

Зори С.А. "АК", 2020

Параллельные архитектуры

Классификация

- Различают:
 - программную модель – то, как выглядит компьютер с точки зрения программы
 - аппаратную архитектуру
- Они могут отличаться
- Одна аппаратная архитектура может поддерживать разные программные модели

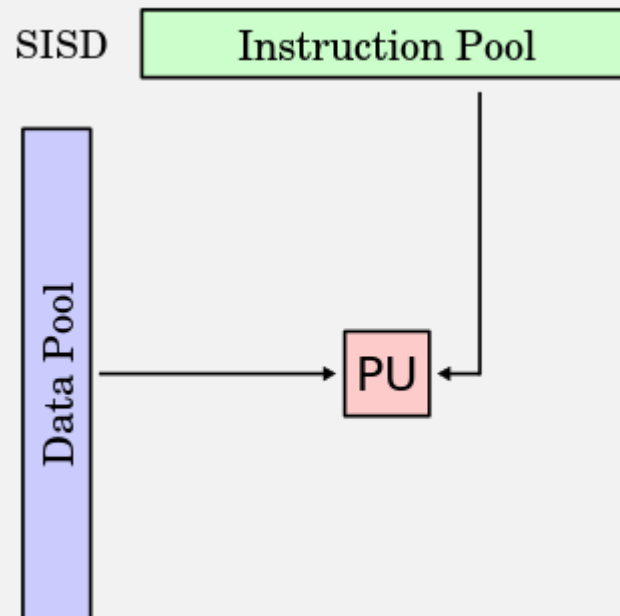
Классификация

- Простейшая классификация параллельных архитектур – по Флинну

Поток команд	Поток данных	
	Один (Single Data)	Много (Multiple Data)
Один (Single Instruction)	SISD (ОКОД)	SIMD (ОКМД)
Много (Multiple Instruction)	MISD (МКОД)	MIMD (МКМД)

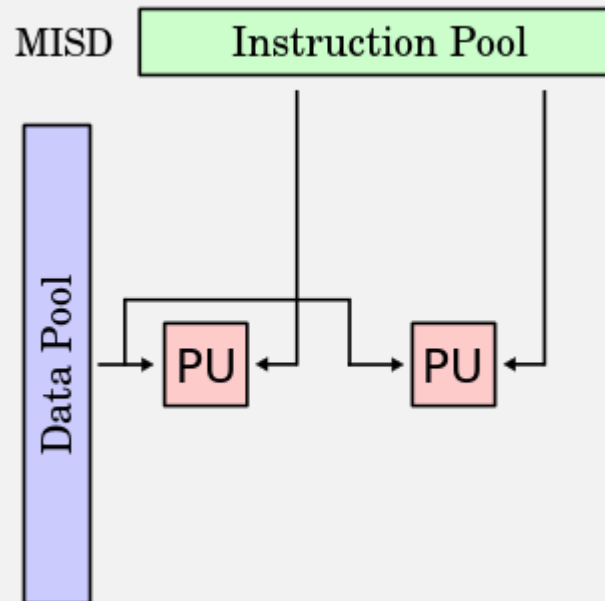
SISD

- Наиболее старая, простая и медленная - Одноядерная машина
- Все лабораторные в этом курсе до настоящего момента использовали *программную модель SISD*



MISD

- Несколько инструкций применяются к одному данному
- Редко используется (трудно найти применение)



MISD

- Примеры:
 - отказоустойчивые компьютеры, вычисляющие одну и ту же инструкцию параллельно во избежание ошибок. Пример – система управления полетом Шаттла
 - конвейерные архитектуры
 - систолический массив

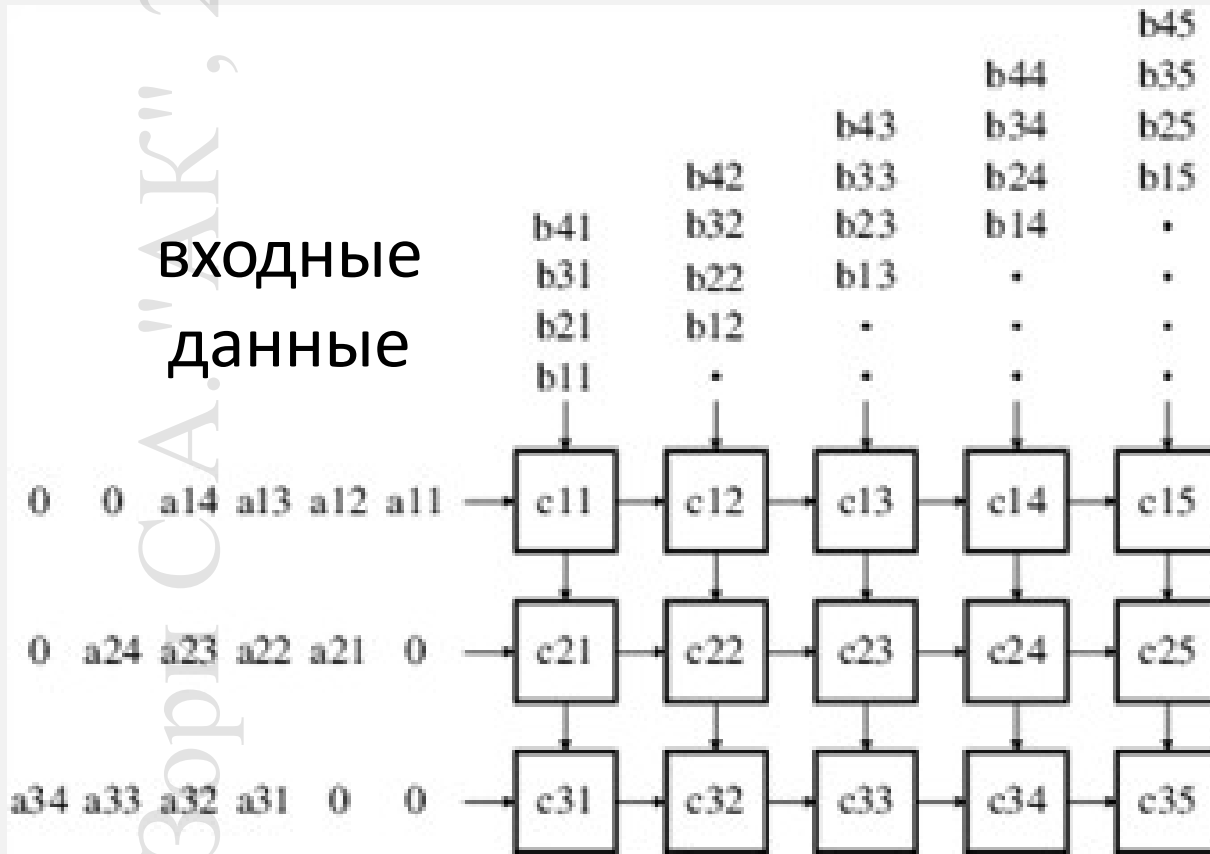
Систолический массив

- Наиболее эффективная вычислительная структура для параллельного выполнения однотипных операций над массивами
- Узко специализирован
- Состоит из матрицы однотипных вычислительных элементов
- Соседние элементы обмениваются данными
- За такт каждый элемент вычисляет результат

Систолический массив

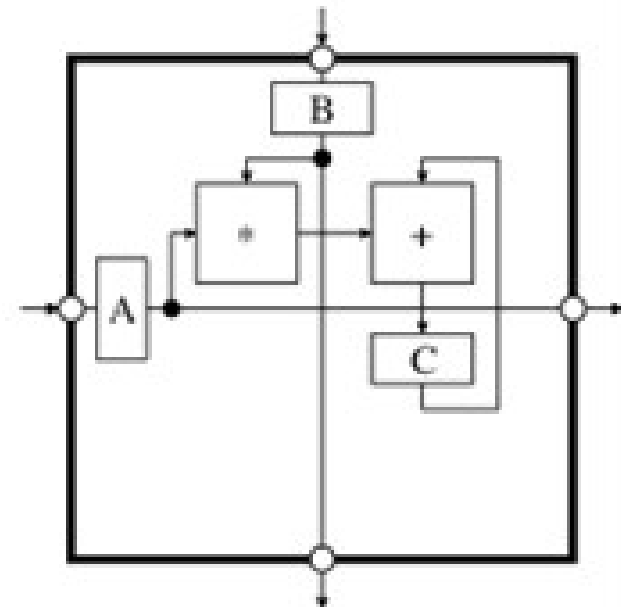
- Пример:
$$c_{ij} = \sum_{k=1}^4 a_{ik} \cdot b_{kj} \quad (1 \leq i \leq 3, 1 \leq j \leq 5).$$

входные
данные



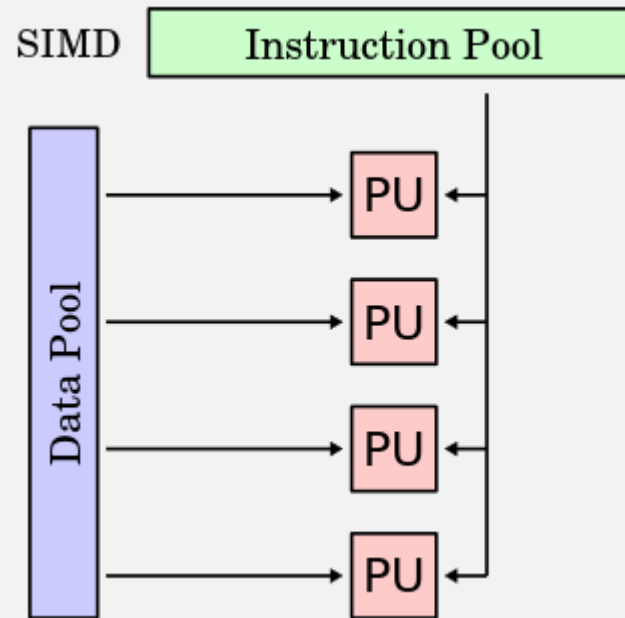
систолический массив

вычислительный
элемент



SIMD

- Одна инструкция применяется к нескольким данным



SIMD

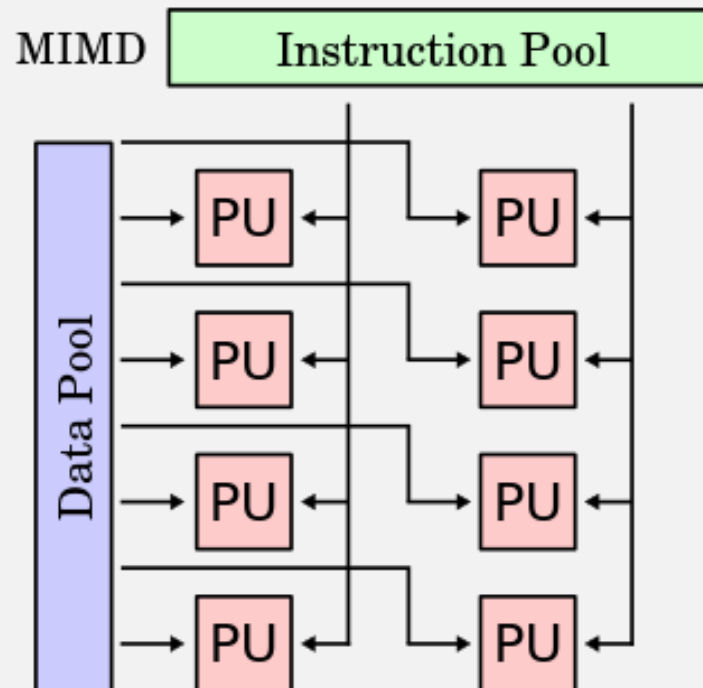
- Примеры:
 - SIMD-инструкции архитектуры x86
 - GPU (видеокарты – де-факто самый доступный, дешевый и популярный вариант многоядерных SIMD по своей сути, K-8 = 2500 процессорных ядер)
- Достоинства:
 - эффективность
 - масштабируемость

SIMD

- Недостатки:
 - не для всех классов задач
 - сложности с ветвлениями
 - сложности с редукцией и “горизонтальными” вычислениями
 - существующие компиляторы не генерируют SIMD код из SISD программы => большие трудозатраты при программировании
 - требуют определенного расположения данных в памяти
- Подробнее – в курсе параллельные вычисления

MIMD

- Несколько инструкций применяется параллельно к нескольким данным



MIMD

- Наиболее сложная и универсальная архитектура
- Состоит однотипных *ядер*, работающих независимо, и планировщика процессов
- К ней относят все современные многоядерные производственные GPU и CPU
- Достоинства:
 - универсальность
 - масштабируемость
 - реальная многозадачность

MIMD

- Проблемы:
 - существующие компиляторы не генерируют MIMD код из SISD программы
 - сложности при программировании (состязания за ресурсы, взаимоблокировки)
 - сложности отладки (недетерминированность поведения)

MIMD

Типы:

1. С общей (разделяемой) памятью,
доступной всем ядрам

- Примеры:

- многоядерные процессоры x86

- GPU

MIMD

1. С общей (разделяемой) памятью

- Проблемы:
 - обеспечение когерентности кэша
 - сложность при программировании – доступ к общим данным
 - ограниченная масштабируемость

MIMD

2. С распределенной памятью. Каждый процессор имеет свою память, недоступную другим.
- Процессоры обмениваются данными в виде сообщений
 - Подробнее – в курсе параллельные вычисления

MIMD

2. С распределенной памятью.

- Примеры:
 - COW (Cluster of Workstations)
 - MPP (massively parallel processors)
 - GRID
 - Cloud Computing

MIMD

2. С распределенной памятью.

- Достоинства:
 - неограниченная масштабируемость
 - возможность географической распределенности
- Проблемы с обменом сообщениями:
 - трудозатраты при программировании
 - накладные расходы
 - пропускная способность канала