

ОТМП

Тема №1 Введение

Введение.

Вычислительные машины по принципу действия делятся на аналоговые (АВМ) и цифровые (ЦВМ).

- **Аналоговые вычислительные машины** отличаются тем, что вводимые в них математические величины выражаются в виде физических величин (в основном электрических напряжений или токов).
- Точность работы АВМ зависит от точности исполнения функциональных преобразований блоков, входящих в их состав.
- Блоки предназначены для выполнения какой-либо определенной математической операции (сложения, вычитания, умножения, деления, интегрирования и т. д.) и соединяются в определенной последовательности для обеспечения решения конкретной задачи.
- Наиболее часто АВМ используют для решения дифференциальных уравнений, моделирования различных физических процессов
- Создание электронных АВМ связано с развитием радиоэлектроники, изобретением и усовершенствованием операционного усилителя постоянного тока. Серийное производство АВМ на электронных лампах в СССР началось в 1946 г. Затем был освоен выпуск АВМ на полупроводниковых приборах. Выпуск АВМ на базе интегральных микросхем был начат в 70-е годы.
- В настоящее время практически не используются.

Введение.

Вычислительные машины по принципу действия делятся на аналоговые (АВМ) и цифровые (ЦВМ).

- **Цифровые вычислительные машины.**
- В ЦВМ математические и логические операции производятся над числами, представленными в виде цифровых кодов, для чего в них имеются электронные устройства, с помощью которых и осуществляются эти операции.
- Последовательность математических вычислений и логических преобразований в ЦВМ осуществляется в соответствии с программой.
- Основные преимущества ЦВМ перед АВМ — универсальность и большая точность вычисления, зависящая от количества разрядов, используемых в машине для представления чисел.
- На сегодняшний день АВМ потеряли актуальность и выведены из эксплуатации. В связи с этим, этот тип ЭВМ рассматривать не будем и в дальнейшем под ЭВМ будем понимать ЦВМ.

Введение.

- В зависимости от применяемой элементной базы различают ЭВМ нескольких поколений.
- **ЭВМ первого поколения** (1945-1954 гг.) - электронные лампы. Быстродействие до десятков тысяч арифметических операций в секунду и емкость памяти до нескольких тысяч слов.
- **ЭВМ второго поколения** (1955-1965 - полупроводниковые диоды и транзисторы. Быстродействие до миллиона операций в секунду, емкость памяти до нескольких десятков тысяч слов.
- **ЭВМ третьего поколения** (1965-1976 гг.) - интегральные схемы. Переход на интегральные схемы способствовал повышению надежности ЭВМ, уменьшению их габаритов и потребляемой мощности.
- Начиная с **середины 1970-х** годов стройная картина смены поколений нарушается.
- **Период с 1975 г. это период четвертого поколения.** Элементная база - большие интегральные схемы (БИС).
- Другое мнение - достижения периода 1975-1985 г.г. не настолько велики, чтобы считать его равноправным поколением. «Третье с половиной» поколение.
- **Только с 1985г.**, когда появились сверхбольшие интегральные схемы (СБИС), собственно четвертое поколение.
- **Пятое поколение** - это ЭВМ, ориентированные на решение задач искусственного интеллекта.

Характеристики ЭВМ (один из возможных вариантов)

- 1. Операционные ресурсы ЭВМ** – это перечень возможностей ЭВМ:
 - способы представления информации в ЭВМ
 - система команд ЭВМ
 - способы адресации
- Операционные ресурсы ЭВМ напрямую связаны с аппаратными средствами, которые характеризуют пригодность ЭВМ для решения тех или иных задач.
- 2. Объем памяти**
 - оперативной ОЗУ – внутренняя
 - внешняя, например HDD
 - объем сверхоперативной кэш- памяти.
- 3. Быстродействие ЭВМ** характеризует скорость обработки информации компьютером.
- 4. Надежность ЭВМ** – свойство ЭВМ выполнять возложенные на нее функции в течение заданного промежутка времени. Типы отказов:
 - внезапный отказ
 - деградация параметров ЭВМ
- 5. Показатель стоимости:**
 - суммарная стоимость оборудования
 - стоимость единицы вычислений

По назначению

По назначению ЭВМ можно разделить на:

- 1) ЭВМ общего назначения (универсальные). Имеют архитектуру, позволяющую решать широкий круг различных задач.
- 2) Проблемно-ориентированные – для решения узкого круга задач.
- 3) Специализированные, например, управляющие (*Контроллеры*).

Некоторые определения

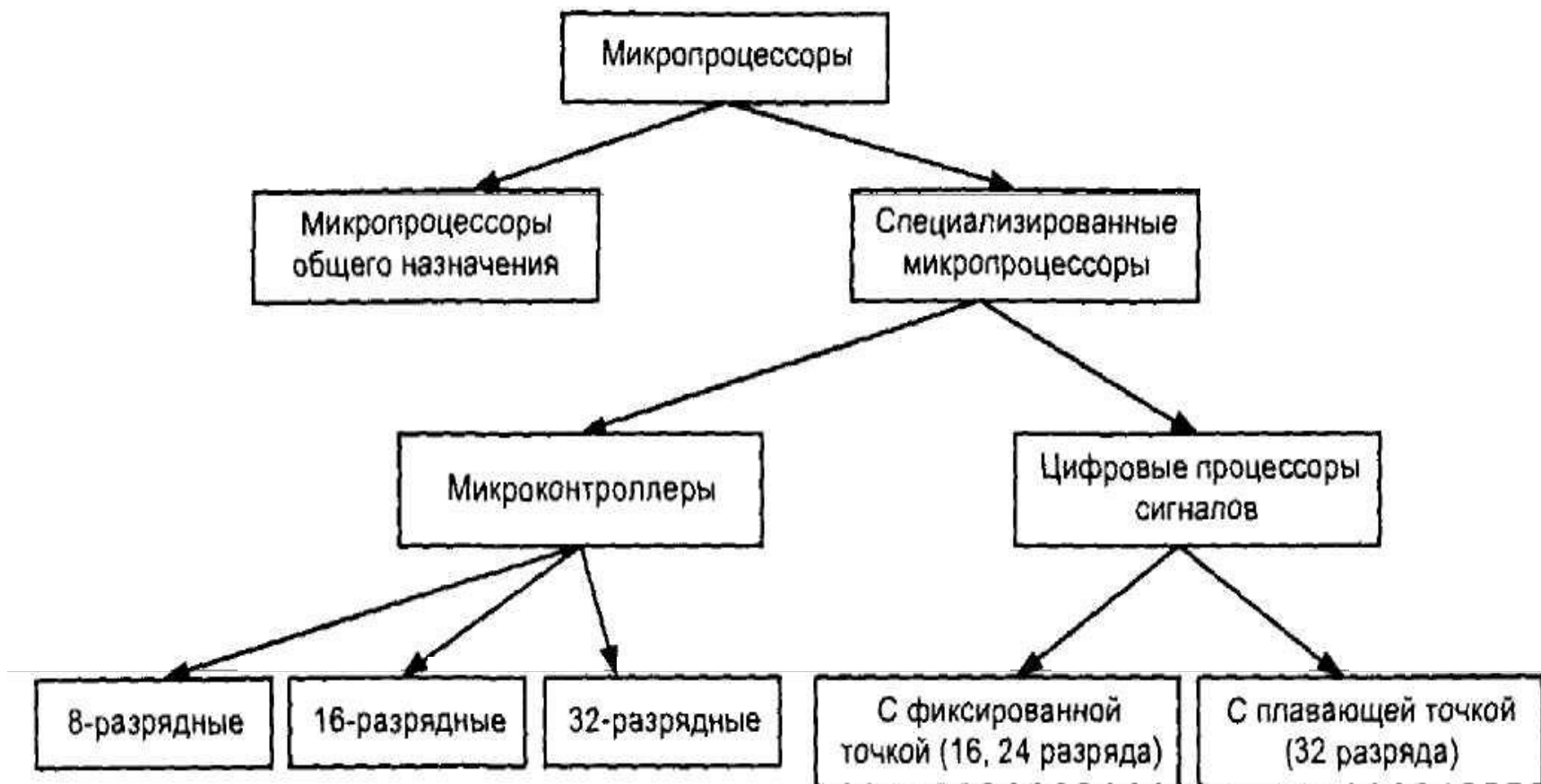
- *Микропроцессорная техника* включает технические и программные средства, используемые для построения различных микропроцессорных систем, устройств и персональных микро-ЭВМ.
- *Микропроцессорная система* представляет собой функционально законченное изделие, состоящее из одного или нескольких устройств, главным образом микропроцессорных: микропроцессора и/или микроконтроллера.
- *Микропроцессорное устройство* представляет собой функционально и конструктивно законченное изделие, состоящее из нескольких микросхем, в состав которых входит микропроцессор; оно предназначено для выполнения определенного набора функций: получение, обработка, передача, преобразование информации и управление.
- Под *микропроцессором* будем понимать программно-управляемое устройство, осуществляющее процесс цифровой обработки информации и управления и построенное, как правило, на одной БИС.

Микроконтроллер

- Термин «микроконтроллер» (МК) вытеснил из употребления термин «однокристальная микро-ЭВМ».
- На одном кристалле размещены не только микропроцессор, но память и устройства ввода-вывода.
- С появлением однокристальных микро-ЭВМ связывают начало эры компьютерной автоматизации в области управления. Это обстоятельство и определило термин «*микроконтроллер*» (control - управление).
- Расширение сферы использования МК повлекло за собой развитие архитектуры за счет размещения на кристалле устройств (модулей), отражающих своими функциональными возможностями специфику решаемых задач. Такие дополнительные устройства стали называться *периферийными*.
- По этой же причине появились не только семейства МК, которые объединяют родственные МК с одинаковой системой команд, разрядностью, но и стали выделяться подвиды МК: ***коммуникационные, для управления*** и т. д.

Микропроцессоры (вариант)

- **МП** в настоящее время преимущественно используются для производства персональных ЭВМ
- **МК** являются основой создания различных встраиваемых систем, телекоммуникационного, портативного оборудования и т.д.



Проектирование (этапы)

При разработке системы любого назначения на базе микроконтроллеров, в общем случае, необходимо выполнить следующие этапы:

1. Системный анализ задачи - выделяются процессы и функции, реализация которых будет возложена на МК.
2. Алгоритмизация процессов и функций - разрабатываются алгоритмы решения задачи.
3. Выбор МК и комплексная разработка программно аппаратных средств. Осуществляется выбор технических средств соответствующей компании, инструментальных средств поддержки процесса проектирования (отладочных средств, языков программирования и т. д.), а также операционной системы реального времени, если это требуется для решения задачи.
4. Производится программирование алгоритмов, полученных на втором этапе, изготовление системы на базе выбранного МК и комплексная отладка.
5. На каждом этапе следует анализировать стоимость того или иного решения и добиваться оптимума по критерию функциональность-стоимость.

Принципы выбора МК

Области использования МК

Значение критерия	Характеристика задач	Разрядность МК/ производительность
Мало данных – мало вычислений	Задачи логического управления несложными объектами и процессами	8/Низкая
Мало данных – много вычислений	Локальные регуляторы, системы управления электрическими двигателями, подвижными аппаратами, различными электрическими агрегатами, роботами-манипуляторами, станками, портативное оборудование и т. д.	16/Средняя
Много данных – мало вычислений	Многие сетевые задачи, системы управления потоками данных, коммутаторы, концентраторы, маршрутизаторы и т.п.	32/Высокая
Много данных – много вычислений	Задачи управления реального времени, обработка сигналов с интенсивным обменом, системы распознавания речи, изображений и т. п.	32/Сверхвысокая

Архитектура процессора

- **Архитектурой процессора** называется комплекс его аппаратных и программных средств, предоставляемых пользователю - набор программно-доступных регистров и исполнительных (операционных) устройств, система основных команд и способов адресации, объем и структура адресуемой памяти, виды и способы обработки прерываний.
- При описании архитектуры процессора обычно используется его представление в виде совокупности программно-доступных регистров, образующих *регистровую* или *программную модель*.
- В этих регистрах содержатся обрабатываемые данные (операнды) и управляющая информация.
- Соответственно, в регистровую модель входит группа *регистров общего назначения*, служащих для хранения операндов, и группа *служебных регистров*, обеспечивающих управление выполнением программы и режимом работы процессора, организацию обращения к памяти (защита памяти, сегментная и страничная организация и др.).
- Регистры общего назначения образуют - *внутреннюю регистровую память* процессора. Состав и количество служебных регистров определяется архитектурой процессора.

Служебные регистры

Обычно в их состав входят:

- -программный счетчик PC;
- -регистр состояния SR (или EFLAGS);
- -регистры управления режимом работы процессора CR (Control Register);
- -регистры, реализующие сегментную и страничную организацию памяти; регистры, обеспечивающие отладку программ и тестирование процессора.
- Кроме того, различные модели микропроцессоров содержат ряд других специализированных регистров.

Архитектура МП (CISC)

- **CISC (Complex Instruction Set Computer)** – архитектура реализована во многих типах микропроцессоров, выполняющих большой набор разноформатных команд с использованием многочисленных способов адресации.
- Эта классическая архитектура процессоров, которая начала свое развитие в 1940-х годах с появлением первых компьютеров. Типичным примером CISC-процессоров являются первые микропроцессоры i8080, i-x86.
- Большое многообразие выполняемых команд и способов адресации позволяет программисту реализовать наиболее эффективные алгоритмы решения различных задач.
- Существенно усложняется структура микропроцессора, особенно его устройства управления, что приводит к увеличению размеров и стоимости кристалла, снижению производительности.
- Многие команды и способы адресации используются достаточно редко. Поэтому, начиная с 1980-х годов, интенсивное развитие получила архитектура процессоров с сокращенным набором команд (RISC-процессоры).

Архитектура МП (RISC)

- **RISC (Reduced Instruction Set Computer)** - архитектура отличается использованием ограниченного набора команд фиксированного формата.
- Значительно сокращается число используемых способов адресации.
- Для сокращения количества обращений к памяти RISC-процессоры имеют увеличенное количество внутренних регистров.
- Существенно упрощается структура микропроцессора, сокращаются его размеры и стоимость, значительно повышается производительность.
- Указанные достоинства RISC-архитектуры привели к тому, что во многих современных CISC-процессорах используется RISC-ядро, выполняющее обработку данных. Поступающие сложные и разноформатные команды предварительно преобразуются в последовательность простых RISC-операций, быстро выполняемых этим процессорным ядром. Так работают многие модели микропроцессоров, которые по внешним показателям относятся к CISC-процессорам. Использование RISC-архитектуры является характерной чертой многих современных микропроцессоров.

Архитектура МП (VLIW)

- **Very Large Instruction Word** – архитектура появилась относительно недавно - в 1990-х годах.
- Особенностью является использование очень длинных команд (до 128 бит и более), отдельные поля которых содержат коды, обеспечивающие выполнение различных операций.
- Одна команда вызывает выполнение сразу нескольких операций параллельно в различных операционных устройствах, входящих в структуру микропроцессора.
- При трансляции программ, написанных на языке высокого уровня, соответствующий компилятор производит формирование «длинных» VLIW-команд, каждая из которых обеспечивает реализацию процессором целой процедуры или группы операций.
- Данная архитектура реализована в некоторых типах современных микропроцессоров (PA8500 компании «Hewlett-Packard», некоторые типы DSP-цифровых процессоров сигналов) и является весьма перспективной для создания нового поколения сверхвысокопроизводительных процессоров.

Фон-Нейман, Гарвард

- Кроме набора выполняемых команд и способов адресации важной архитектурной особенностью микропроцессоров является используемый ***вариант реализации памяти и организация выборки команд и данных.***
- По этим признакам различаются процессоры с
- **Принстонской (иначе – архитектура фон Неймана) и**
- **Гарвардской архитектурой.**
- Эти архитектурные варианты были предложены в конце 1940-х годов специалистами соответственно Принстонского и Гарвардского университетов США для разрабатываемых ими моделей компьютеров.

Архитектура Фон-Неймана (Принстонская архитектура)

- **Принстонская архитектура** (Фон-Неймана), характеризуется использованием общей оперативной памяти для хранения программ, данных, а также для организации стека. Для обращения к этой памяти используется **общая системная шина**, по которой в процессор поступают и команды, и данные.
- Достоинства: а) наличие общей памяти позволяет оперативно перераспределять ее объем для хранения отдельных массивов команд, данных и реализации стека. Значительно упрощается отладка, тестирование и текущий контроль функционирования системы. Архитектура Фон-Неймана в течение долгого времени доминировала в вычислительной технике.
- Недостатки: а) основным из них является необходимость **последовательной выборки команд и обрабатываемых данных** по общей системной шине. Общая шина становится «узким местом» (bottleneck - «бутылочное горло»), которое **ограничивает производительность** системы.
- Постоянно возрастающие требования к производительности МП систем и возможность размещения ПЗУ программ на одном кристалле с ядром процессора вызвали все более широкое применение Гарвардской архитектуры при создании многих типов современных микропроцессоров.

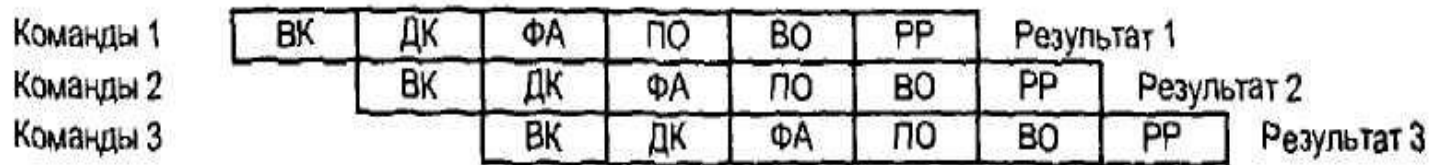
Гарвардская архитектура

- **Гарвардская архитектура** характеризуется физическим разделением памяти команд (программ) и памяти данных. В оригинальном варианте использовался также отдельный стек. Каждая память соединяется с процессором ***отдельной шиной***, что позволяет одновременно с чтением-записью данных при выполнении текущей команды производить выборку и декодирование следующей команды. Благодаря такому разделению потоков команд и данных и совмещению операций их выборки реализуется более высокая производительность, чем при использовании Принстонской архитектуры.
- Недостатки связаны с необходимостью проведения большего числа шин, а также с фиксированным объемом памяти, выделенной для команд и данных, назначение которой не может оперативно перераспределяться в соответствии с требованиями решаемой задачи.
- Развитие микроэлектронной технологии позволило в значительной степени преодолеть указанные недостатки. Гарвардская архитектура широко применяется во внутренней структуре современных микропроцессоров, где используется отдельная кэш-память для хранения команд и данных. В то же время во внешней структуре большинства микропроцессорных систем реализуются принципы Принстонской архитектуры.
- Широко используется в микроконтроллерах – специализированных микропроцессорах для управления различными объектами, рабочая программа которых обычно хранится в отдельном ПЗУ.

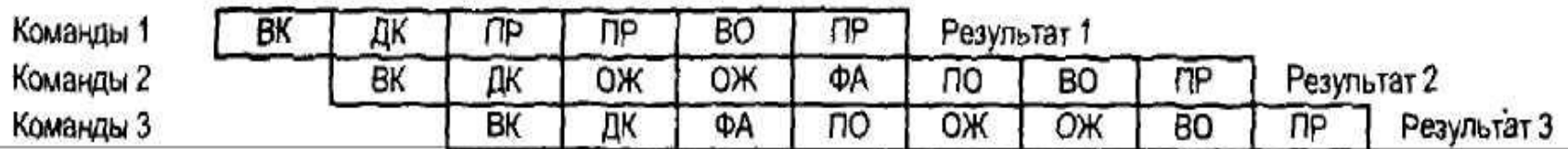
Конвейер

Во внутренней структуре современных высокопроизводительных микропроцессоров реализуется *конвейерный принцип выполнения команд*. При этом процесс выполнения команды разбивается на ряд этапов. Приведен пример разбиения команды на шесть этапов ее выполнения:

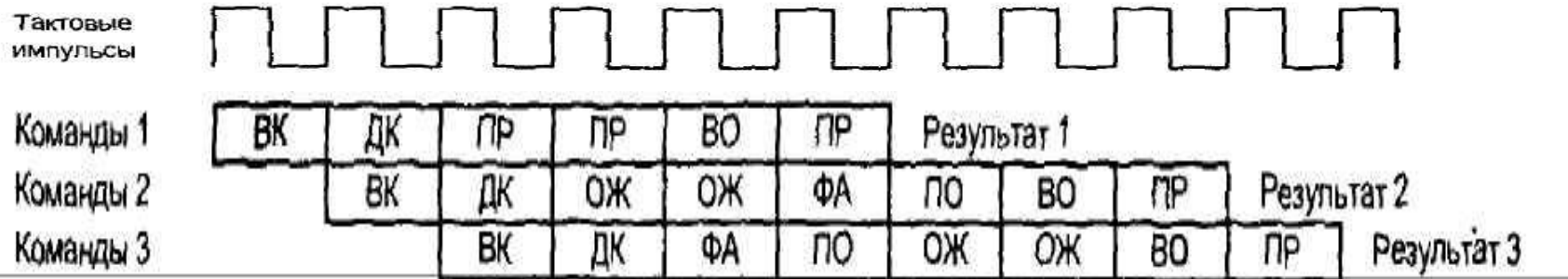
1. выборка очередной команды (ВК);
2. декодирование выбранной команды (ДК);
3. формирование адреса операнда (ФА);
4. прием операнда из памяти (ПО);
5. выполнение операции (ВО);
6. размещение результата в памяти (РР).



б)



Ожидание, простой



- Эффективная работа конвейера обеспечивается только при его равномерной загрузке однотипными командами. Реально отдельные ступени конвейера могут оказаться незагруженными, находясь в состоянии ожидания или простоя.
- *Ожиданием* называется состояние исполнительной ступени, когда она не может выполнить требуемую микрооперацию, так как еще не получен необходимый операнд, являющийся результатом выполнения предыдущей команды.
- *Простоем* называется состояние ступени, когда она вынуждена пропустить очередной такт, так как поступившая команда не требует выполнения соответствующего этапа. Например, при выполнении безадресных команд не требуется производить формирование адреса и прием операнда (простой на ступенях ФА и ПО конвейера).

Эффективность использования конвейера

- *Эффективность использования конвейера* определяется типом поступающих команд
- При поступлении однородных команд обеспечивается сокращение числа состояний простоя и ожидания в процессе их выполнения, в результате чего повышается производительность процессора.
- При использовании в программе разноформатных команд, содержащих различное количество байтов, число состояний простоя и ожидания, которые приходится вводить в процессе выполнения команд, значительно увеличивается.
- Принятый во многих RISC-процессорах стандартный по длине формат команд обеспечивает существенное сокращение числа ожиданий и простоев конвейера, что позволяет значительно повысить производительность.
- Другой причиной снижения эффективности конвейера являются *команды условного ветвления*.

Предсказания ветвлений

- Если выполняется условие ветвления, то приходится производить перезагрузку конвейера командами из другой ветви программы, что вызывает значительное снижение производительности.
- Одним из основных условий эффективной работы конвейера является сокращение числа его перезагрузок при выполнении условных переходов.
- Эта цель достигается с помощью реализации различных механизмов предсказания направления ветвления, которые обеспечиваются с помощью специальных устройств - *блоков предсказания ветвления*, вводимых в структуру процессора.
- Наиболее простой способ состоит в том, что процессор фиксирует результат выполнения предыдущих команд ветвления по данному адресу и считает, что следующая команда с обращением по этому адресу даст аналогичный результат.
- Для реализации этого способа предсказания ветвления используется специальная память ВТВ (Branch Target Buffer), где хранятся адреса ранее выполненных условных переходов.
- Вероятность правильного предсказания составляет 80% и более.

Суперскалярная архитектура

- Повышение производительности процессора достигается также при введении в структуру процессора **нескольких** параллельно включенных **операционных устройств**, обеспечивающих одновременное выполнение нескольких операций.
- Такая структура процессора называется **суперскалярной**.
- В идеальном случае число одновременно выполняемых команд равно числу операционных устройств.
- При выполнении реальных программ трудно обеспечить полную загрузку всех операционных устройств, поэтому на практике эффективность использования суперскалярной структуры оказывается несколько ниже.
- Современные суперскалярные процессоры содержат от 4 до 10 различных операционных устройств, параллельная работа которых обеспечивает выполнение за один такт в среднем от 2 до 6 команд.

Спекулятивное выполнение команд

- Чтобы обеспечить максимально полную загрузку операционных устройств, в процессе анализа и группировки декодированных команд возможно **изменение порядка их следования**.
- При этом обычно реализуется *спекулятивная* (с нарушением следования) выборка операндов, чтобы для поступающих на исполнение команд уже были готовы операнды, которые записываются в специальные регистры.
- В результате команды **выполняются не в порядке их выборки** из памяти, а по мере готовности необходимых операндов и исполнительных устройств. Таким образом, позже поступившие команды могут быть выполнены до ранее выбранных.
- Чтобы запись в память результатов происходила в соответствии с исходной последовательностью поступления команд программы, на выходе данных включается специальная буферная память, восстанавливающая порядок выдачи результатов согласно выполняемой программе.