

## ЛЕКЦИЯ N 7

### АРХИТЕКТУРА ЭВМ.

Каждая промышленная система базируется на строго определенном подходе к реализации процесса обработки информации. Термин архитектура в вычислительной технике употребляется для теоретической классификации способа обработки данных.

В широком смысле архитектура охватывает понятие принципов организации системы (аппаратно-программного комплекса), включающее такие высокоуровневые аспекты разработки компьютера как систему памяти, структуру системной шины, организацию ввода/вывода, систему команд, форматы данных, операционную систему и т.п. (рис.1)

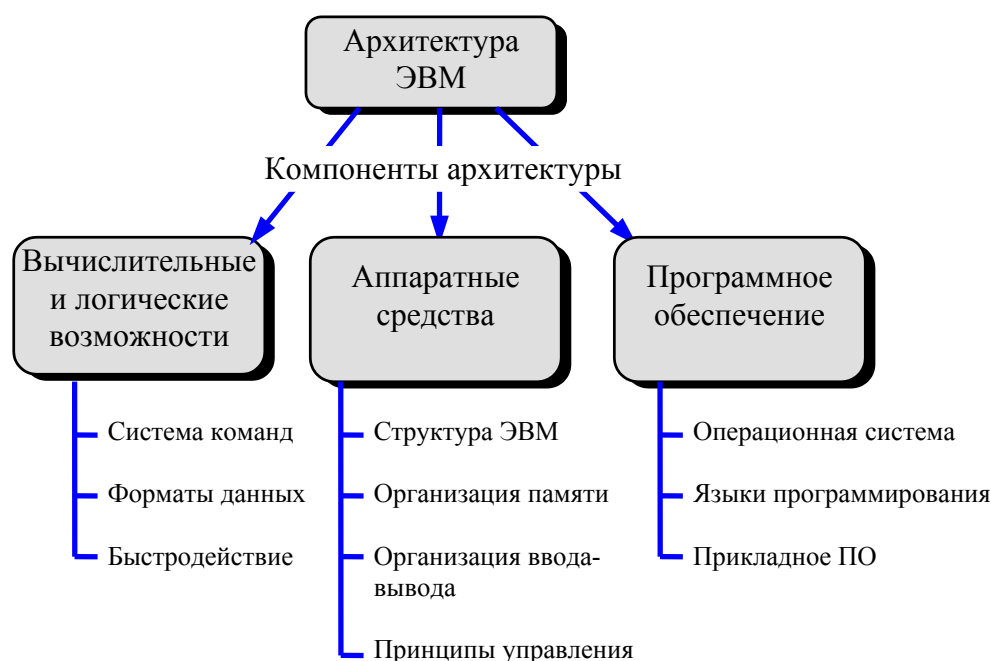


Рисунок 1 Общее понятие архитектуры ЭВМ

**Архитектура (architecture)** - концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов сложного объекта.

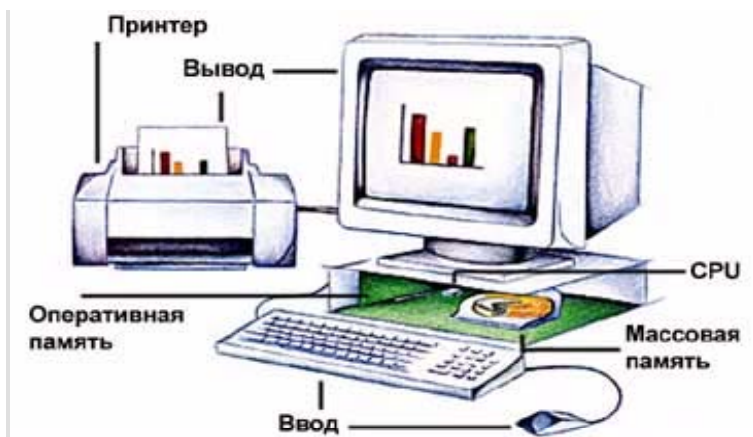
Архитектура вычислительной системы предполагает многоуровневую организацию. Архитектура набора команд служит границей между аппаратурой и программным обеспечением и представляет ту часть системы, которая видна программисту.

Основная масса современных вычислительных систем функционирует в соответствии с принципом фон Неймана - этот принцип называется "**управление потоком команд задаваемых программой**". Данные занимают подчиненное положение, последовательность и способ их обработки определяется командами программы, т.е. ход выполнения вычислительного процесса определяется только потоком команд. В простейшем случае процессор по очереди выбирает команды программы и также по очереди обрабатывает данные.

### Архитектура компьютера: модель фон Неймана

Чтобы собирать, сохранять и передавать двоичную информацию, компьютер строится по архитектурной аппаратной модели, впервые предложенной в 40-х годах математиком Джоном фон Нейманом. Эта модель, модель фон Неймана, разбивает все оборудование компьютерной системы на пять главных элементов:

- центральный вычислительный блок (CPU);
- устройства ввода;
- устройства вывода;
- память;
- массовое хранилище данных.



**Рисунок 2 Модель компьютерной архитектуры фон Неймана**

Каждый из этих элементов ассоциируется с конкретными физическими устройствами (аппаратурой) и каждый нуждается в программном обеспечении для совместной работы с остальными частями компьютерной системы. Модель фон Неймана легла в основу практически всех когда-либо построенных компьютеров - от занимавшего целый зал компьютера ENIAC до самых миниатюрных современных карманных PC.

Каким бы он ни был - микропроцессором Intel 386, 486 или Pentium в персональных компьютерах IBM и совместимых с ними, микропроцессором серии Motorola 68000 в компьютерах Apple Macintosh, RISC-процессором в PowerPC, процессором SPARC или MIPS в рабочих станциях UNIX, центральный вычислительный блок является "мозгом" любого современного компьютера.

Эта тончайшая пластинка кремния имеет самый сложный массив из сотен тысяч и даже миллионов электронных двухпозиционных переключателей, или транзисторов, вытравленных на его поверхности. Используя невидимые электрические потоки, CPU выполняет базовые команды, называемые инструкциями, которые переправляют информацию с места на место по своим микропроводам со скоростью, почти достигающей скорости света.

Современные CPU выполняют многие миллионы инструкций в секунду.

CPU обменивается инструкциями и данными с различными периферийными устройствами, такими как жесткие и гибкие дисковые накопители, экран дисплея, клавиатура, мышь, факс-модем, принтер или сканер. Эта периферия называется устройствами ввода/вывода и подключается к CPU и системной памяти через магистрали передачи данных, называемые шинами ввода/вывода.

Для работы CPU необходима информация. Чем быстрее он ее получит, тем быстрее сможет выполнить каждую полученную инструкцию. Однако CPU работает во много раз быстрее, чем любой дисковый накопитель или шина ввода/вывода. Поэтому во избежание заторов в потоках информации между CPU и остальной частью системы и за счет этого предоставления ему возможности работать на своей внутренней скорости, части операционной системы и используемых в настоящее время прикладных программ, а также данные, непосредственно с которыми и работают эти программы, записываются в специальную разновидность сверхбыстрой твердотельной памяти. Эта память получила название памяти с произвольным доступом (RAM).

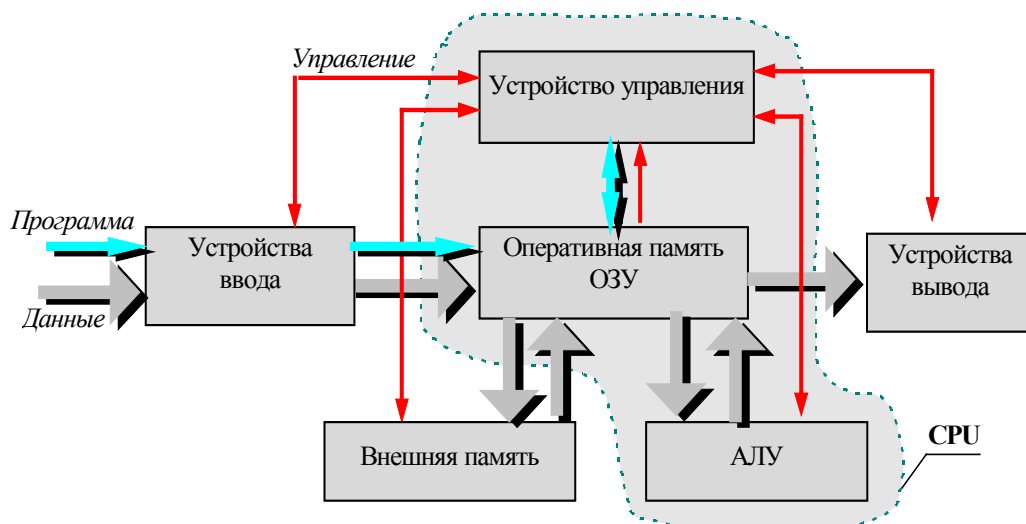


Рисунок 3. Поток программных команд и данных в ЭВМ.

В ЭВМ вводится информация двух типов:

1. Программа - набор команд, указывающая центральному процессору как нужно осуществлять решение задачи. Программа помещается в некоторую область памяти ЭВМ и используется только устройством управления.

2. Данные - это определенные факты, цифры, необходимые для решения конкретной задачи. Они направляются к различным устройствам внутри ЭВМ и обрабатываются в АЛУ (арифметико-логическом устройстве). Данные не нужны для устройства управления.

Структура ЭВМ (рис.1) представляет собой систему, состоящую из 5 основных функциональных блоков:

1. Устройство ввода информации - обеспечивает передачу от человека или первичных датчиков информации к ЭВМ. Здесь осуществляется кодирование информации с языка человека (аналоговый сигнал) на язык двоичных кодов для ЭВМ.

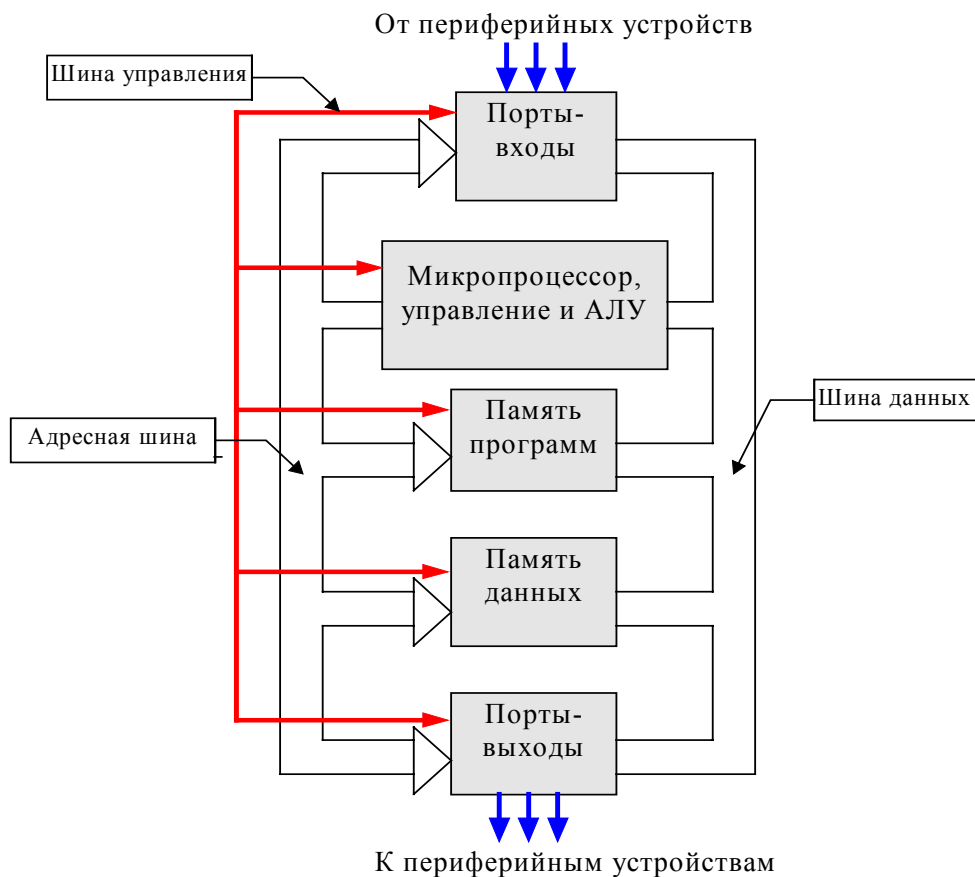
2. Память (внутренняя и внешняя) - хранилище данных и программ.

3. Арифметическое устройство (АЛУ) - складывает, вычитает, сравнивает, выполняет другие логические операции. АЛУ связано с ОЗУ двунаправленным потоком данных.

4. Устройство управления - последовательно считывает содержимое ячеек памяти, где находится программа и организует ее выполнение. Устройство управления регламентирует порядок работы всех остальных блоков ЭВМ. Порядок выполнения команд может быть изменен с помощью команд передачи управления (перехода) - это позволяет организовать циклы, ветвления и т.д., т.е. выполнять сложные программы.

5. Устройство вывода информации - связующее звено между машиной и человеком или исполнительным механизмом. Здесь осуществляется дешифрация с языка ЭВМ на язык понятный человеку (или аналоговый сигнал).

В вычислительной системе, состоящей из множества подсистем, необходим механизм для их взаимодействия. Эти подсистемы должны быстро и эффективно обмениваться данными с помощью центральной шины или системы нескольких шин.



**Рисунок 4 Шины ЭВМ**

В ЭВМ можно выделить несколько типов шин:

Адресная шина служит для выбора определенной ячейки памяти, порта ввода-вывода (16 линий).

Шина данных - двунаправленная пересылка данных к микропроцессору и от него (8 линий).

Шина управления - координирует работу всех устройств.

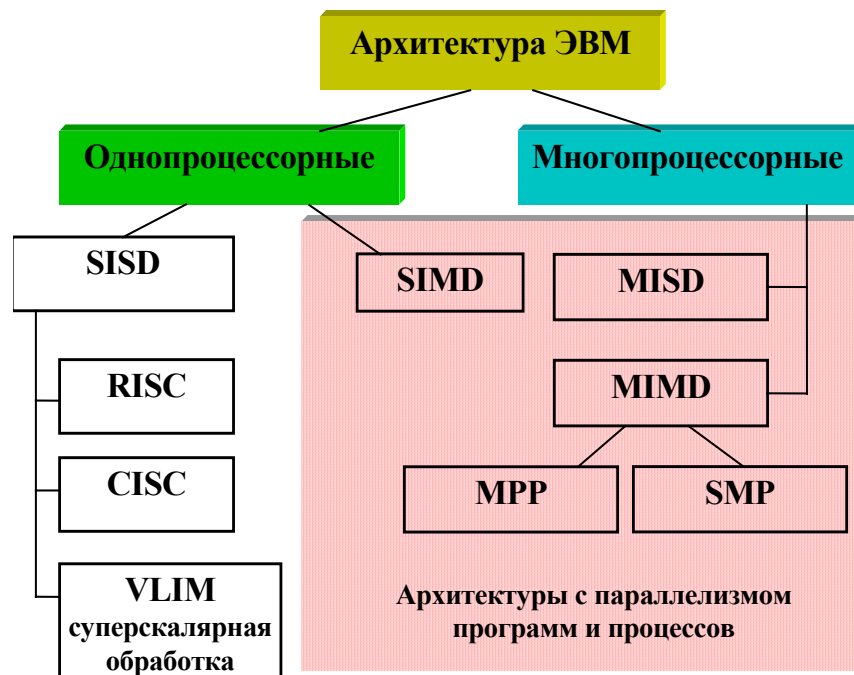
### Архитектура системы команд. Классификация процессоров

**Архитектура системы команд** это перечень команд, которые способен выполнить процессор ЭВМ, она служит границей между аппаратурой и программным обеспечением и представляет ту часть системы, которая видна программисту или разработчику компиляторов.

Система команд устанавливает, какие конкретно операции может выполнять процессор, каким должен быть формат команд. Количество команд сравнительно невелико - несколько десятков.

На современном этапе развития вычислительной техники компьютерной промышленностью используются две основные архитектуры набора команд - архитектуры CISC и RISC. Обе они относятся к однопроцессорным архитектурам - Single Instruction Single Date (SISD) - *один поток команд - один поток данных*.

CISC ( Complete Instruction Set Computer ) архитектура с полным набором команд - это архитектура фон Неймана, т.е. однопроцессорная архитектура со



**Рисунок 5 Классификация видов архитектуры ЭВМ**

строго последовательным выполнением команд. Принцип фон Неймана можно сформулировать так: **"управление потоком команд"**, так как стержень процесса образуется последовательностью (потоком) команд, заданных программой. Данные занимают подчиненное положение. Процессор по очереди выбирает команды программы и также по очереди обрабатывает данные - все строго последовательно.

Лидером в разработке микропроцессоров с полным набором команд CISC считается компания Intel со своей серией x86 и Pentium. Эта архитектура является практическим стандартом для рынка микрокомпьютеров. Для CISC-процессоров характерно: сравнительно небольшое число регистров общего назначения; большое количество машинных команд, некоторые из которых нагружены семантически аналогично операторам высокоуровневых языков программирования и выполняются за много тактов; большое количество методов адресации; большое количество форматов команд различной разрядности; преобладание двухадресного формата команд; наличие команд обработки типа регистр-память.

Основой архитектуры современных рабочих станций и серверов является архитектура компьютера с сокращенным набором команд (RISC - Reduced Instruction Set Computer). До появления RISC процессоров все машины ориентировались на выполнение команд обработки данных, сохраняемых в памяти. Причина такого подхода к организации обработки через ОЗУ в том, что при малой интеграции элементной базы трудно (или невозможно) организовать в центральном процессоре регистровую память большого объема. RISC позволяет осуществить глубокую конвейеризацию процесса вычислений. Поэтому производительность такого процессора существенно выше.

*Зачатки этой архитектуры уходят своими корнями к компьютерам CDC6600, разработчики которых (Торнтон, Крэй и др.) осознали важность упрощения набора команд для построения быстрых вычислительных машин. Эту традицию упрощения архитектуры С. Крэй с успехом применил при создании широко известной серии суперкомпьютеров компании Cray Research.*

*Однако окончательно понятие RISC в современном его понимании сформировалось на базе трех исследовательских проектов компьютеров: процессора 801 компании IBM, процессора RISC университета Беркли и процессора MIPS Стенфордского университета.*

*Разработка экспериментального проекта компании IBM началась еще в конце 70-х годов, но его результаты никогда не публиковались и компьютер на его основе в промышленных масштабах не изготавливался. В 1980 году Д.Паттерсон со своими коллегами из Беркли начали свой проект и изготовили две машины, которые получили названия RISC-I и RISC-II. Главными идеями этих машин было отделение медленной памяти от высокоскоростных регистров и использование регистровых окон. В 1981 году Дж.Хеннесси со своими коллегами опубликовал описание стенфордской машины MIPS, основным аспектом разработки которой была эффективная реализация конвейерной обработки посредством тщательного планирования компилятором его загрузки.*

*Ко времени завершения университетских проектов (1983-1984 гг.) обозначился также прорыв в технологии изготовления сверхбольших интегральных схем. Простота архитектуры и ее эффективность, подтвержденная этими проектами, вызвали большой интерес в компьютерной индустрии и с 1986 года началась активная промышленная реализация архитектуры RISC. К настоящему времени эта архитектура прочно занимает лидирующие позиции на мировом компьютерном рынке рабочих станций и серверов.*

RISC архитектура отделяет команды обработки от команд работы с памятью и делает упор на эффективную конвейерную обработку. Система команд разрабатывалась таким образом, чтобы выполнение любой команды занимало небольшое количество машинных тактов (предпочтительно один машинный такт). Сама логика выполнения команд с целью повышения производительности ориентировалась на аппаратную, а не на микропрограммную реализацию. Другая особенность RISC-архитектуры - наличие достаточно большого регистрового файла (в типовых RISC-процессорах реализуются 32 или большее число регистров по сравнению с 8 - 16 регистрами в CISC-архитектурах), что позволяет большему объему данных храниться в регистрах на процессорном кристалле большее время и упрощает работу компилятора по распределению регистров под переменные. Составление программ для таких процессоров существенно упрощается, а производительность процессора возрастает. RISC архитектура эффективна при выполнении научных и инженерных расчетов, где над каждой единицей данных производится большой объем вычислений.

Следует отметить, что в последних разработках компании Intel (имеется в виду Pentium P54C и процессор следующего поколения P6), а также ее последователей-конкурентов (AMD R5, Cyrix M1, NexGen Nx586 и др.) широко используются идеи, реализованные в RISC-микропроцессорах, так что многие различия между CISC и RISC стираются. Однако сложность архитектуры и системы команд x86 остается и является главным фактором, ограничивающим производительность процессоров на ее основе.

### **Параллелизм процесса вычислений.**

Принцип фон Неймана основан на строго последовательной обработке команд и данных. Однако, вскоре выяснилось, что компьютерные вычисления

обладают естественным параллелизмом, т.е. часть команд программы может выполняться независимо друг от друга одновременно. Вся дальнейшая история вычислительной техники развивалась в соответствии с логикой параллелизма программ и процессоров. Любая вычислительная система (будь то супер-ЭВМ или персональный компьютер) достигает своей наивысшей производительности благодаря использованию высокоскоростных элементов и параллельному выполнению большого числа операций. Именно возможность параллельной работы различных устройств системы (работы с перекрытием) является основой ускорения основных операций.

Американский ученый М.Флинн (M. Flynn) обратил внимание на то, что существует всего две причины, порождающие вычислительный параллелизм:

1. Независимость потоков команд , одновременно существующих в системе;

2. Независимость и несвязанность данных, обрабатываемых в одном потоке команд.

Флинн выделяет 6 уровней программного параллелизма:

1- ВЕРХНИЙ УРОВЕНЬ ПАРАЛЛЕЛИЗМА	
1 – независимые задания	Достигается за счет независимых командных потоков (параллельные потоки команд)
2 – шаги задания и программы	
3 – программы и подпрограммы	
2- НИЖНИЙ УРОВЕНЬ ПАРАЛЛЕЛИЗМА	
4 – циклы и итерации	Достигается за счет несвязанных потоков данных (параллельные потоки данных)
5 – операторы и команды	
6 – фазы команд	

Параллелизм верхнего уровня достигается за счет многопроцессорной обработки.

Параллелизм нижнего уровня достигается за счет конвейерной обработки - каждый блок процессора выполняет только одну операцию, но очень быстро. Процесс выполнения команды разбивается на последовательность типовых этапов ( *выборка команды, дешифрация операции, чтение операндов, исполнение , запись результата*). Большая часть вычислительного процесса проходит в режиме совмещения выполнения команд в конвейере.

Каждая архитектура компьютера ориентирована на определенный вид параллелизма, поэтому для обеспечения наибольшей отдачи от работы вычислительной системы пользователь должен знать какой тип параллелизма доминирует в его задачах.

Параллельные ЭВМ по классификации Флинна подразделяются на машины типа **SIMD** (Single Instruction Multiple Data - с одним потоком команд при множественном потоке данных) и **MIMD** (Multiple Instruction Multiple Data - с множественным потоком команд при множественном потоке данных).

Можно выделить четыре основных типа архитектуры систем параллельной обработки:

1) **Конвейерная и векторная обработка данных.**

Основу конвейерной обработки составляет раздельное выполнение некоторой операции в несколько этапов (за несколько ступеней) с передачей



данных одного этапа следующему. Производительность при этом возрастает благодаря тому, что одновременно на различных ступенях конвейера выполняются несколько операций. Конвейеризация эффективна только тогда, когда загрузка конвейера близка к полной, а скорость подачи новых операндов соответствует максимальной производительности конвейера. Если происходит задержка, то параллельно будет выполняться меньше операций и суммарная производительность снизится.

Другой вид параллелизма – векторные операции. Рассмотрим операцию сложения двух одномерных матриц (массивов):

$$\begin{array}{cccc} a_1 & a_2 & \dots & a_n \\ b_1 & b_2 & \dots & b_n \end{array}$$

Компьютер фон Неймана должен выполнить  $n$  последовательных операций сложения. Идея векторной обработки заключается в том, что в систему команд компьютера вводится векторная операция сложения, которая выполняет параллельно сложение всех  $n$  пар элементов векторов  $\mathbf{a}$  и  $\mathbf{b}$ .

Векторные операции обеспечивают идеальную возможность полной загрузки вычислительного конвейера. При выполнении векторной команды одна и та же операция применяется ко всем элементам вектора (или чаще всего к соответствующим элементам пары векторов). Для настройки конвейера на выполнение конкретной операции может потребоваться некоторое установочное время, однако затем операнды могут поступать в конвейер с максимальной скоростью, допускаемой возможностями памяти. При этом не возникает пауз ни в связи с выборкой новой команды, ни в связи с определением ветви вычислений при условном переходе. Таким образом, главный принцип вычислений на векторной машине состоит в выполнении некоторой элементарной операции или комбинации из нескольких элементарных операций, которые должны повторно применяться к некоторому блоку данных. Таким операциям в исходной программе соответствуют небольшие компактные циклы.

## 2) Машины типа SIMD.

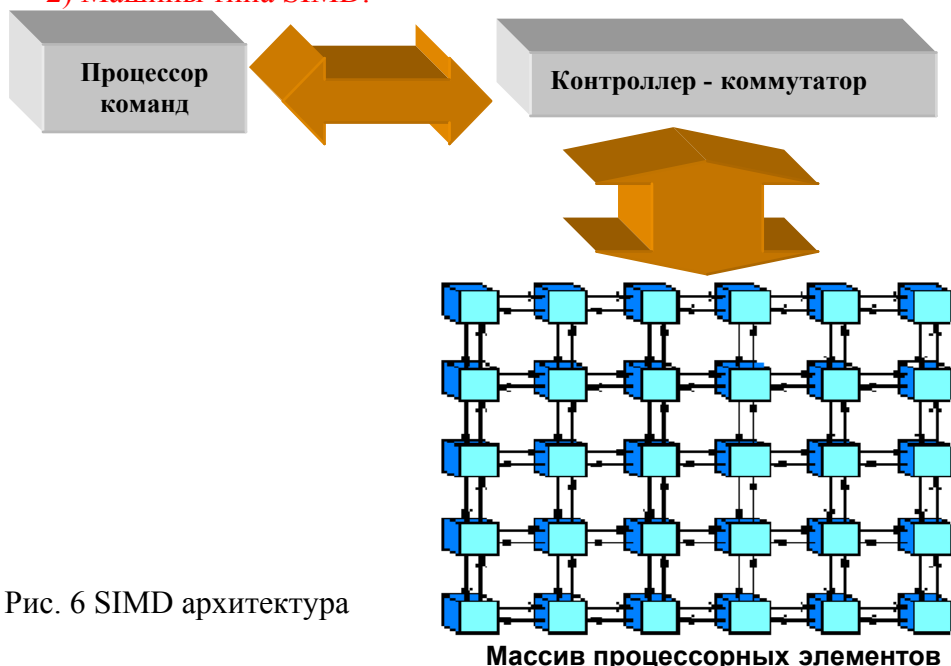


Рис. 6 SIMD архитектура



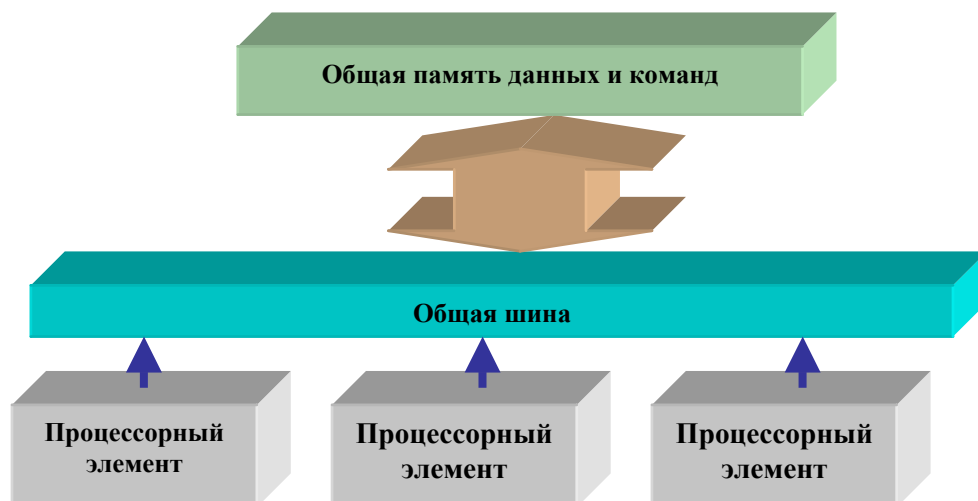
Машины типа SIMD состоят из большого числа идентичных процессорных элементов, имеющих собственную память. Все процессорные элементы в такой машине выполняют одну и ту же программу. Очевидно, что такая машина, составленная из большого числа процессоров, может обеспечить очень высокую производительность только на тех задачах, при решении которых все процессоры могут делать одну и ту же работу. Модель вычислений для машины SIMD очень похожа на модель вычислений для векторного процессора: одиночная операция выполняется над большим блоком данных.

В отличие от ограниченного конвейерного функционирования векторного процессора, матричный процессор (синоним для большинства SIMD-машин) может быть значительно более гибким. Обрабатывающие элементы таких процессоров - это универсальные программируемые ЭВМ, так что задача, решаемая параллельно, может быть достаточно сложной и содержать ветвления. Обычное проявление этой вычислительной модели в исходной программе примерно такое же, как и в случае векторных операций: циклы на элементах массива, в которых значения, вырабатываемые на одной итерации цикла, не используются на другой итерации цикла.

Модели вычислений на векторных и матричных ЭВМ настолько схожи, что эти ЭВМ часто обсуждаются как эквивалентные.

### 3) Машины типа MIMD.

Термин "мультипроцессор" покрывает большинство машин типа MIMD и (подобно тому, как термин "матричный процессор" применяется к машинам типа SIMD) часто используется в качестве синонима для машин типа MIMD. В мультипроцессорной системе каждый процессорный элемент (ПЭ) выполняет свою программу достаточно независимо от других процессорных элементов. Процессорные элементы, конечно, должны как-то связываться друг с другом, что делает необходимым более подробную классификацию машин типа MIMD. В мультипроцессорах с общей памятью (сильносвязанных мультипроцессорах) имеется память данных и команд, доступная всем ПЭ. С общей памятью ПЭ связываются с помощью общей шины или сети обмена. В противоположность этому варианту в слабосвязанных многопроцессорных системах (машинах с локальной памятью) вся память делится между процессорными элементами и каждый блок памяти доступен только связанному с ним процессору. Сеть обмена связывает процессорные элементы друг с другом.



Базовой моделью вычислений на MIMD-мультипроцессоре является совокупность независимых процессов, эпизодически обращающихся к разделяемым данным. Существует большое количество вариантов этой модели. На одном конце спектра - модель распределенных вычислений, в которой программа делится на довольно большое число параллельных задач, состоящих из множества подпрограмм. На другом конце спектра - модель потоковых вычислений, в которых каждая операция в программе может рассматриваться как отдельный процесс. Такая операция ждет своих входных данных (операндов), которые должны быть переданы ей другими процессами. По их получении операция выполняется, и полученное значение передается тем процессам, которые в нем нуждаются. В потоковых моделях вычислений с большим и средним уровнем гранулярности, процессы содержат большое число операций и выполняются в потоковой манере.

#### 4) Многопроцессорные машины с SIMD-процессорами.

Многие современные супер-ЭВМ представляют собой многопроцессорные системы, в которых в качестве процессоров используются векторные процессоры или процессоры типа SIMD. Такие машины относятся к машинам класса MSIMD.