



# 1. Уровни параллелизма. Классификации архитектур вычислительных систем.

СЕМИНАРНОЕ ЗАНЯТИЕ ПО ПРЕДМЕТУ  
«ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ»

ПОДГОТОВИЛ СТУДЕНТ ГР. 16ВВ1  
ГРУШЕВСКИЙ АЛЕКСАНДР

# Параллелизм в работе ЭВМ

- ▶ **Параллелизм** – основа высокопроизводительной работы всех подсистем вычислительных машин. Организация памяти любого уровня иерархии, организация системного ввода/вывода, организация мультиплексирования шин и т.д. базируются на принципах параллельной обработки запросов. Современные операционные системы являются многозадачными и многопользовательскими, имитируя параллельное исполнение программ посредством **механизма прерываний**.
- ▶ Развитие процессоростроения также ориентировано на *распараллеливание* операций, т.е. на выполнение процессором большего числа операций за такт. Ключевыми ступенями развития архитектуры процессоров стали гиперконвейеризация, суперскалярность, неупорядоченная модель обработки, векторное процессирование (технология *SIMD*), архитектура *VLIW*. Все ступени были ориентированы на повышение степени параллелизма исполнения.
- ▶ В настоящее время мощные сервера представляют собой мультипроцессорные системы, а в процессорах активно используется *параллелизм* уровня потоков.

# Уровни параллелизма

- ▶ Распараллеливание операций – перспективный путь повышения производительности вычислений. Согласно закону Мура число транзисторов экспоненциально растет, что позволяет в настоящее время включать в состав CPU большое количество исполнительных устройств самого разного назначения. Прошли времена, когда функционирование ЭВМ подчинялось принципам фон Неймана.
- ▶ В 70-е годы стал активно применяться принцип конвейеризации вычислений. Сейчас конвейер Intel Pentium 4 состоит из 20 ступеней. Такое распараллеливание на микроуровне – первый шаг на пути эволюции процессоров. На принципах конвейеризации базируются и внешние устройства. Например, динамическая память (организация чередования банков) или внешняя память (организация RAID).



# Уровни параллелизма

- ▶ Но число транзисторов на чипе росло. Использование микроуровневого параллелизма позволяло лишь уменьшать CPI (Cycles Per Instruction), так как миллионы транзисторов при выполнении одиночной инструкции простаивали. На следующем этапе эволюции в 80-е годы стали использовать параллелизм уровня команд посредством размещения в CPU сразу нескольких конвейеров. Такие суперскалярные CPU позволяли достигать  $CPI < 1$ . Параллелизм уровня инструкций (ILP) породил неупорядоченную модель обработки, динамическое планирование, станции резервации и т.д. От CPI перешли к IPC (Instructions Per Clock). Но ILP ограничен алгоритмом исполняемой программы. Кроме того, при увеличении количества ALU сложность оборудования экспоненциально растет, увеличивается количество горизонтальных и вертикальных потерь в слотах выдачи. Параллелизм уровня инструкций исчерпал свои резервы, а тенденции Мура позволили процессоростроителям осваивать более высокие уровни параллелизма. Современные методики повышения ILP основаны на использовании процессоров класса SIMD. Это векторное процессирование, матричные процессоры, архитектура VLIW.



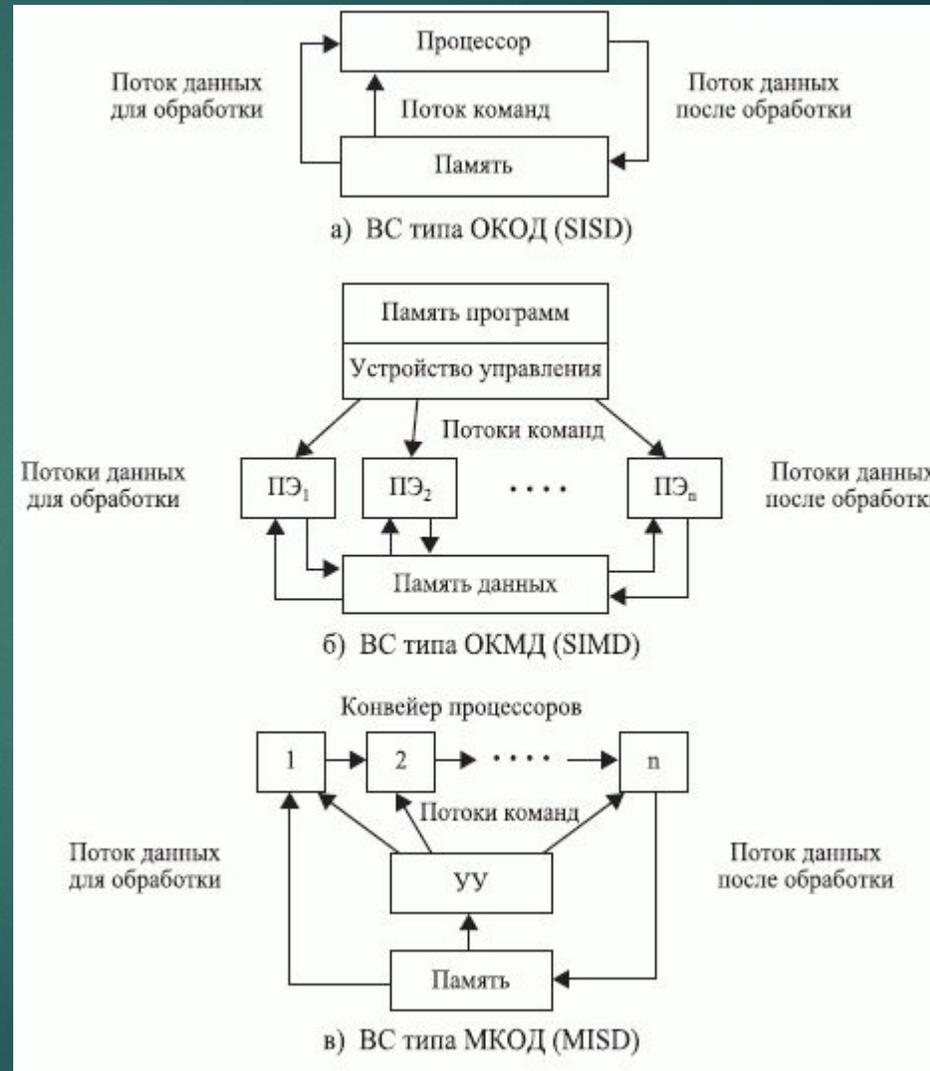
# Уровни параллелизма

- ▶ Параллелизм уровня потоков и уровня заданий применяется в процессорах класса MIMD. Многопоточные процессоры позволяют снижать вертикальные потери в слотах выдачи, а Simultaneous Multithreading (SMT) процессоры – как вертикальные, так и горизонтальные потери. Закон Мура обусловил также выпуск многоядерных процессоров (CMP). Лучшие современные вычислители – это мультикомпьютерные мультипроцессорные системы.
- ▶ Параллелизм всех уровней свойственен не только процессорам общего назначения (GPP), но и процессорам специального назначения (ASP (Application-Specific Processor), DSP (Digital Signal Processor)).
- ▶ Иногда классифицируют параллелизм по степени гранулярности как отношение объема вычислений к объему коммуникаций. Различают мелкозернистый, среднезернистый и крупнозернистый параллелизм. Мелкозернистый параллелизм обеспечивает сам CPU, но компилятор может и должен ему помочь для обеспечения большего IPC. Среднезернистый параллелизм – прерогатива программиста, которому необходимо разрабатывать многопоточные алгоритмы. Здесь роль компилятора заключается в выборе оптимальной последовательности инструкций (с большим IPC) посредством различных методик (например, символическое разворачивание циклов). Крупнозернистый параллелизм обеспечивает ОС.



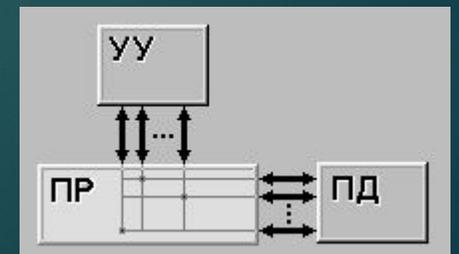
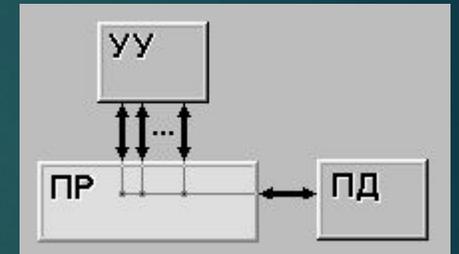
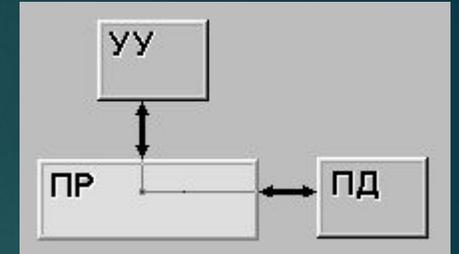
# Классификации Флинна

- ▶ Известно более десятка различных классификаций вычислительных систем. Однако ключевой является классификация Флинна, на базе которой возникли новые классификации, детализирующие ее исходные классы.
- ▶ В 1966 году Флинн предложил рассматривать и потоки команд, и потоки данных либо как одиночные, либо как множественные. Так появились четыре класса вычислительных систем: *SISD*, *MISD*, *SIMD*, *MIMD*.



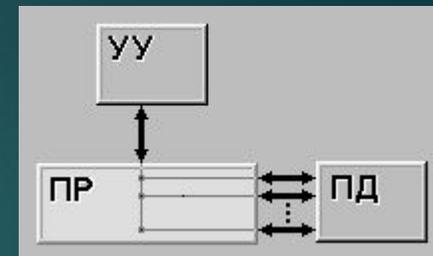
# Классификации Флинна

- ▶ Классические фон-неймановские машины попадают в тривиальный класс *SISD*, в котором одиночный поток инструкций обрабатывает одиночный поток данных. Классические языки высокого уровня (такие, как C++) также ориентированы на программирование в классе *SISD*. В настоящее время выпуск *SISD* процессоров почти прекращен из-за их низкой производительности, которая обусловлена низким уровнем параллелизма вычислений (используется только мелкозернистый параллелизм).
- ▶ Класс *MISD*, в котором множественный поток инструкций обрабатывает одиночный поток данных, пуст (бессмысленен).
- ▶ Все современные передовые процессоры, как общего, так и специального назначения, попадают в класс *MIMD*. Они одновременно исполняют сразу несколько независимых потоков инструкций, аппаратно обеспечивая крупнозернистый параллелизм. Изъян классификации Флинна заключается в неразличимости современных направлений процессоростроения, что привело к возникновению множества новых классификаций.



# Классификации Флинна

- ▶ Чистых представителей класса *SIMD* совсем немного. Класс ориентирован на выполнение программ, для которых характерна обработка больших регулярных массивов чисел. Здесь одиночный поток инструкций обрабатывает множественный поток данных. Именно представители класса *SIMD* впервые достигли производительности порядка *GFLOPS*.
- ▶ Наиболее популярная идея класса *SIMD* – векторное процессирование. Векторный процессор поддерживает обработку не только скалярных, но и векторных операндов. Эффективное декодирование инструкций и удобные данные сказываются на производительности крайне позитивно. Поэтому векторную обработку внедряют и в процессоры классов *SISD* и *MIMD*. Например, *SIMD* расширение IA32 – технологии *MMX* и *SSE*.
- ▶ Другим примером инкапсуляции техники *SIMD* является процессор *STI Cell* (альянс Sony, Toshiba и IBM). Процессор является гибридным объединением ведущего процессора (*Power Processor Element, PPE*) и восьми векторных сопроцессоров (*Synergistic Processor Element, SPE*). В качестве ведущего процессора используется 64-битный 2-поточный *RISC* процессор от IBM, традиционно включающий векторное расширение (*Vector Multimedia eXtensions, AltiVec*). *SPE* представляют собой параллельно работающие 128-битные *SIMD* процессоры.



# Класс SIMD

- ▶ Основные представители класса SIMD: векторные процессоры, матричные процессоры и процессоры с архитектурой VLIW.
- ▶ У матричных процессоров нет аналогии с векторными. Матричный процессор – это массив процессоров с единым потоком команд. Большой исходный массив данных разделяют на части, подлежащие идентичной обработке. Каждый процессор массива обрабатывает соответствующую часть данных, выполняя единый поток инструкций.
- ▶ Перспективным представителем класса SIMD является архитектура VLIW (очень длинное командное слово). Одна инструкция в такой системе команд представляет собой кортеж из нескольких RISC инструкций, которые независимы по данным между собой. VLIW процессору не нужно проверять инструкции кортежа на выявление структурных зависимостей, зависимостей по данным или по управлению. Теперь эти функции возложены на компилятор. Процессор сразу может переходить к фазе исполнения. Такие блоки, как динамический планировщик, станции резервации и т.д., здесь упразднены. Высвободившиеся ресурсы (транзисторы) перераспределяются для повышения производительности системы (увеличиваются размеры кешей, буферов BTB и TLB). Таким образом, процессор и компилятор обеспечивают хороший уровень параллелизма команд. Один из самых мощных VLIW процессоров – Intel Itanium 2.

# Класс MIMD

- ▶ Представителей класса MIMD можно разделить на системы с общей (tightly coupled) или распределенной (loosely coupled) памятью.
- ▶ Системы с общей памятью (мультипроцессоры) строятся посредством увеличения количества процессоров в машине. Такая мультипроцессорная система симметрична по процессорам. Т.е. задание, вытесненное на одном из предыдущих квантов времени, может быть возобновлено на любом из процессоров. С точки зрения ОС каждый процессор – обычный ресурс, который перераспределяется между заданиями (потоками). Безусловно, ОС стремится к минимизации частоты смены процессора для исполняемых программ. Это позволяет наиболее эффективно использовать ресурсы каждого из процессоров (кешей, буфера ВТВ, буфера TLB и т.д.). Такой режим симметричного мультипроцессорирования (Symmetric Multiprocessor, SMP) поддерживается во всех современных ОС.
- ▶ Количественные изменения по закону Мура в настоящее время привели к возможности размещения на одном чипе сразу нескольких процессоров (Chip Multiprocessor, CMP). Такой многоядерный (Multi-Core) процессор с точки зрения ОС выглядит как несколько одноядерных и допускает симметричное мультипроцессорирование. Иногда различные ядра имеют общий кеш второго или третьего уровней.
- ▶ Мультикомпьютерные системы (мультикомпьютеры) – системы с распределенной памятью. Такие системы представляют собой массив мощных серверов, объединенных в единый вычислительный ресурс при помощи высокопроизводительной коммуникационной сети (Massively Parallel Processing, MPP). Кластерные системы – упрощенный вариант MPP. Например, несколько персональных компьютеров в сети Ethernet плюс некоторый механизм распределения вычислительной нагрузки – это уже простейший кластер!

# Класс MIMD

- ▶ Итак, в настоящее время мощные центры вычислений базируются на крупнозернистом параллелизме и строятся либо как SMP, либо как MPP.
- ▶ Процессоры в SMP системе представляют собой монолитный ресурс. Программа может быть легко переброшена с перегруженного процессора на недогруженный процессор. Многопоточной программе динамически выделяется то или иное количество процессоров в зависимости от загруженности системы в целом. Всплески вычислительной нагрузки легко перераспределяются планировщиком ОС. Более того, даже зависание одной или нескольких программ не приводит ОС в режим голодания! Но такие позитивные черты обусловлены общей памятью, к которой предъявляются очень высокие требования по производительности. Именно память ограничивает масштабируемость SMP.
- ▶ В MPP частая передача вычислительной нагрузки между серверами противопоказана, так как приводит к медленным процедурам внешнего ввода/вывода. Фактически, это не единый мощный ресурс. Это объединение ресурсов для решения той или иной задачи, требующее программирования специального вида. Требование специальной организации вычислений вынуждает либо существенно модифицировать имеющееся программное обеспечение, либо разрабатывать его вновь, что, безусловно, затрудняет переход от SMP к MPP. Ключевым преимуществом MPP является отличная масштабируемость, которая ограничивается только предельными параметрами коммуникационной сети.

# Классификации Флинна

- ▶ Классификация Флинна широко используется и сегодня для начального описания вычислительных систем.
- ▶ У этой классификации есть очевидные недостатки:
- ▶ в нее четко не вписываются отдельные наследные применения архитектуры. Например, векторно-конвейерные компьютеры и компьютеры, управляемые потоками данных;
- ▶ класс MIMD очень перегружен: в него вошли все многопроцессорные системы. При этом они существенно отличаются по ряду признаков (числом процессоров, природе и топологией и видами связей между ними, способами организации памяти и технологиями программирования).
- ▶ Несколько классификаций, предложенных позже, расширяют классификацию М. Флинна. Примером такой классификации может служить классификация Ванга и Бригса.

# Классификация Ванга и Бриггса

Эта классификация по сути, является дополнением к классификации Флинна. В ней сохранены четыре базовых класса (*SISD*, *SIMD*, *MISD*, *MIMD*), К. Ванг и Ф. Бриггс внесли следующие изменения.

В классе *MIMD* выделяются:

- ▶ вычислительные системы со слабо связанными процессорами,
- ▶ вычислительные системы с сильной связанными процессорами.

К первой группе относятся системы с распределенной памятью, ко второй - системы с общей памятью.

Класс *SISD* делится на два подкласса:

- ▶ архитектуры, имеющие одно функциональное устройство;
- ▶ архитектуры, в состав которых входит несколько функциональных устройств.

Класс *SIMD* с учетом способа обработки данных делится на два подкласса:

- ▶ архитектуры с разрядно-последовательной обработкой данных;
- ▶ архитектуры с пословно-последовательной обработкой данных.

# Классификация Фенга

Т. Фенг предложил в основу классификации вычислительных систем положить две простые характеристики:

- ▶ число бит  $n$  в машинном слове, которые обрабатываются параллельно при выполнении машинных команд;
- ▶ числу слов  $m$ , одновременно обрабатываемых вычислительной системой. Используя эту терминологию работу любого компьютера можно интерпретировать как параллельную обработку  $n$  битовых слоев. В каждом слое независимо преобразуются  $m$  бит. При такой интерпретации, вторую характеристику называют шириной битового слоя.

Используя предельные верхние значения числа бит  $n$  и числа слов  $m$ , вычислительную систему можно охарактеризовать двумя числами  $(n, m)$ . Величина  $n * m$  определяет интегральную характеристику потенциала параллельности  $P$  архитектуры. Эта характеристика называется максимальной степенью параллелизма вычислительной системы:  $P(S) = m * n$ . По сути, это значение характеризует пиковую производительность. Рассматривая в качестве классификационного признака вычислительных систем способ обработки информации, заложенный в их архитектуру, введенные понятия позволяют разделить все вычислительные системы на следующие классы.

- ▶ Разрядно-последовательные, пословно-последовательные  $(n = m = 1)$ .
- ▶ Разрядно-параллельные, пословно-последовательные  $(n > 1, m = 1)$ . К этому классу можно отнести большинство классических последовательных компьютеров.
- ▶ Разрядно-последовательные, пословно-параллельные  $(n = 1, m > 1)$ .
- ▶ Разрядно-параллельные, пословно-параллельные  $(n > 1, m > 1)$ . К этому классу относится большинство современных параллельных вычислительных систем.

# Классификация Фенга

Эта классификация имеет ограничения. Они связаны со способом вычисления ширины битового слоя  $m$ .

В соответствии с этой классификацией отсутствуют различия между многопроцессорными системами, векторно-конвейерными компьютерами и процессорными матрицами.

Данная классификация не позволяет понять специфику той или иной высокопроизводительной вычислительной системы.

Достоинством классификации Фенга является введение единой числовой характеристики для всех типов вычислительных систем, которая позволяет сравнивать их между собой.

Из интересных видов классификации можно отметить подход Базу, который строит классификацию по последовательности решений, принимаемых на этапе проектирования архитектуры. Согласно А. Базу (A. Basu), любую параллельную вычислительную систему можно однозначно описать последовательностью решений, принятых на этапе ее проектирования, а сам процесс проектирования представить в виде дерева.

В корне "дерева Базу" размещается вычислительная система, последующие ярусы дерева служат для описания иерархии принятия решений при проектировании вычислительной системы. В итоге формируется описание проектируемой системы, представляемое значениями классификационных признаков в системе Базу.

Таким образом, на основе приведенных данных можно утверждать:

- ▶ Ни одна из существующих классификаций не является полной по системе классификационных признаков и, как правило, не дает однозначного ответа на вопрос: "К какому классу относится конкретная вычислительная система?".
- ▶ Классификация Флинна на настоящее время является базовой для остальных классификаций, которые ее детализируют, отражая другие, более узкие отличия в архитектуре.