

Специализированные компьютерные системы

Часть I

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.

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 микропроцессора

- **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.

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)

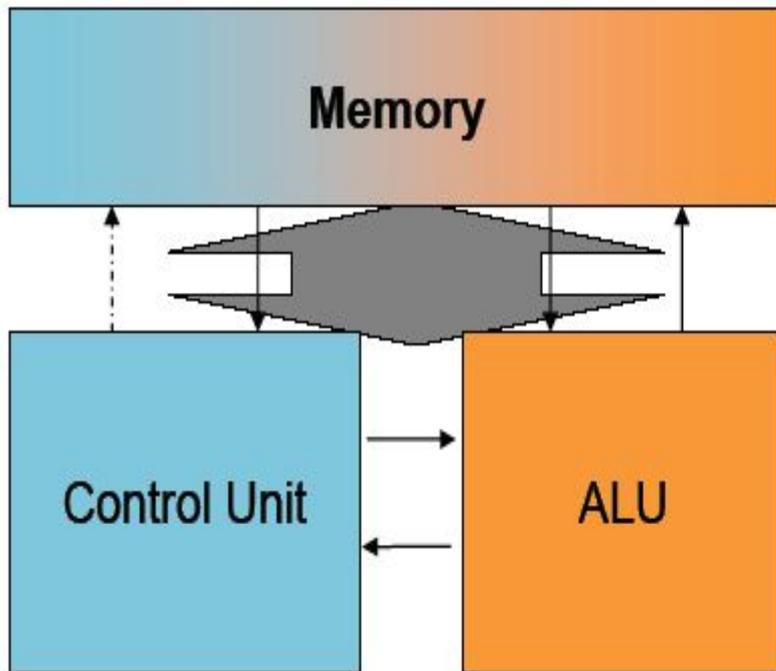
- **Very Large Instruction Word**) – архитектура появилась относительно недавно - в 1990-х годах.
- Особенностью является использование очень длинных команд (до 128 бит и более), отдельные поля которых содержат коды, обеспечивающие выполнение различных операций.
- Одна команда вызывает выполнение сразу 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

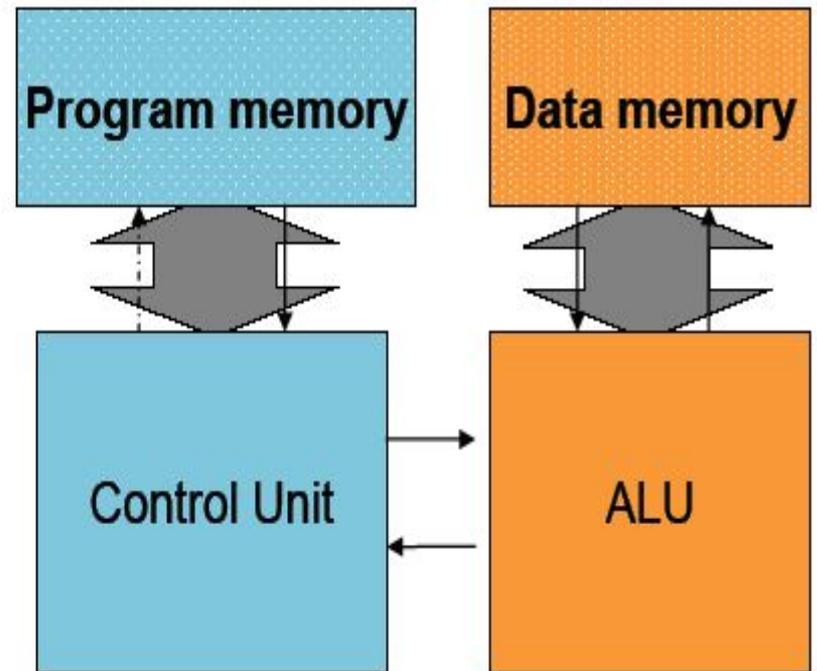
- Кроме набора выполняемых команд и способов адресации важной архитектурной особенностью микропроцессоров является используемый ***вариант реализации памяти и организация выборки команд и данных.***
- По этим признакам различаются процессоры с
- **Принстонской (иначе – архитектура фон Неймана) и**
-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

Гарвард vs Фон Нейман

Von Neumann architecture



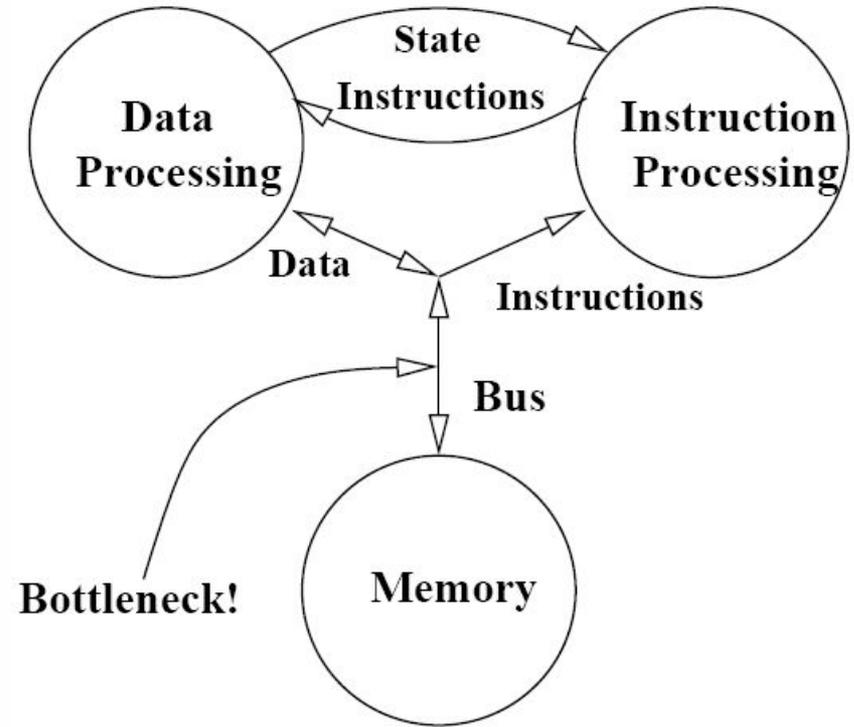
Harvard architecture



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.

«Бутылочное горлышко» архитектуры Фон-Неймана



A common bus is used for data as well as instructions.

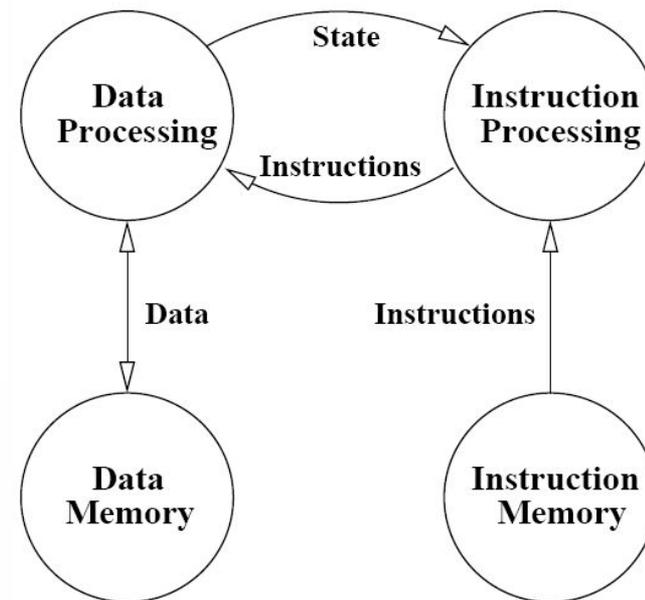
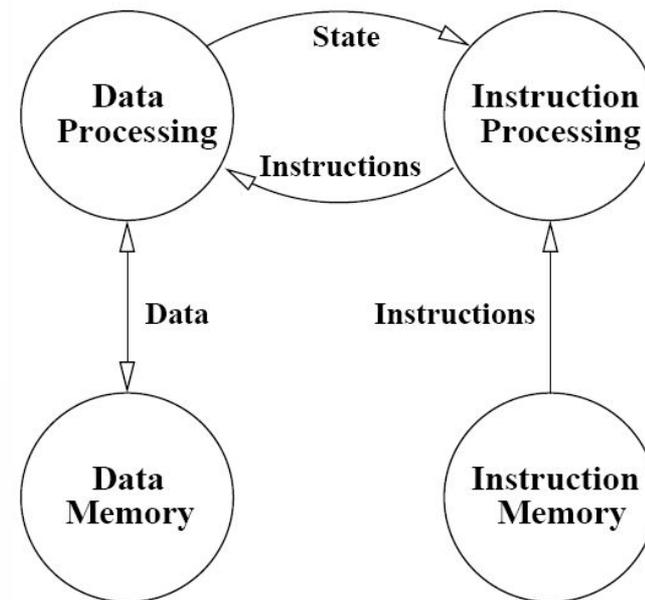
The system can become 'bus bound'.

Практически монопольное использование до конца 70-х.

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.

Гарвардская архитектура



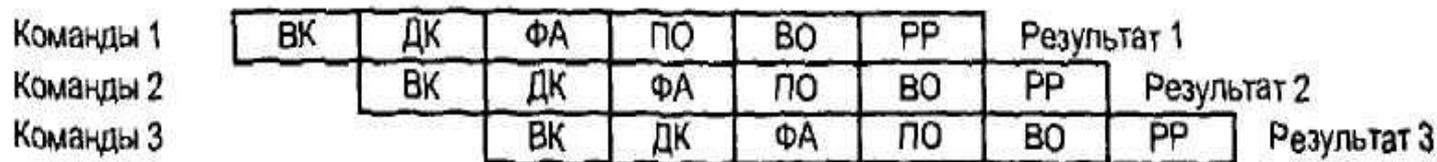
Separate data and instruction paths;
Good performance;
Needs 2 buses - expensive!
Traffic on the buses is not balanced.

Гарвардская архитектура почти не использовалась до конца 70-х.

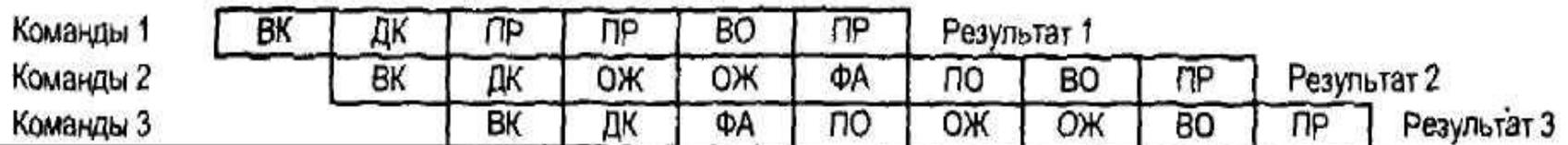
Конвейер

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:

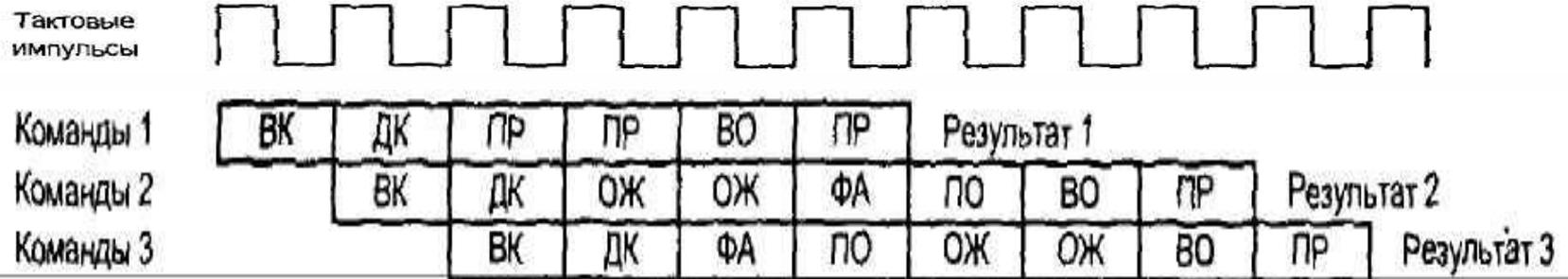
1. выборка очередной команды (ВК);
2. декодирование выбранной команды (ДК);
3. формирование адреса операнда (ФА);
4. прием операнда из памяти (ПО);
5. выполнение операции (ВО);
6. размещение результата в памяти (РР).



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).

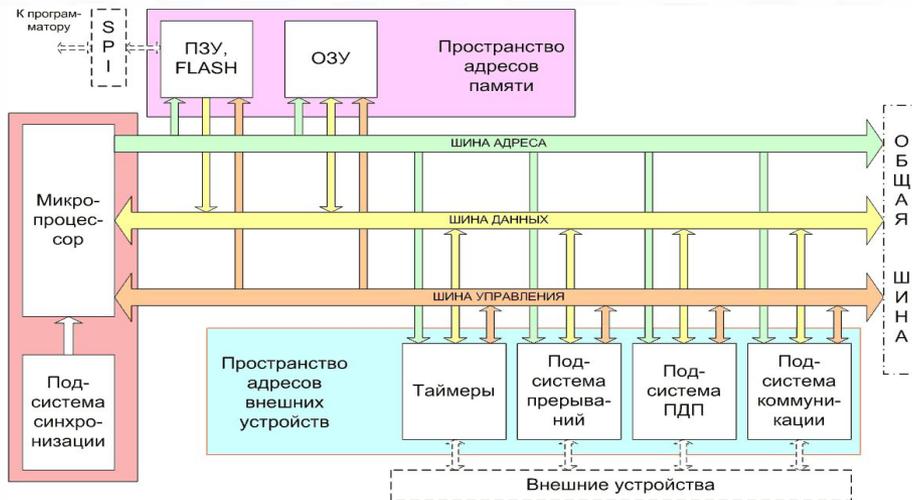
Эффективность использования конвейера

- *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.

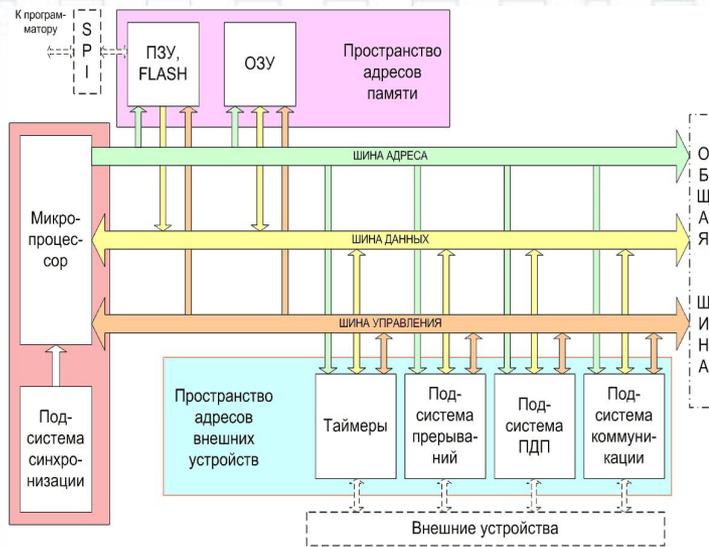
Структура микропроцессорного устройства



Наиболее широко применяется магистрально-модульный принцип построения микропроцессорных систем (МПС) при котором отдельные устройства (модули), входящие в состав системы, обмениваются информацией по общей системной шине - магистрали.

- **Системная шина** содержит большое количество проводников, которые в соответствии с их функциональным назначением подразделяются на отдельные *шины - адреса, данных и управления*.
- **Шина адреса** служит для передачи адреса, который формируется микропроцессором и позволяет выбрать необходимую ячейку памяти ОЗУ (ПЗУ) или требуемое внешнее устройство.
- **Шина данных** служит для выборки команд и данных из ПЗУ, обмена данными с ОЗУ или внешними устройствами.
- По **шине управления** передаются разнообразные управляющие сигналы, определяющие режимы работы памяти (запись или считывание), интерфейсных устройств (ввод или вывод информации) и микропроцессора (запуск, запросы внешних устройств на обслуживание, информация о текущем режиме работы и другие сигналы).

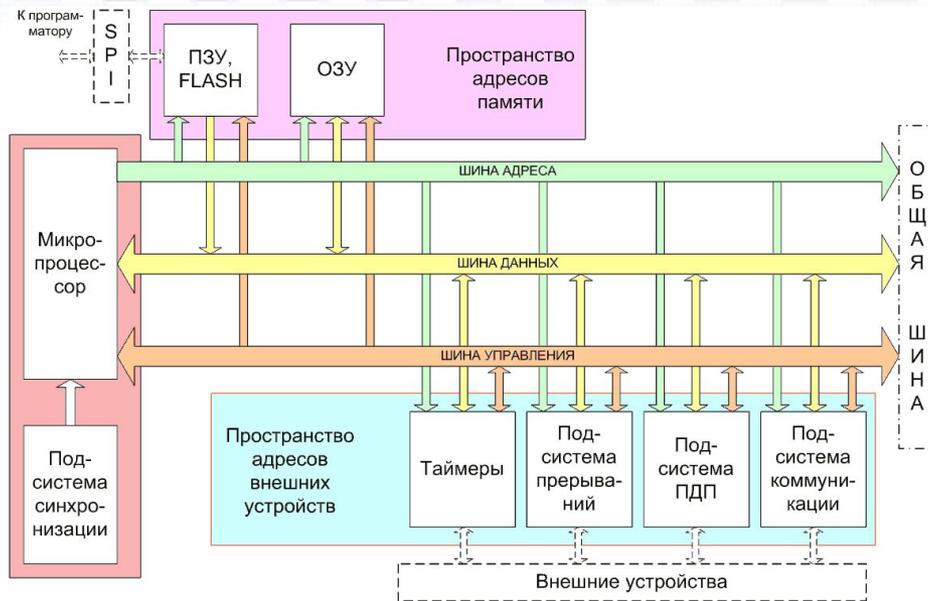
Архитектура Фон Неймана (Принстонская)



Основным преимуществом архитектуры Фон Неймана является простота. Содержимое ОЗУ (RAM - Random Access Memory) может быть использовано как для хранения данных, так и для хранения программ. В некоторых приложениях программе необходимо иметь доступ к содержимому стека. Это предоставляет большую гибкость для разработчика программного обеспечения.

- Упрощенная структура МПС *принстонской* архитектуры (фон-Неймана), использующая трехшинную магистраль. С целью сокращения количества физических соединений (проводов), возможно использование двухшинной магистрали. В случае двухшинной магистрали, шина адреса и данных объединена в единую, т.е. адреса и данные передаются по одним и тем же физическим проводникам с разделением по времени-*мультиплексирование линий адреса и данных*. Передача адреса при этом сопровождается дополнительным сигналом, указывающим на то, что в данный момент времени на шине выставлен адрес, а не данные. При этом падает скорость обмена между процессором и остальными узлами МПС.

Системная шина

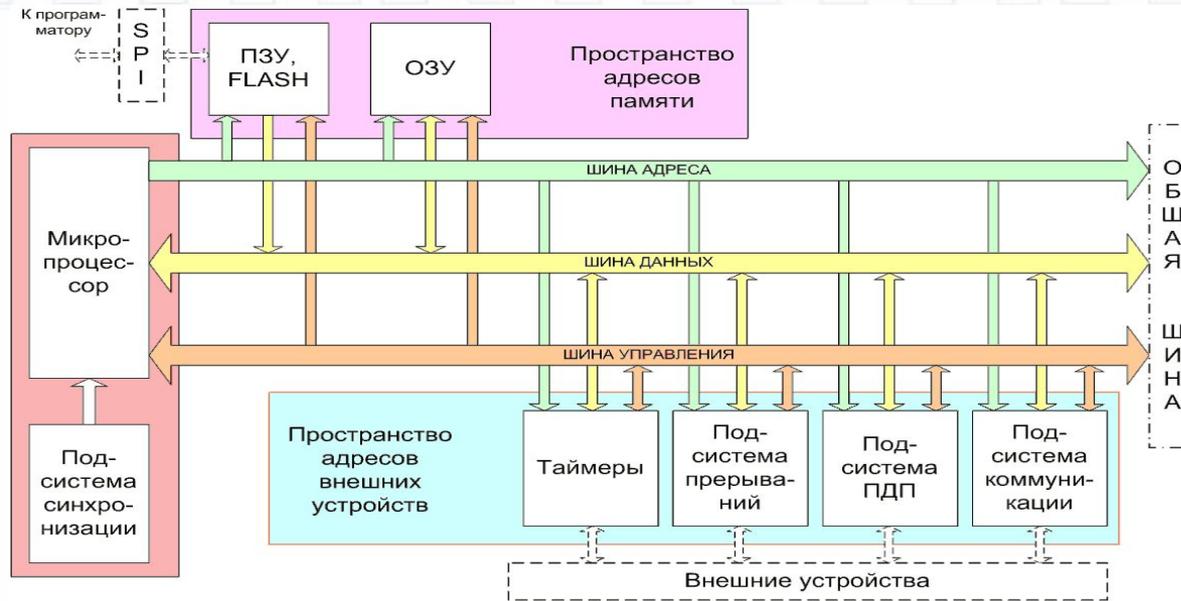


По системной шине происходит три вида передачи данных:

1. процессор - память;
2. процессор - интерфейс;
3. память - интерфейс.

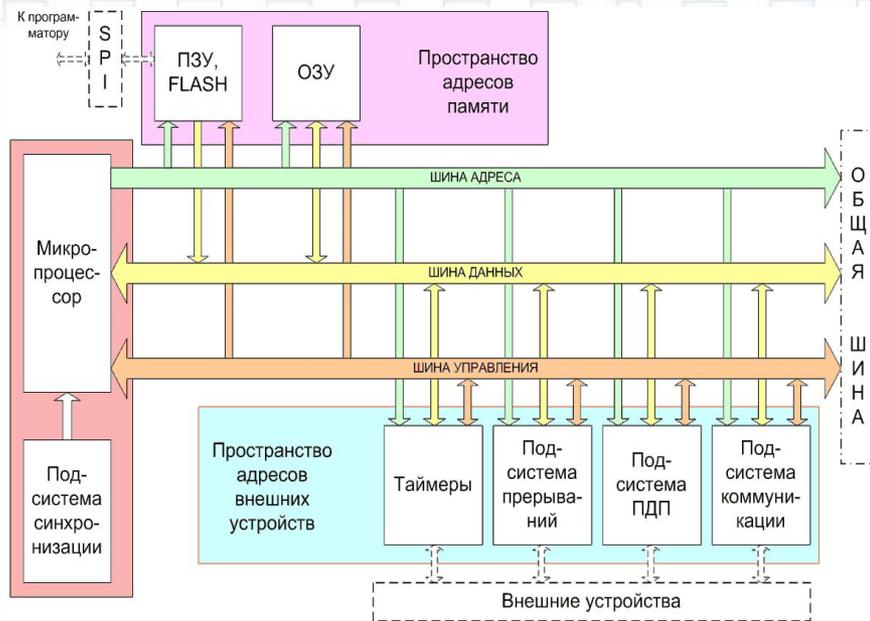
- В первых двух случаях передачей данных управляет процессор. Память или интерфейс по управляющему сигналу от процессора осуществляют передачу данных.
- Направление передачи данных, как правило, определяется процессором.
- Пересылка данных с шины данных внутрь процессора называется считыванием, обратный процесс - записью.
- Третий случай осуществляется механизмом прямого доступа к памяти ПДП и будет рассмотрен позднее.

ЦПУ, Синхронизация



- **Центральным процессорным устройством (ЦПУ)** в системе является микропроцессор (МП), выполняющий арифметические и логические операции над данными, осуществляющий управление выборкой команд и данных из памяти и организующий взаимодействие всех устройств, входящих в систему.
- Работа МП происходит под воздействием тактовых сигналов, вырабатываемых *схемой синхронизации*, часто выполняемой в виде отдельной микросхемы (генератора тактовых импульсов).

Разрядность шины. Шина данных.

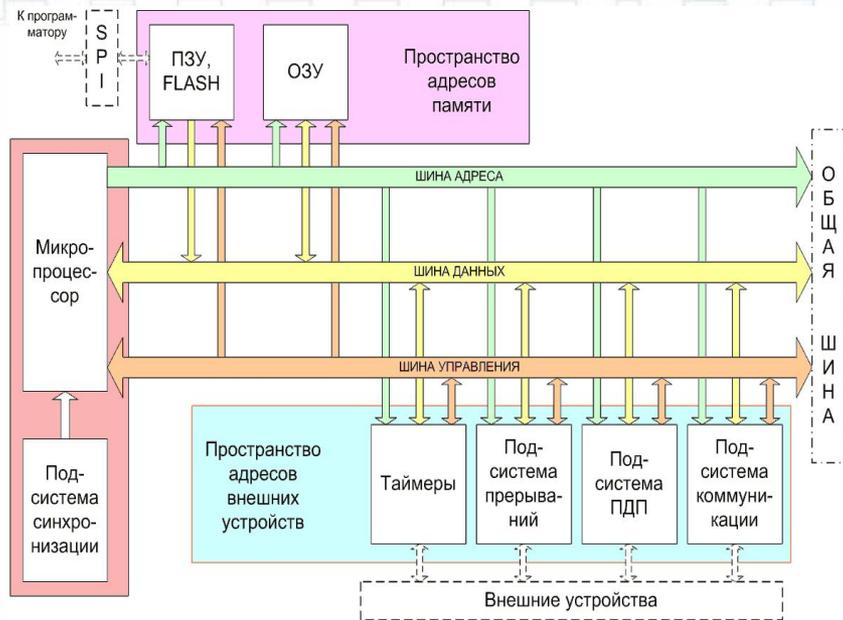


Если в качестве шины использовать восемь сигнальных проводов, можно параллельно (одновременно) передавать 8 бит информации, соответственно по 16-жильному проводу можно передавать 16 бит информации.

Количество сигнальных линий определяет разрядность шины. Обычно разрядность шины данных и длину слов, обрабатываемых процессором, выбирают одинаковыми.

- **Шина данных** используется как для передачи данных в направлении процессор - память и процессор - интерфейс, так и для их передачи в обратном направлении. Шина данных является *двунаправленной*.
- Передача по шине данных может производиться в обоих направлениях, однако в каждый заданный момент времени она осуществляется лишь в одном направлении. Это означает, что для передачи данных в систему и их приема из системы микропроцессор меняет направление работы шины данных. По всем разрядам шины в текущий момент времени данные передаются лишь в одном направлении, т. е. в любой момент по всем линиям шины они могут либо только, передаваться либо только приниматься процессором.).

Шина адреса.

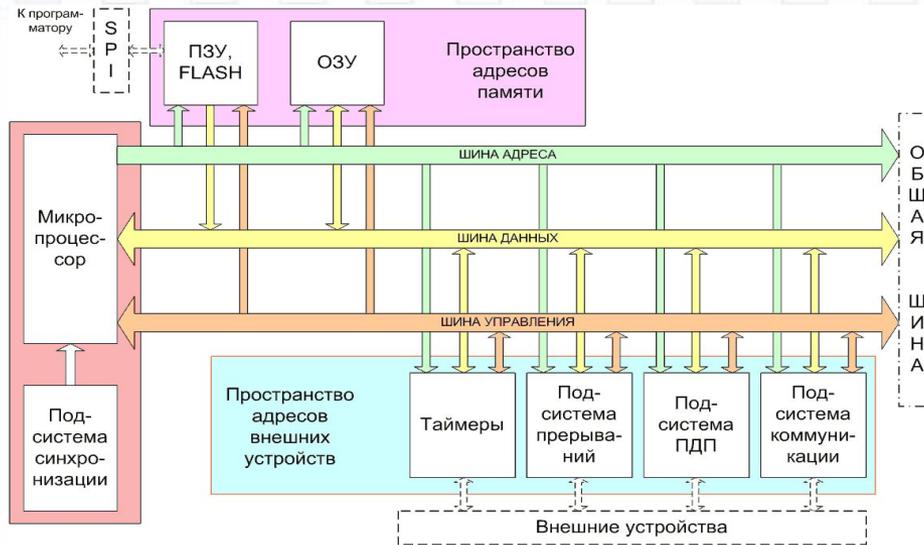


Передача адресов происходит по шине адресов. Адрес, по которому осуществляет доступ, определяет процессор. Эта шина является выходной однонаправленной по отношению к процессору.

Если обозначить разрядность адресной шины n , то наибольшее число адресов, к которым можно обращаться по этой шине, равно 2^n . Так, для 16-разрядной шины это число равно 64К, для 24-разрядной - 16М, для 32-разрядной - 4G (где $1К=2^{10}$, $1М=2^{24}$, $1G = 2^{32}$). Распределяемая область адресов называется **адресным пространством**.

- Все **адресное пространство** МПС разделено на *пространство адресов памяти* и пространство адресов *внешних устройств* - иначе устройств ввода-вывода. Это означает, что по одному и тому же адресу, выставляемому процессором на шину адреса, возможно обращение (обмен данными) к ячейке памяти или контроллеру внешнего устройства. Куда конкретно произойдет обращение, в этом случае, определяют сигналы шины управления. *Адреса ОЗУ и ПЗУ находятся в пространстве адресов памяти.*
- Остальные устройства находятся, *как правило*, в пространстве адресов ввода-вывода и называются внешними устройствами. Внешние устройства подключаются посредством специальных программируемых устройств сопряжения – интерфейсов.

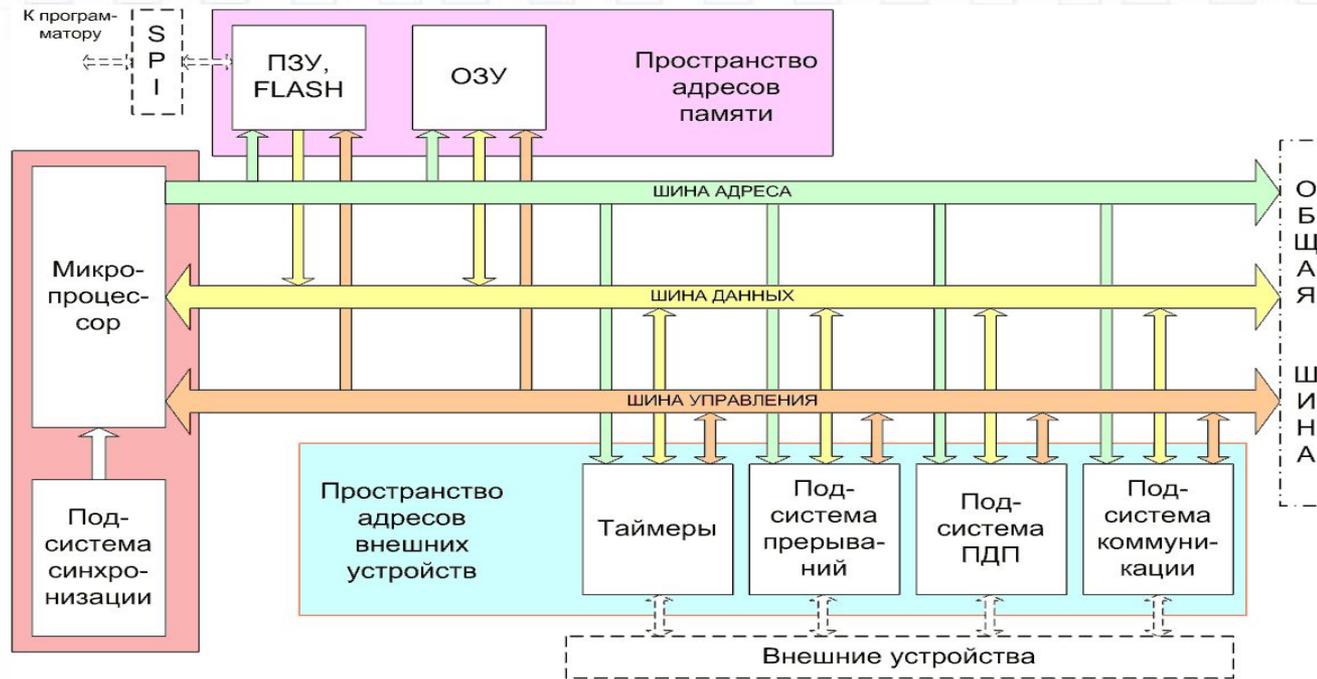
Шина управления.



Шина управления служит для передачи различных управляющих сигналов, часть которых является *выходными сигналами*, а другая часть - *входными сигналами* по отношению к процессору. Конкретный состав этих сигналов зависит от типа процессора.

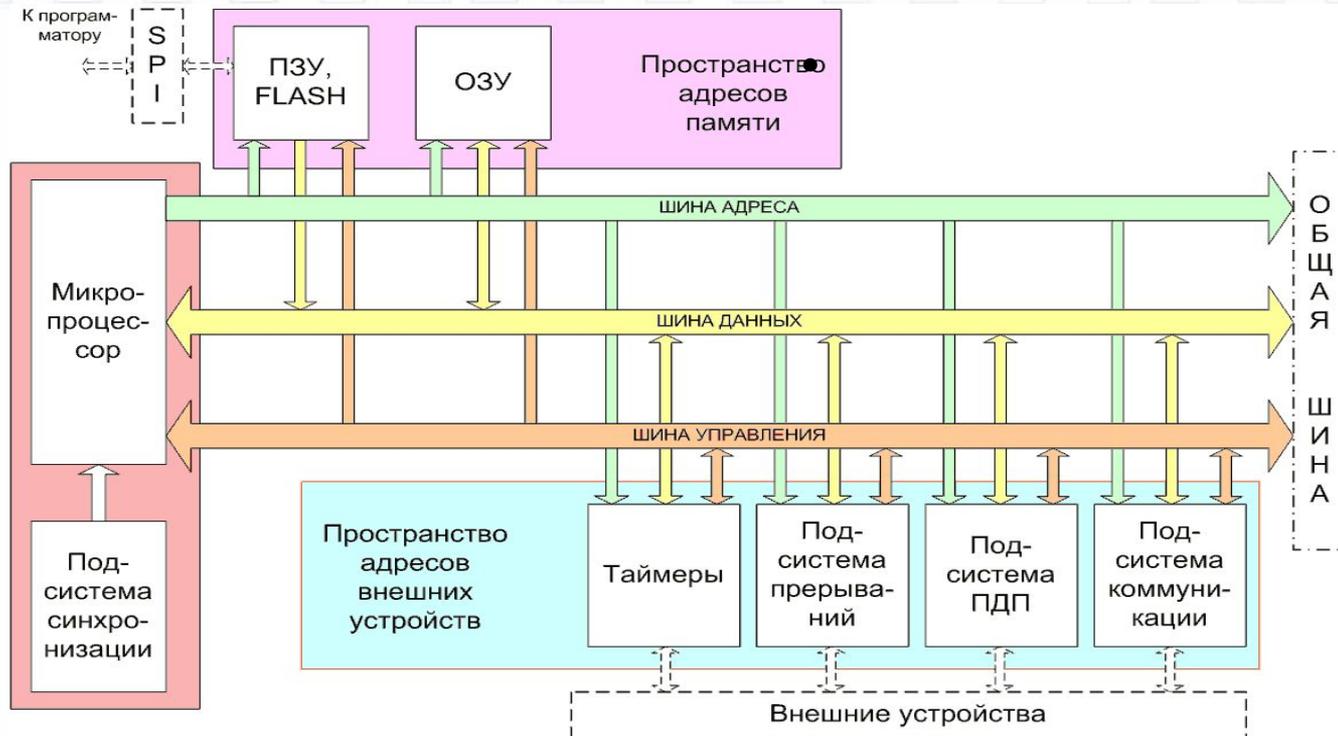
- Основными сигналами, передаваемыми по этой шине, являются сигналы управления записью-чтением из памяти и сигналы управления вводом-выводом данных из соответствующего порта. Если применяется способ изолированного ввода-вывода, передаются четыре управляющих сигнала:
 - MEMRD (MRD) - (“memory read” - считывание данных из памяти);
 - MEMWR(MWR) - (“memory write” - запись данных в память)
 - IORD - (“in-out read” - ввод данных из порта ввода)
 - IOWR - (“in-out write” - вывод данных в порт вывода).
- В случае ввода-вывода с отображением адресов УВВ на адреса памяти, порты ввода-вывода и память не различаются, обращение производится одними и теми же командами, генерирование специальных сигналов управления не происходит, поэтому можно использовать только сигналы RD-чтение и WR-запись.

Таймеры



- **Таймеры** предназначены для
 - формирования временных интервалов,
 - подсчета входных сигналов,
 - деления частоты.
- Применяются, например, при настройке скорости приема-передачи последовательного канала связи.
- Размещаются, как правило, в пространстве адресов ввода-вывода.

Подсистема прерываний



- **Контроллер прерываний** осуществляет прием сигналов - запросов аппаратных прерываний от различных внешних устройств, генерирование сигнала запроса прерывания процессору, генерирование векторов аппаратных прерываний, осуществляет механизм приоритета прерываний.

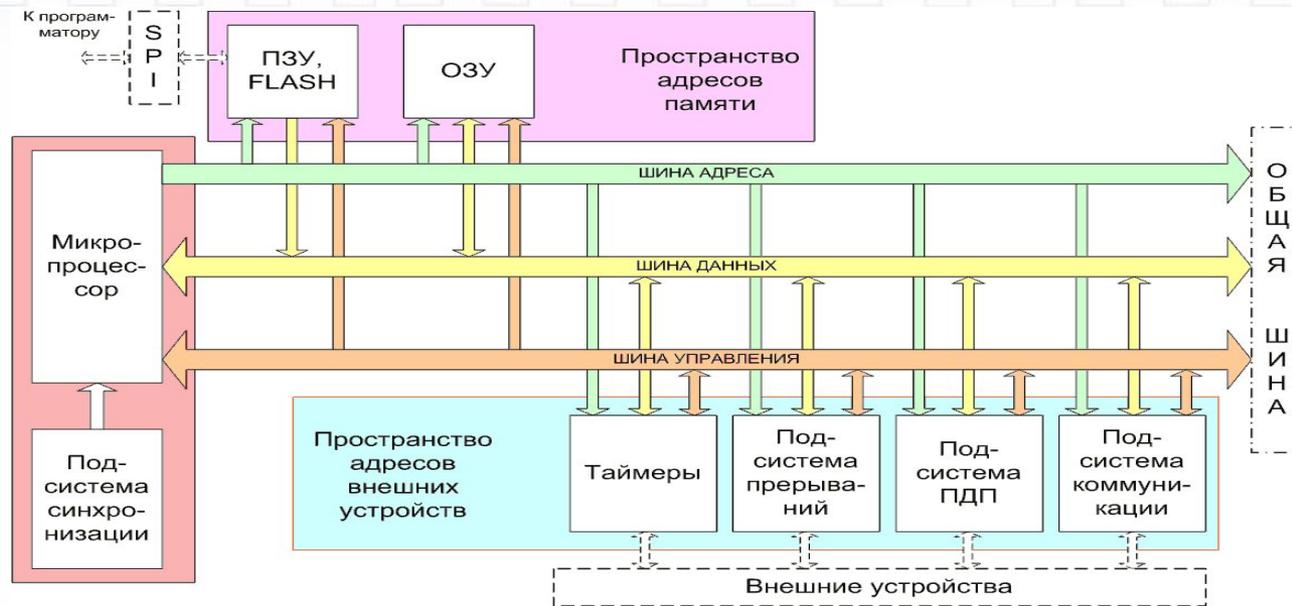
Подсистема прерываний

- Во всех этих ситуациях микропроцессор **завершает выполнение очередной команды** и заносит в стек текущее содержимое программного счетчика (**PC** – program counter), которое является адресом возврата к прерванной программе после реализации подпрограммы обслуживания прерывания, и содержимое регистра состояний (**SR** – status register (A+F)).
- Если запрос прерывания поступает от внешнего устройства, то процессор формирует сигнал **подтверждения прерывания**, который информирует устройство, инициировавшее прерывание, о том, что начато обслуживание данного запроса.
- Затем в счетчик **PC** загружается **вектор прерывания** - начальный адрес соответствующей подпрограммы обслуживания. Векторы являются **адресами входов в подпрограммы обслуживания** и хранятся в **таблице векторов прерываний**, которая обычно записывается в ОЗУ или поступают по шине данных от контроллера прерываний. Адреса могут быть также фиксированы на аппаратном уровне, т.е. их адреса поменять невозможно.
- **Завершается подпрограмма обслуживания прерывания специальной командой возврата из прерывания**, по которой происходит **выборка из стека содержимого PC и SR и загрузка его обратно в эти регистры**, обеспечивая возвращение к выполнению прерванной программы.

Прерывания

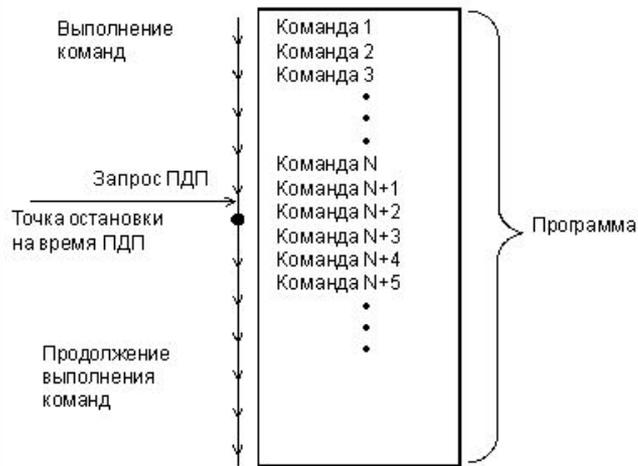
- Аппаратные прерывания могут быть **маскируемые** или **немаскируемые**.
- Запросы **маскируемых прерываний** обслуживаются только в том случае, если они были разрешены с помощью специальных команд. Можно разрешить или запретить обслуживание поступивших аппаратных прерываний при выполнении определенных программ или их фрагментов.
- При одновременном поступлении нескольких запросов обслуживание запросов реализуется в соответствии с их **приоритетом**. В ряде микропроцессорных систем для обеспечения приоритетного обслуживания запросов от многих внешних устройств включаются специальные микросхемы - **контроллеры прерываний**. Некоторые типы микропроцессоров имеют внутренние контроллеры для организации приоритетных прерываний.
- **Немаскируемые запросы** на прерывания обслуживаются в первоочередном порядке и не могут быть маскированы. Обычно микропроцессор имеет один вход для подачи немаскируемых запросов, которые формируются при возникновении каких-либо аварийных ситуаций, например, сбоя по питанию.
- Причинами **исключений** могут быть различные ошибки и нестандартные ситуации, возникающие при работе системы. Различные типы микропроцессоров контролируют разные варианты такого рода ситуаций. Типичными причинами исключений являются, например, использование нулевого делителя при выполнении команды деления (деление на 0); выборка неправильного кода команды; выход за границы разрешенного сегмента памяти, поступление команд, выполнение которых запрещено при данном режиме функционирования микропроцессора и ряд других. Соответствующие причины возникновения ситуации исключений зависят от конкретных типов микропроцессоров.
- В простейших микропроцессорах нет механизма обслуживания исключений

Прямой доступ к памяти



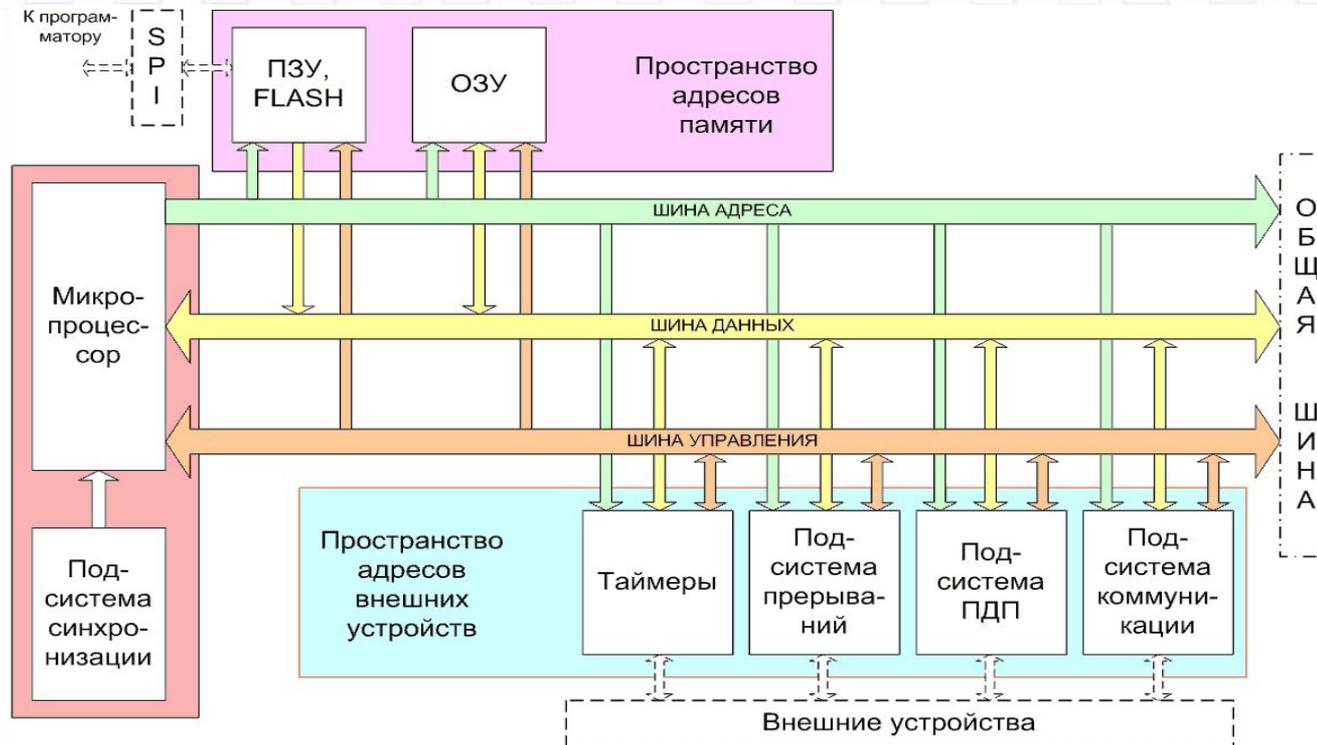
- **Контроллер прямого доступа к памяти** осуществляет механизм прямого доступа к памяти (**DMA, ПДП**). Режим прямого доступа к памяти DMA (Direct Memory Access) используется, если необходимо произвести **пересылку значительного массива информации между ОЗУ и каким-либо внешним устройством**, которое подает в систему соответствующий запрос. Реализация такой пересылки с помощью соответствующей программы обмена требует выполнения отдельной команды пересылки для передачи каждого байта или слова. При этом необходим определенный объем памяти для хранения программы и требуется значительное время для ее выполнения.

Механизм ПДП



- В большинстве современных микропроцессорных систем пересылка массивов информации обеспечивается с помощью специальных устройств - **контроллеров DMA**, которые реализуют режим прямого доступа к памяти.
- При поступлении запроса от внешнего устройства контроллер выдает соответствующий сигнал микропроцессору. **Микропроцессор завершает очередной цикл обмена по системной шине и отключается от нее, то есть переводит свои выходы, подключенные к шинам в отключенное (высокоимпедансное) состояние.** При этом микропроцессор выдает контроллеру DMA сигнал разрешения на осуществление прямого доступа.
- Получив сигнал разрешения, контроллер принимает на себя управление системой. Он выдает на шину адреса ячеек ОЗУ, с которыми выполняется текущий цикл обмена, формирует необходимые сигналы, определяющие режим работы ОЗУ (запись или считывание) и интерфейсного устройства, через которое производится пересылка информации (ввод или вывод).
- Взаимодействие между микропроцессором и контроллером DMA производится по соответствующим линиям шины управления.
- **Предварительно контроллер DMA должен быть запрограммирован для выполнения указанных функций. В соответствующие регистры контроллера DMA записываются начальные адреса массивов памяти, с которых начинается процесс обмена, и размеры массивов, подлежащих пересылке.**
- Обычно контроллер DMA обслуживает запросы от нескольких внешних устройств, поэтому он **программируется на реализацию определенного приоритета обслуживания в случае одновременного поступления нескольких запросов.** Программирование контроллера производится путем записи необходимых управляющих слов в **управляющие регистры контроллера DMA.**

Подсистема коммуникации



- **Подсистема коммуникации** осуществляет связь МП системы с удаленными объектами.
- В нее могут входить параллельные периферийные адаптеры, последовательные адаптеры, сетевые адаптеры и т.п.