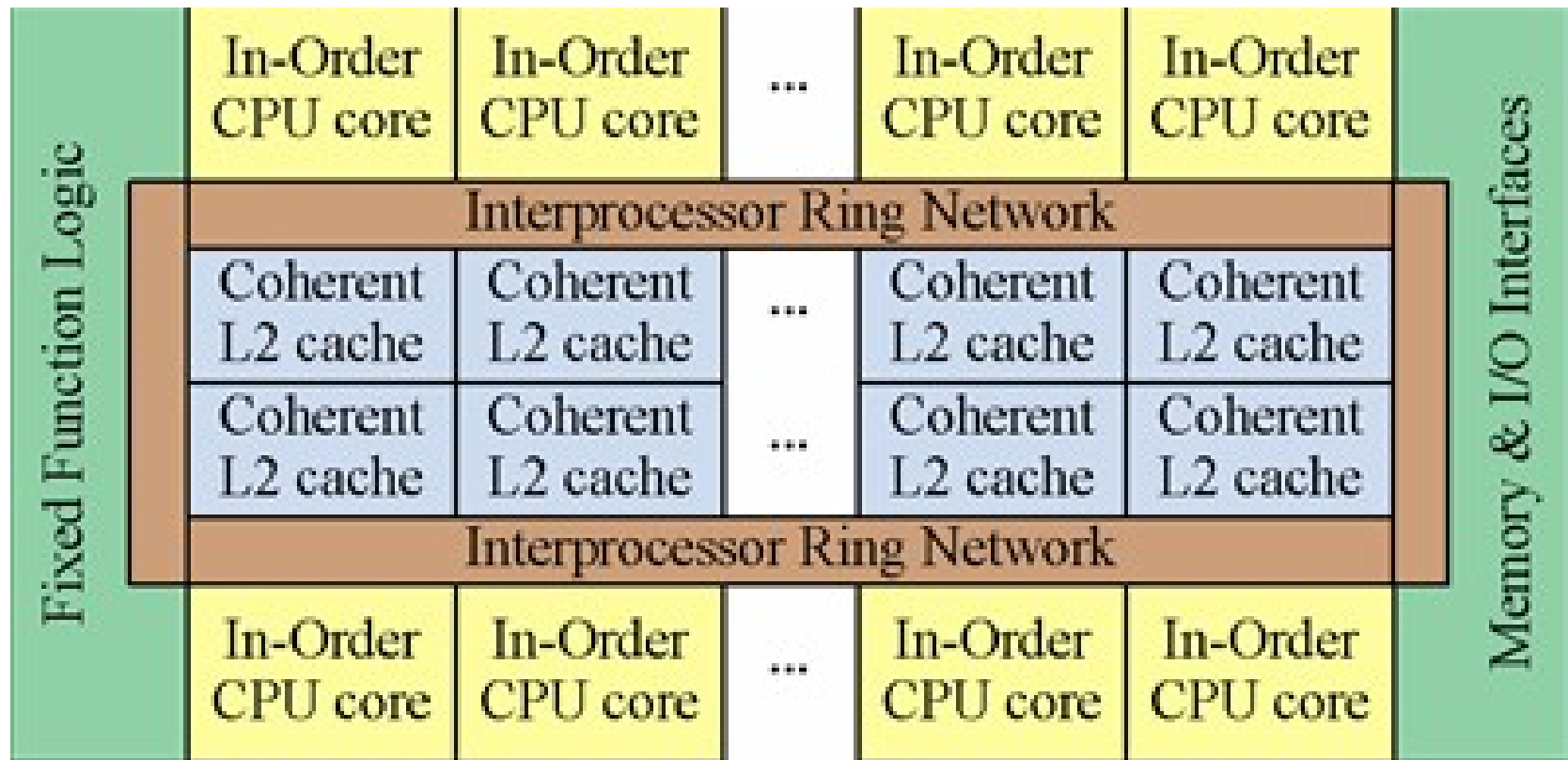
1. Определить доступные ресурсы в гетерогенной системы и выбрать подходящие
2. Создать последовательность инструкций, которые будут выполняться на ресурсах
3. Подготовить начальные данные для вычислений
4. Заставить эти ресурсы выполнить OpenCL-инструкции, обрабатывающие подготовленные данные
5. Собрать результаты вычислений

OpenACC

Особенности:

1. Лёгкая работа с помощью директив
2. Возможен контроль за использованием памяти
3. Прозрачная работа с GPU

Архитектура MIC



Особенности MIC:

- Набор команд x86
- 512-битные векторные АЛУ
- Когерентный L2 КЭШ
- Сверхширокая кольцевая шина для связи ядер и контроллера памяти
- ОС на основе FreeBSD

OpenMP

Ключевыми элементами OpenMP являются:

- Работа в системах с общей памятью
- Конструкции для создания потоков (директива `parallel`)
- Конструкции распределения работы между потоками (директивы `DO/for` и `section`)
- Конструкции для управления работой с данными (`shared` и `private`)
- Конструкции для синхронизации потоков (директивы `critical`, `atomic` и `barrier`)
- Процедуры библиотеки поддержки времени выполнения (например, `omp_get_thread_num`)
- Переменные окружения (например, `OMP_NUM_THREADS`)

Intel Cilk Plus и векторные операции

Особенности Cilk

- Поддержка в компиляторах icc и gcc
- Директивы управления потоками (spawn, sync, Inlet, abort)

Особенности работы с векторными инструкциями:

- Векторизация производится компилятором автоматически
- Для компилятора icc существует набор директив отвечающих за ручную векторизацию

Спасибо за внимание

apburtsev@gmail.com