

# Архитектура ЭВМ и систем

## *Лекция № 10*

Вычислительные системы  
параллельной обработки.  
Многопроцессорные и многоядерные  
системы

# План лекции

1. Параллельная обработка данных.
2. Классификация систем параллельной обработки данных.
  - 2.1. Классификация Флинна.
  - 2.2 . Классификация Головкина.
  - 2.3. Классификация многопроцессорных систем по способу организации памяти
3. Вычислительные системы на кристалле. Многоядерные системы.
4. Тенденции развития ВС.

# Вычислительные системы параллельной обработки. Многопроцессорные и многоядерные системы

В условиях постоянно возрастающих требований к производительности вычислительной техники всё более очевидными становятся ограничения фон-неймановских ВМ. Дальнейшее развитие вычислительной техники связано с переходом к параллельным вычислениям как в рамках одной ВМ, так и путём создания многопроцессорных систем и вычислительных сетей, объединяющих большое количество отдельных процессоров или отдельных ВМ. Для такого подхода используется термин «вычислительная система».

Отличительной особенностью вычислительных систем является наличие в них средств, реализующих параллельную обработку за счёт построения параллельных ветвей в вычислениях. Кроме того, возможности современной микроэлектроники позволяют размещать блоки вычислительных систем на одном кристалле, что позволяет снизить стоимость, повысить быстродействие и надёжность вычислительных систем .

**Вычислительная система (ВС)** – это взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для решения определённых классов задач.

# Параллельная обработка данных

Параллельная обработка может выполняться на нескольких уровнях:

1. *Уровень заданий.* Несколько независимых заданий одновременно выполняются на разных процессорах, практически не взаимодействуя друг с другом. Этот уровень реализуется в ВС со множеством процессоров в многозадачном режиме.
2. *Уровень программ.* Части одной программы выполняются на множестве процессоров, что достигается в параллельных ВС.
3. *Уровень команд.* Выполнение команды разделяется на фазы, а фазы нескольких последовательных команд могут быть перекрыты за счёт конвейеризации. Этот уровень достижим в ВС с одним процессором.
4. *Уровень битов.* Если биты слова обрабатываются одновременно, то говорят о бит-параллельной операции, что достижимо в обычных и суперскалярных процессорах.

Варианты параллельной обработки, а также конвейерные и суперскалярные процессоры были рассмотрены ранее (Лекция 4).

# Параллельная обработка данных

**Гранулярность** – это мера отношения объёма вычислений, выполненных в параллельной задаче, к объёму коммуникаций (для обмена сообщениями). В зависимости от степени гранулярности выделяют крупнозернистый, среднезернистый и мелкозернистый параллелизм.

- *Крупнозернистый параллелизм.* Характеризуется тем, что каждое параллельное вычисление достаточно независимо от остальных, причём требуется относительно редкий обмен информацией между отдельными вычислениями. Единицами распараллеливания являются большие и независимые программы, включающие тысячи команд. Этот уровень параллелизма обеспечивается ОС.
- *Среднезернистый параллелизм.* Единицами распараллеливания являются отдельные процедуры, включающие в себя сотни команд. Обычно организуется как программистом, так и компилятором.
- *Мелкозернистый параллелизм.* Характеризуется тем, что каждое параллельное вычисление достаточно мало и элементарно, составляется из десятков команд. Единицами распараллеливания обычно являются элементы выражения или итерации цикла, имеющие небольшие зависимости по данным. Характерной особенностью такого уровня параллелизма является приблизительное равенство интенсивности вычисления и обмена данными. Мелкозернистый параллелизм реализуется распараллеливающим компилятором.

# Параллельная обработка данных

Преимущества параллельных архитектур:

- 1) *применение мультипроцессорных архитектур* – это единственный способ построения высокопроизводительных ВС. Производительность является важнейшим показателем ВС. Построив даже самый мощный процессор для данного уровня развития вычислительной техники, можно ещё больше увеличить производительность, объединив в единую систему несколько процессоров;
- 2) *мультипроцессорная система является масштабируемой*, т.е. модульная организация таких систем позволяет увеличивать производительность за счёт наращивания модулей и (или) совершенствования взаимосвязей между ними;
- 3) *мультипроцессорная система обладает большей отказоустойчивостью за счёт использования структурной избыточности*. Например, при выходе какого-либо процессора из строя ВС может продолжать работу, перераспределив вычислительную работу между оставшимися процессорами.

# Параллельная обработка данных

Однако, несмотря на отмеченные преимущества многопроцессорных систем, им присущ и ряд недостатков, свойственных системам параллельной обработки:

- 1) *в параллельных системах* возникают новые, ранее не известные типы ошибок: тупики (взаимные блокировки процессов), голодание (невозможность получить ресурс) и т.п.;
- 2) *сложности понимания и анализа параллельных процессов.* Мышление людей имеет последовательный характер, поскольку мысли выражаются словами;
- 3) *недостаточная разработанность технологии параллельного программирования,* поскольку необходимо создание новых ОС, языков программирования, трансляторов, прикладных программ, наиболее полно использующих возможности параллельных архитектур.

В истории развития ВС, начиная с 70-х гг. 20 века, отмечались и рост и снижение интереса к параллельным архитектурам. В связи с успешными реализациями параллельных систем, с конца 90-х гг. 20 века и начала 21 века параллельные архитектуры являются перспективным направлением развития ВС.

# Параллельная обработка данных

В идеальном случае система из  $n$  процессоров могла бы ускорить вычисления в  $n$  раз. Однако в реальности такого показателя достичь не удаётся по ряду причин, и, прежде всего, потому, что ни одну задачу невозможно распараллелить полностью. Кроме того, на практике добиться равномерной загрузки параллельно работающих процессоров достаточно трудно. Замечено, что с ростом размерности задач доля вычислений, допускающих распараллеливание, растёт, что в целом позволяет увеличить скорость вычислений при решении задач.

Параллельно выполняемые единицы обработки могут быть различной величины: от программ до команд и микроопераций. Глубина распараллеливания, а также способы организации отдельных элементов многопроцессорных систем легли в основу классификаций систем параллельной обработки.



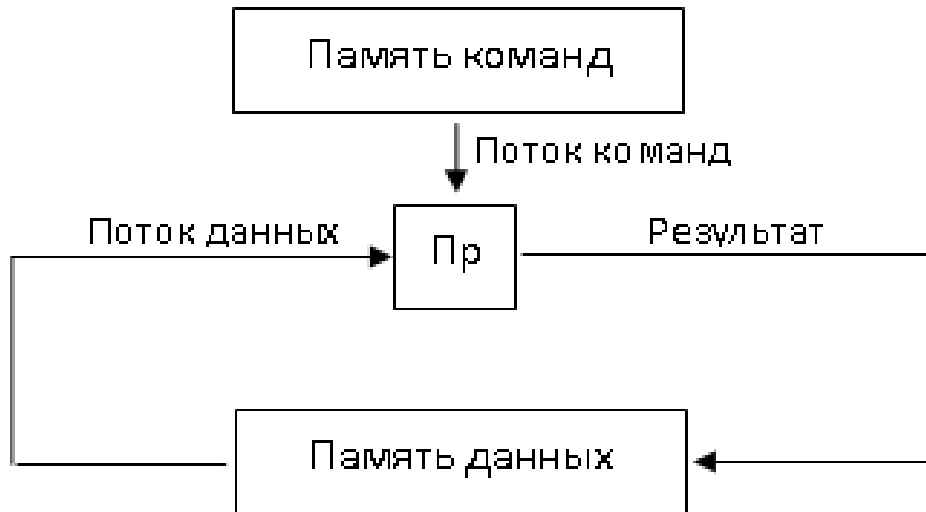
# Классификация систем параллельной обработки данных

Одной из первых является классификация Майкла Флинна (1966 год), которая основана на двух характеристиках организации процесса обработки: *потоке команд* и *потоке данных*.

Одиночность и множественность потоков определяется максимально возможным числом команд (или данных), одновременно находящихся на одинаковых стадиях исполнения.

В соответствии с классификацией Флинна, можно выделить четыре класса архитектур.

# Классификация систем параллельной обработки данных

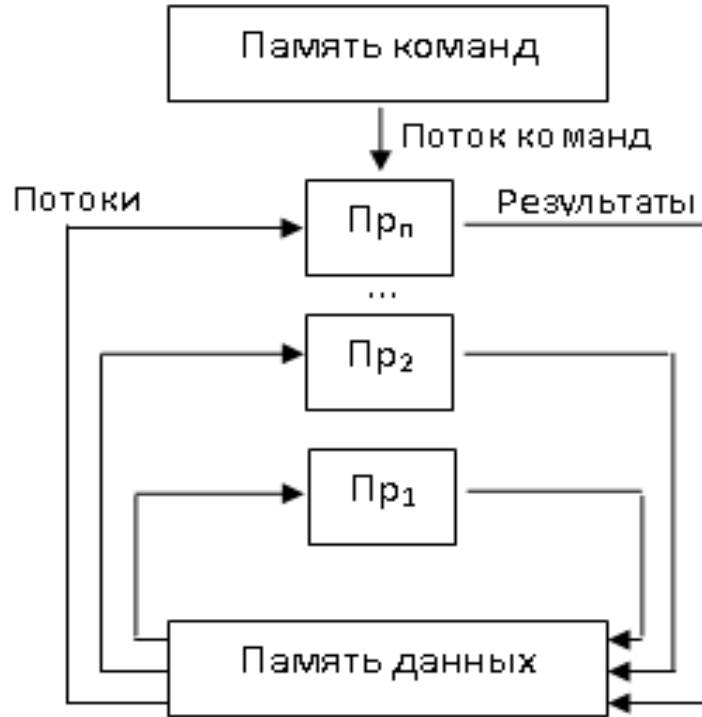


*Одиночный поток команд,  
одиночный поток данных  
(ОКОД) или Single Instruction  
Stream Single Data Stream  
(SISD)*

Сюда относятся ВС с одним процессором, т.е., классические фон-неймановские ВМ, в которых имеется только один поток команд, команды обрабатываются последовательно, и каждая команда инициирует одну операцию с одним потоком данных.

То, что для увеличения скорости обработка команд и скорости выполнения арифметических операций может применяться конвейерная обработка, значения не имеет.

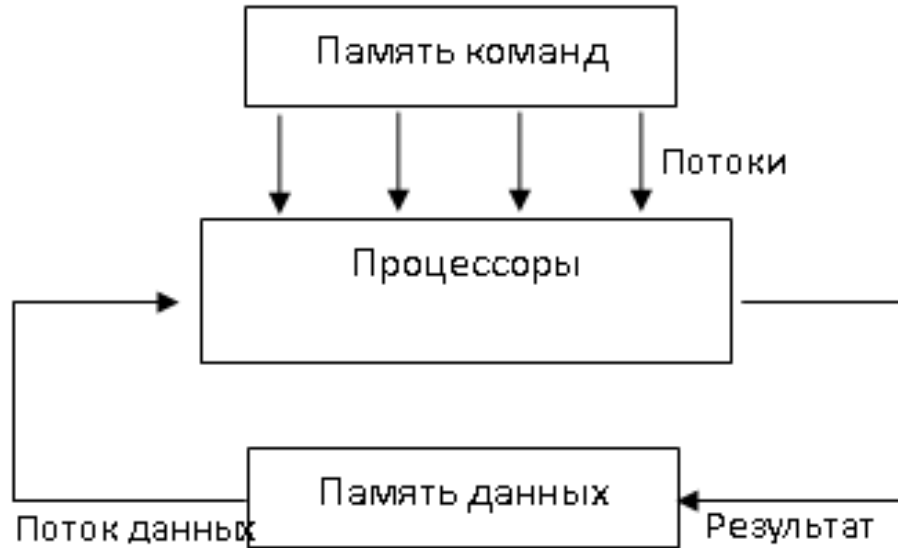
# Классификация систем параллельной обработки данных



*Одиночный поток команд,  
множественный поток данных  
(ОКМД) или Single Instruction  
Stream Multiple Data Stream  
(SIMD)*

Сюда относятся векторные ВС, в которых одинаковые преобразования выполняются над различными элементами данных (компонентами вектора). Представителями этого класса считаются матрицы процессоров, где единое УУ контролирует множество процессорных элементов. Все процессорные элементы получают от УУ одинаковую команду и выполняют её над своими локальными данными.

# Классификация систем параллельной обработки данных



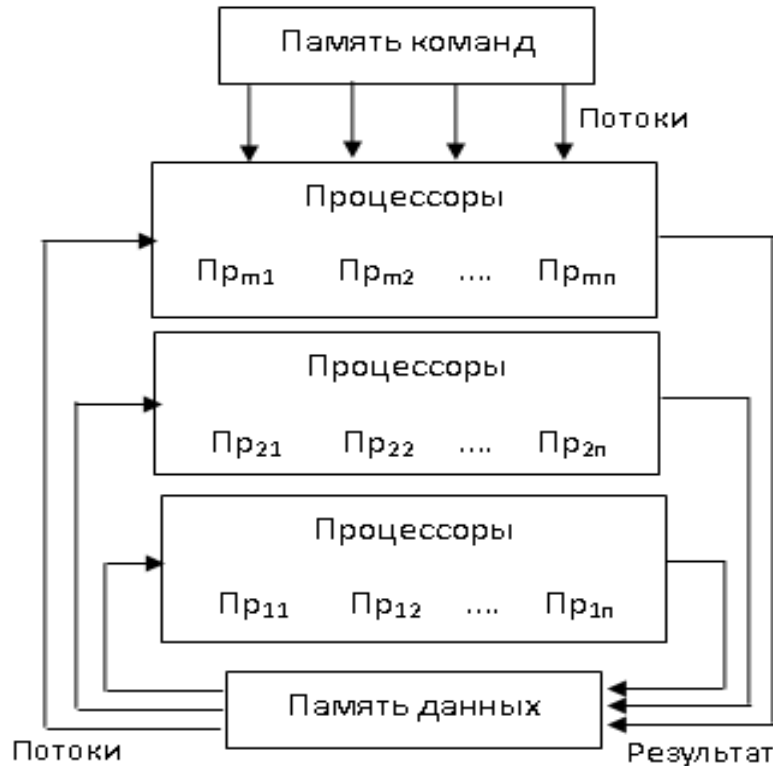
*Множественный поток команд, одиночный поток данных (МКОД) или Multiple Instruction Stream Single Data Stream (MISD)*

Из определения следует, что в таких ВС присутствует множество процессоров, обрабатывающих один и тот же поток данных. По поводу данного класса ВС нет однозначного мнения.

Согласно одним источникам, до сих пор специалисты в области архитектуры ВМ не сумели представить убедительный пример такой реально существующей ВС, и этот класс считается пустым.

В соответствии с другими источниками, сюда относятся конвейерные ВС, данные в которых обрабатываются за несколько стадий различными операционными устройствами под управлением соответствующих команд.

# Классификация систем параллельной обработки данных



Сюда относятся мультипроцессорные системы, в которых отдельные порции данных обрабатываются различными операционными устройствами, каждое из которых выполняет свою программу.

Данный класс ВС чрезвычайно широк, поскольку включает в себя разнообразные многопроцессорные системы.

*Множественный поток команд,  
множественный поток данных  
(МКМД) или Multiple Instructions  
Stream Multiple Data Stream  
(MIMD)*

# Классификация систем параллельной обработки данных

Классификация, предложенная профессором Б.А. Головкиным.  
Выделяют четыре уровня обработки: *уровень программ, уровень команд, уровень данных (регистровых передач), уровень слов.*

Для *программ* как единиц обработки в качестве средств обработки выступают вычислительные машины и сети.

Параллельно могут выполняться части программ (*команды*), которые выполняют процессоры. На уровне команд параллельно могут выполняться отдельные фазы (выборка команды, декодирование, выборка операндов, выполнение операции). Их выполняют функциональные блоки процессора.

*Данным*, обрабатываемым на уровне регистровых передач, соответствуют АЛУ.

Параллельно могут обрабатываться компоненты векторов, матриц и других структур данных (*слова*). Выполняют преобразования функциональные блоки АЛУ. Параллельно могут обрабатываться разряды слов функциональными преобразователями разрядов в составе ОБ АЛУ.

На каждом уровне обработки возможны четыре варианта организации процессов: последовательный, конвейерный, параллельный (многоэлементный), параллельно-конвейерный. Тогда всего возможно  $4^4 = 256$  классов обработки и соответствующих им типов архитектур.

# Классификация систем параллельной обработки данных

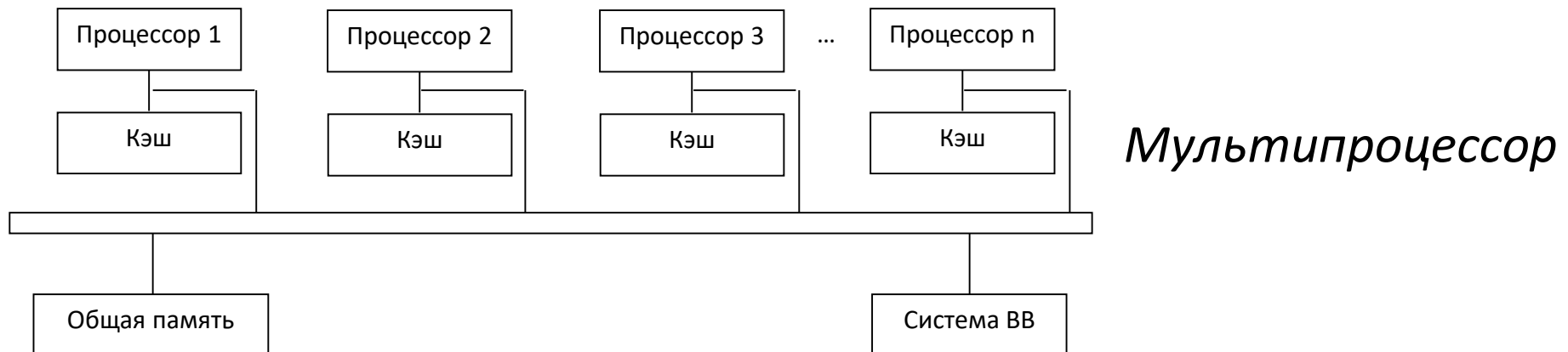
По способу организации основной памяти выделяют два типа архитектур:

- *многопроцессорные системы с общей памятью (мультипроцессоры) и*
- *многопроцессорные системы с распределённой памятью (мультикомпьютеры).*

# Классификация систем параллельной обработки данных

В **мультипроцессоре** все процессоры совместно используют общую физическую память. Любой процессор может считать информацию из памяти или записать в память с помощью соответствующих команд чтения и записи. В мультипроцессорах функционирует только одна копия ОС, поскольку адресное пространство является единым. При этом процессор не просто решает часть задачи, но и может оперировать с ней целиком.

Помимо всего, в состав мультипроцессоров также входят УВВ (диски, сетевые адаптеры, ПУ и т.д.). В одних мультипроцессорах только определённые процессоры получают доступ к УВВ, в других – каждый процессор имеет доступ к любому УВВ.





# Классификация систем параллельной обработки данных

Если все процессоры имеют равный доступ ко всем модулям памяти и УВВ, и между процессорами возможна полная взаимозаменяемость, такие мультипроцессоры называют **симметричными** – *SMP (Symmetric Multi-Processors)*.

В больших мультипроцессорах память обычно делится на несколько модулей. По способу доступа к общей памяти можно выделить следующие разновидности мультипроцессоров:

1. *системы с однородным доступом к памяти – UMA (Uniform Memory Access),*
2. *системы с неоднородным доступом к памяти – NUMA (Non Uniform Memory Access) и*
3. *системы с доступом только к кэш-памяти – COMA (Cache Only Memory Access).*

# Классификация систем параллельной обработки данных

В **UMA-системах** каждый процессор имеет одно и то же время доступа ко всем модулям памяти.

Самые простые мультипроцессоры имеют всего одну шину, которая используется для взаимодействия процессорами и модулями памяти. При этом в каждый момент времени взаимодействовать могут только два устройства: один из процессоров с одним из модулей памяти.

При увеличении числа процессоров нагрузка на шину увеличивается, многим процессорам большую часть времени приходится простаивать, что ведёт к снижению общей производительности ВС.

Частично эту проблему решает наличие кэшей, но возникает проблема отслеживания идентичности данных в кэшах и основной памяти, так называемая *когерентность кэшей*.

# Классификация систем параллельной обработки данных

В **NUMA-системах** также имеется единое адресное пространство для всех процессоров, но, в отличие от UMA-систем, доступ к локальным модулям памяти осуществляется быстрее, чем к удалённым. Но в этом случае будет иметь значение порядок размещения данных.

Для эффективного использования неравномерного доступа целесообразно наиболее часто используемые данные размещать в локальных модулях, и постоянно следить за статистикой использования данных.

При использовании кэшей возникает та же проблема их согласованности, что и в UMA-системах.

# Классификация систем параллельной обработки данных

В **СОМА-системах** всё адресное пространство делится на строки кэша, которые по запросу свободно перемещаются в системе.

В этом случае количество кэш-попаданий увеличивается, что ведёт к повышению производительности ВС. Однако это порождает ряд проблем, связанных с размещением и удалением строк кэша.

При попытке перевести блок данных в полный кэш возникает необходимость переместить другой уже хранящийся там блок для создания необходимого свободного пространства.

Поскольку глобальная память отсутствует, то перемещаемый блок размещается в КЭШе другого процессора. Эти данные не будут использоваться тем процессором, что приводит к уменьшению эффективности использования системы.

# Классификация систем параллельной обработки данных

Таким образом, общая память способствует ускорению выполнения единой программы параллельно работающими процессорами за счёт их быстрого доступа к общим данным.

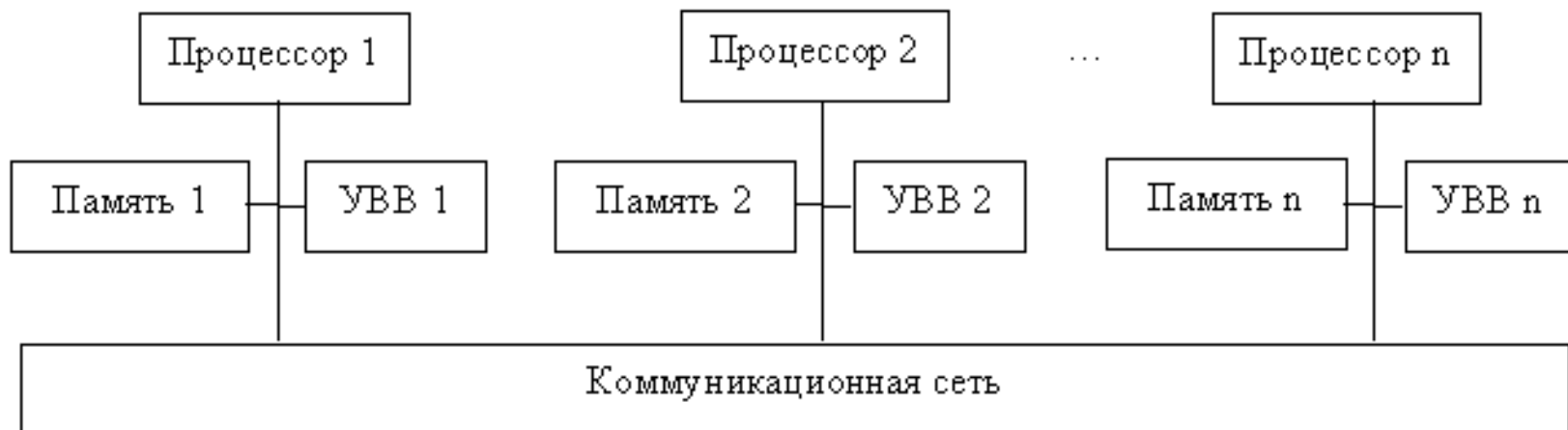
Но в то же время общая память является «узким местом» многопроцессорной системы, так как с ростом числа процессоров их быстрый доступ к ней становится затруднительным.

ВС с общей памятью называют *сильно связанными*.

# Классификация систем параллельной обработки данных

В **мультикомпьютерах** каждый процессор обладает своей собственной памятью, доступной только этому процессору, и может иметь свои устройства для ввода и вывода информации.

Таким образом, мультикомпьютер с помощью команд чтения и записи может обращаться только к своей локальной памяти. Для взаимодействия между собой процессоры в мультикомпьютере обмениваются сообщениями через высокоскоростную коммуникационную сеть.



*Мультикомпьютер*

Разработал: Конюхова О.В.

# Классификация систем параллельной обработки данных

ПО мультимпьютера имеет более сложную структуру, чем ПО мультипроцессора.

Кроме того, в мультимпьютерах основной проблемой становится правильное распределение данных, поскольку их размещение влияет на производительность мультимпьютеров (при частых обращениях к локальным памяти других процессоров общая производительность мультимпьютера снижается).

Программировать мультимпьютеры значительно сложнее, чем мультипроцессоры, однако затраты на их реализацию гораздо меньше затрат на создание мультипроцессоров.

Более того, мультимпьютеры довольно легко масштабируются без потери в производительности.

Процессоры могут быть связаны между собой в систему различными способами, что зависит от способа реализации коммуникационной сети.

# Классификация систем параллельной обработки данных

Мультикомпьютеры подразделяются на две категории.

К первой относятся **системы с массовым параллелизмом** - *MPP* (*Massive Parallel Processing*).

**MPP-системы** – это огромные суперкомпьютеры стоимостью в несколько миллионов долларов. Они используются в различных отраслях науки и техники для выполнения сложных вычислений, обработки большого числа транзакций в секунду, управления большими базами данных, и т.д. В большинстве MPP-систем используются стандартные процессоры: например, Intel Pentium, Sun UltraSPARC, IBM RS/600 и DEC Alpha.

MPP-системы поставляются вместе с дорогостоящим ПО и библиотеками. Ещё одной характеристикой MPP-систем являются огромные объёмы ввода – вывода (до нескольких терабайт).

MPP-системы должны обладать высокой отказоустойчивостью, поэтому в них всегда имеется аппаратная и программная поддержка мониторинга системы обнаружения неполадок и их исправления.

К MPP-системам можно отнести суперкомпьютер IBM BlueGene фирмы «IBM», разработанный для решения задач большой сложности в биологии; суперкомпьютер Red Storm для решения заданий, выдаваемых департаментом энергетики США, и ряд других .



# Классификация систем параллельной обработки данных

Вторая категория мультимпьютеров включает обычные ПК или рабочие станции (иногда смонтированные в стойки), которые связываются коммуникационной сетью определённой топологии.

Принципиальное отличие COW-систем от MPP-систем заключается в том, что все вычисления в MPP-системах выполняются гораздо быстрее, из-за чего управляются и применяются они совершенно по-разному.

Эти «доморощенные» системы называют **кластерами рабочих станций** – COW (*Cluster Of Workstation*). Наиболее ярким примером является кластер Google .

# Классификация систем параллельной обработки данных

Поскольку системы с общей памятью эффективно реализуют сильно связанные программы, а системы с передачей сообщений могут иметь большое число процессоров, то обе архитектуры желательно использовать совместно.

Примером реализации такого подхода являются гибридные **кластерные системы**, т.е. системы, содержащие группы (кластеры) процессоров с относительно высокой связностью, и имеющие более низкую степень связности с процессорами из других групп.

# Вычислительные системы на кристалле.

## Многоядерные системы

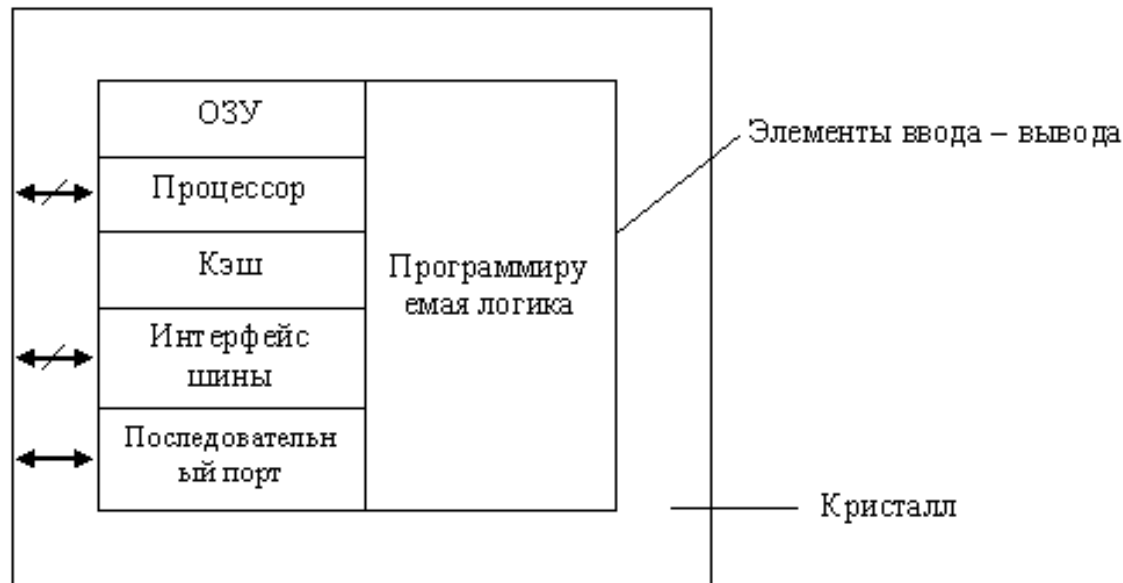
Возможность реализации всех блоков вычислительной системы на одном кристалле открывает новые возможности для разработчиков и позволяет улучшить общетехнические показатели системы: снизить стоимость, повысить быстродействие и надёжность.

Опыт разработки дискретных систем показывает, что в среднем стоимость соединения на кристалле в 10 раз ниже, чем на печатной плате; задержки в связях на печатной плате в 10 раз дольше, чем на кристалле; надёжность и помехозащищённость соединений на кристалле также существенно выше, чем на плате и в межплатных соединениях .

Можно размещать на кристалле как унифицированные блоки: процессор, память, контроллеры шин, так и специализированные устройства, разделяемые средствами программируемой логики на той же СБИС.

# Вычислительные системы на кристалле. Многоядерные системы

Общая структура системы на кристалле – ***System On Programmable Chip (SOPC)*** содержит следующие блоки: процессор, кэш-память, ОЗУ, интерфейс шины для связи с внешней памятью и другими устройствами, последовательный порт, поле программируемой логики для реализации специализированных блоков системы.



*Общая структура СБИС  
системы на кристалле*

# Вычислительные системы на кристалле. Многоядерные системы

Получили развитие два направления создания кристаллов SOPC: *однородные и блочные.*

В однородных кристаллах SOPC блоки конфигурируются в кристалле на базе программируемой логики. В таких системах используются готовые программы крупных блоков системы (процессора, контроллера шины и др.), так называемые «единицы интеллектуальной собственности». Проектировщики системы выбирают необходимые блоки и размещают их на кристалле.

В блочных кристаллах SOPC определённые области кристалла заранее выделены под аппаратные ядра, которые (ядра) реализованы с использованием полного цикла проектирования. Это позволяет уменьшить площадь кристалла под блоки и повысить его быстродействие. Недостатком является некоторое снижение гибкости системы, поскольку реализованные блоки нельзя изменить. Однако поле программируемой логики позволяет создавать специализированные блоки.

# Вычислительные системы на кристалле. Многоядерные системы

По мере роста интеграции параллельная обработка информации стала применяться и на уровне мультимикропроцессорной обработки. Переход к мультимикропроцессорным системам на кристалле позволяет замедлить рост потребляемой мощности при увеличении ресурсов кристалла.

Мультимикропроцессорные системы на кристалле (многоядерные процессоры) обозначают аббревиатурой **CMP (Chip Multi-Processor)**.

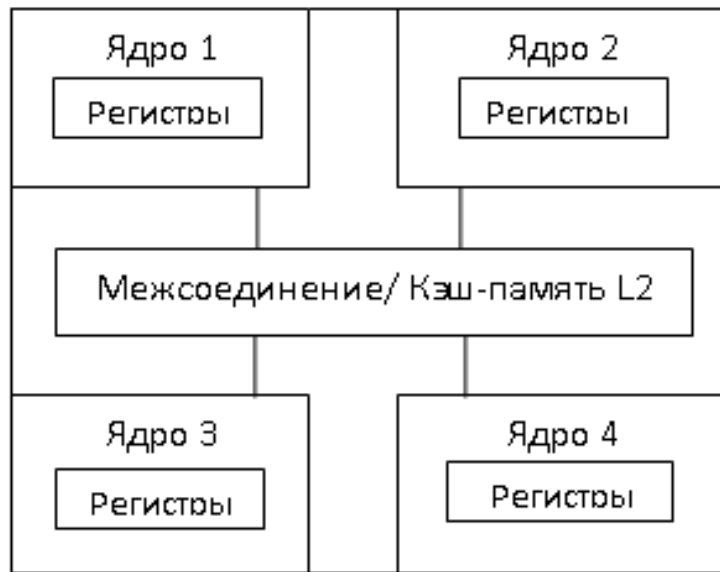
Применение полноценных микропроцессорных систем не получило широкого распространения, поскольку требует сложных и дорогостоящих микропроцессорных материнских плат.

Разработка многоядерных процессорных систем является перспективным направлением повышения производительности современных ВМ.

# Вычислительные системы на кристалле. Многоядерные системы

**Ядро** – это кристалл кремния, на котором посредством микроскопических логических элементов реализована принципиальная схема процессора (архитектура).

**Многоядерный процессор** – это центральный процессор, содержащий два и более вычислительных ядра на одном процессорном кристалле или в одном корпусе .



*Симметричная*

*многоядерная архитектура  
с общей кэш-памятью L2*

Среди SMP-систем получили развитие симметричные системы – ***Symmetric General Purpose cores*** – ***мультипроцессорные системы на кристалле с симметричными ядрами общего назначения.***

Симметричная архитектура предполагает однородность всех ядер и единообразие включения всех ядер в общую схему системы.

В симметричных конфигурациях между ядрами разделяется общая память, например, кэш L2.

# Вычислительные системы на кристалле. Многоядерные системы

Минимальное число ядер в таких системах равно двум. Достоинством является относительная простота разработки, интенсивное взаимодействие ядер, простота программирования.

Все ядра равноправно участвуют и в управлении вычислительным процессом, и в выполнении прикладных задач. Модули ОС выполняются в любом доступном ядре. Ресурсы выделяются для каждой выполняемой задачи по мере возникновения потребностей в них и никак не закрепляются за ядром.

В решении одной задачи могут участвовать сразу несколько ядер, если она допускает распараллеливание на несколько потоков.

В случае отказа одного из ядер, симметричные системы, как правило, сравнительно просто реконфигурируются, что является их большим преимуществом.

В то же время, такие системы плохо масштабируются: увеличение числа ядер ограничивается наличием разделяемой памяти.



# Вычислительные системы на кристалле. Многоядерные системы

Если рассматривать процесс как выполняющуюся программу с необходимыми ей для этого ресурсами (памятью, регистрами, стеком и т.п.), то **поток** (*тред*, от англ. Thread – нить) – это последовательность команд.

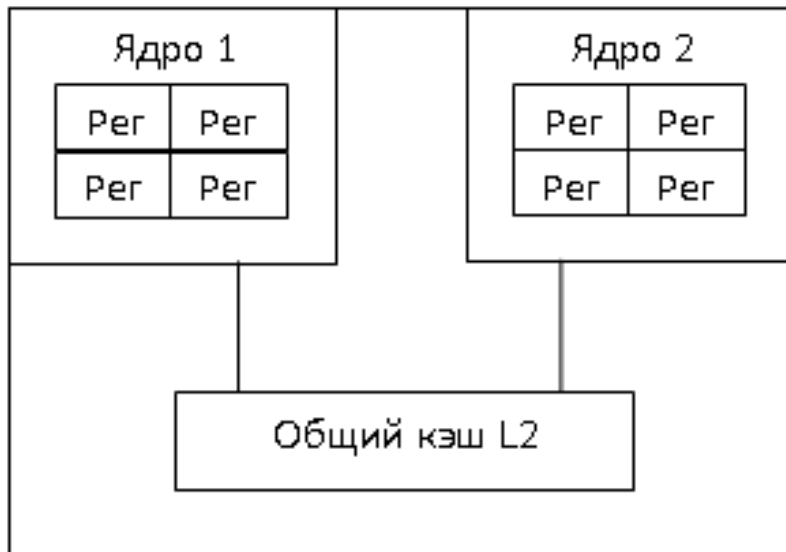
Несколько потоков могут существовать в рамках одного и того же процесса и совместно использовать его ресурсы.

То есть отдельная программа может быть разбита на фрагменты (потоки), содержащие последовательности команд, и эти потоки могут выполняться параллельно.

# Вычислительные системы на кристалле. Многоядерные системы

Применение различных способов организации внутрикристалльной памяти (регистров, кэшей) и механизмов динамического назначения команд на исполнение позволяет организовать параллельное выполнение нескольких тредов (поток) в одном процессорном ядре с совместным использованием ресурсов на кристалле.

На рисунке приведён план кристалла **многопоточковой мультиядерной системы с общим кэшем**. Такие системы называются **мультиплексорными системами на кристалле SMT (Simultaneous Multi-Threading)**.



*Многопоточковая  
многоядерная архитектура с  
общим кэшем*

# Вычислительные системы на кристалле. Многоядерные системы

В *ассиметричных кластерных мультипроцессорных системах АССМР (Asymetric Cluster CMP)* на кристалле размещается ассиметричный кластер процессорных ядер с различной специализированной аппаратурой.

Например, кластер может содержать RISC-ядро процессора, ядра специализированных процессоров для цифровой обработки сигналов, блоки памяти, средства коммуникаций.

Функциональная неоднородность в таких архитектурах влечёт за собой структурные различия во фрагментах системы, содержащих разные процессорные ядра и блоки.

Например, они могут отличаться схемами подключения процессорных ядер к СШ, способами взаимодействия процессорных ядер с устройствами и т.д.

Ассиметричные кластерные архитектуры являются хорошо масштабируемыми.

# Вычислительные системы на кристалле. Многоядерные системы

В ассиметричных архитектурах функции управления вычислительным процессом полностью централизованы, и ОС работает только на одном процессоре. Этот процессор является ведущим, а все остальные – ведомыми.

Ведущий берёт на себя функции распределения задач и ресурсов, а ведомые работают только как обрабатывающие устройства.

Ассиметричные архитектуры позволяют повысить производительность по сравнению с симметричными, но при этом требуется разработка специализированных ядер.

Кроме того, такая специализация снижает надёжность системы, поскольку процессоры не являются взаимозаменяемыми.

# Вычислительные системы на кристалле. Многоядерные системы

Можно выделить три варианта конструктивного расположения ядер на кристалле:

1. *Независимые процессорные ядра* (каждое со своей памятью) расположены на одном кристалле и для связи используют общую системную шину.
2. *Несколько одинаковых ядер* расположены на разных кристаллах, но объединены вместе в одном корпусе процессора (многочиповый процессор).
3. *Ядра могут быть тесно переплетены* между собой на одном кристалле и использовать некоторые общие ресурсы кристалла (например, шину и кэш-память).

# Вычислительные системы на кристалле. Многоядерные системы

Преимущества и недостатки мультиядерных процессоров вытекают из преимуществ и недостатков параллельных ВС.

Следует отметить, что в настоящее время многоядерные процессоры используются крайне неэффективно.

Прирост быстродействия во многом зависит от типа приложения. У программ, которые не рассчитаны на работу с многоядерными процессорами, быстродействие увеличивается всего на 5 %. А вот оптимизированные под многоядерные процессоры приложения работают быстрее уже на 50 %

# Тенденции развития ВС

Существует два направления развития высокопроизводительных вычислений.

1. Первое направление связано с применением дорогих суперкомпьютеров с уникальной архитектурой (специальные векторные процессоры, дорогостоящая сверхбыстрая память, уникальное периферийное оборудование), например, суперкомпьютеров CRAY, NEC, IBM, Эльбрус и т.д.
2. Второе направление связано с созданием ВС массового параллелизма. Основной упор при этом делается на увеличение числа процессоров и повышение степени параллелизма программ. При этом большую популярность приобретают COW-системы, которые постепенно вытесняют MPP.

# Тенденции развития ВС

Мультипроцессорные высокопроизводительные ВС находят своё применение в таких областях как исследовательская, промышленная, образовательная, административная, при производстве самих ВС. Такие системы требуются для точного прогнозирования погоды и климата, развития современного промышленного дизайна, создания новых фармацевтических средств и т.д.

Среди установленных мультипроцессорных систем системы с архитектурой MPP составляют подавляющее большинство (65 %), системы SMP – 21 %, системы с векторными процессорами – 12 %, кластерные системы – 2 %.

Доля кластерных систем постепенно возрастает.



# Тенденции развития ВС

Относительно многоядерных процессоров, лидерами по производству процессоров являются компании «Intel» и «AMD» продолжают наращивать количество ядер в процессорах.

Например, Intel планирует выпускать процессоры, содержащие до сотни ядер.

Появление мультиядерных процессоров, в свою очередь, стимулирует разработку ОС и прикладного программного обеспечения, поддерживающих многоядерность.

Кроме того, в корпорации «Intel» планируют выпускать многоядерный процессор с новой архитектурой, в составе которого будет находиться одно мощное центральное ядро, окружённое множеством вспомогательных ядер. Это поможет более эффективно обрабатывать сложные мультимедийные приложения в многопоточном режиме.

Кроме ядер общего назначения, процессоры будут обладать специализированными ядрами для выполнения различных классов задач: графика, алгоритмы распознавания речи и т.п.