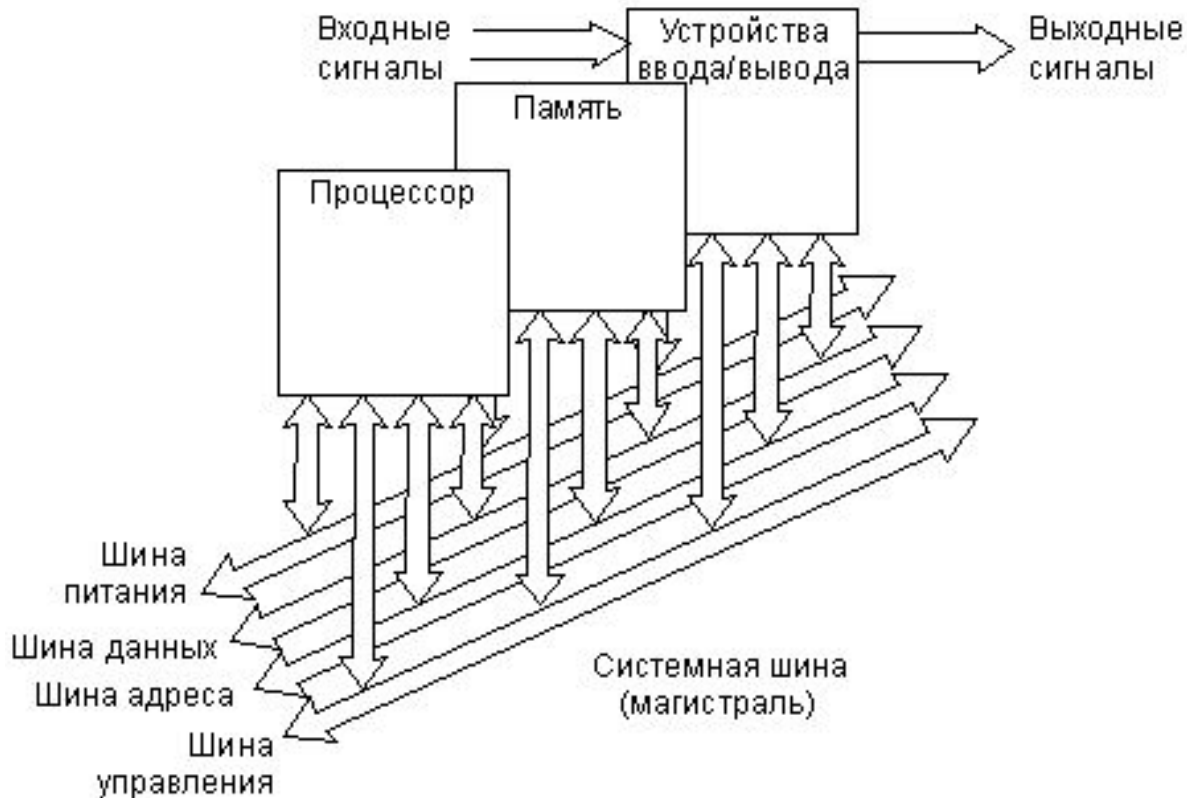


Структура микропроцессорной системы

Структура микропроцессорной системы



Шина адреса (однаправленная) – её разрядность определяет количество используемых в микропроцессорной системе устройств.

Шина данных (двунаправленная) – её разрядность определяет производительность микропроцессорной системы.

Шина управления (разнонаправленная) – её разрядность определяет гибкость микропроцессорной системы.

Шина питания (однаправленная) – определяет универсальность микропроцессорной системы.

Процессор

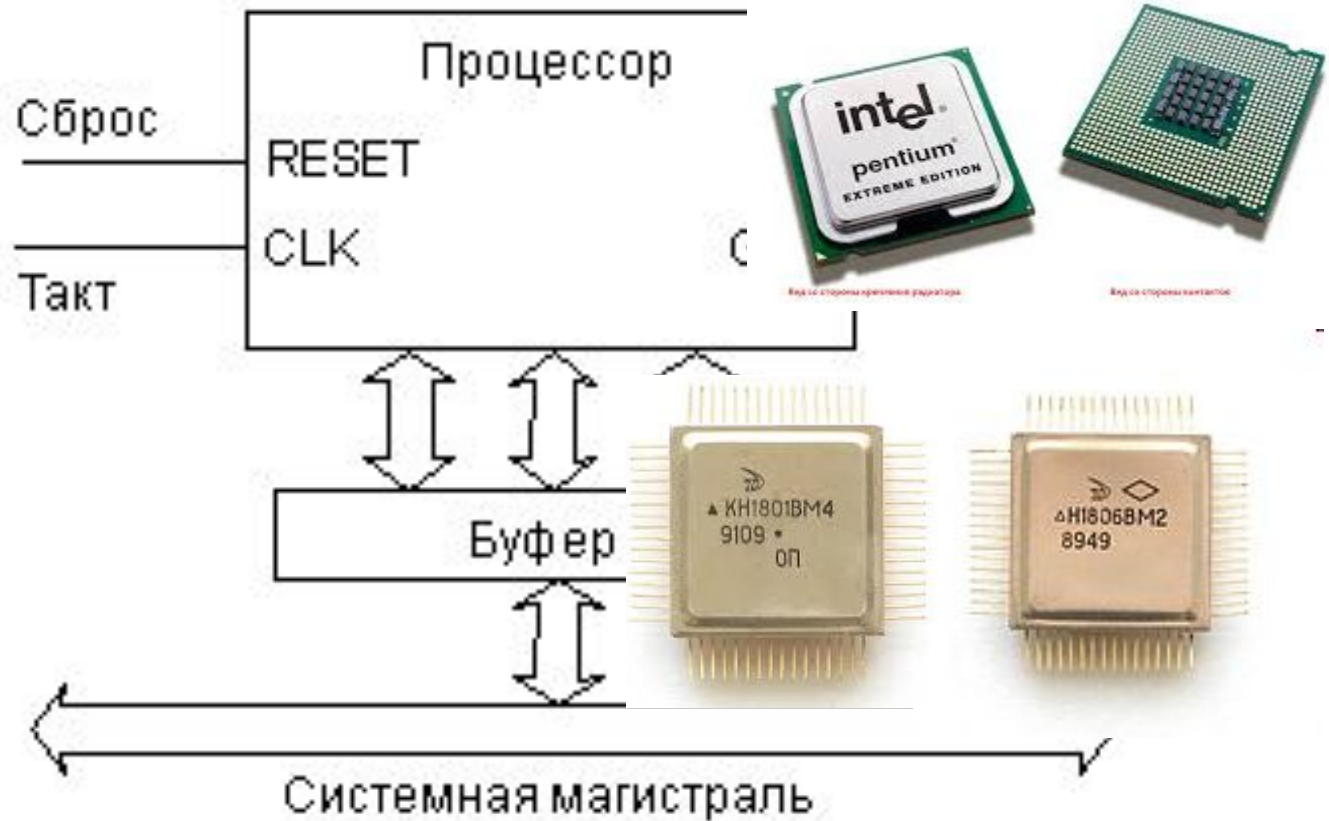


Рис. 1. Схема включения процессора

Процессор – главный элемент на системной магистрали

Конструктивно **Процессор** может представлять собой **отдельную микросхему** или же **часть микросхемы** (в случае микроконтроллера) или **комплект из нескольких микросхем**

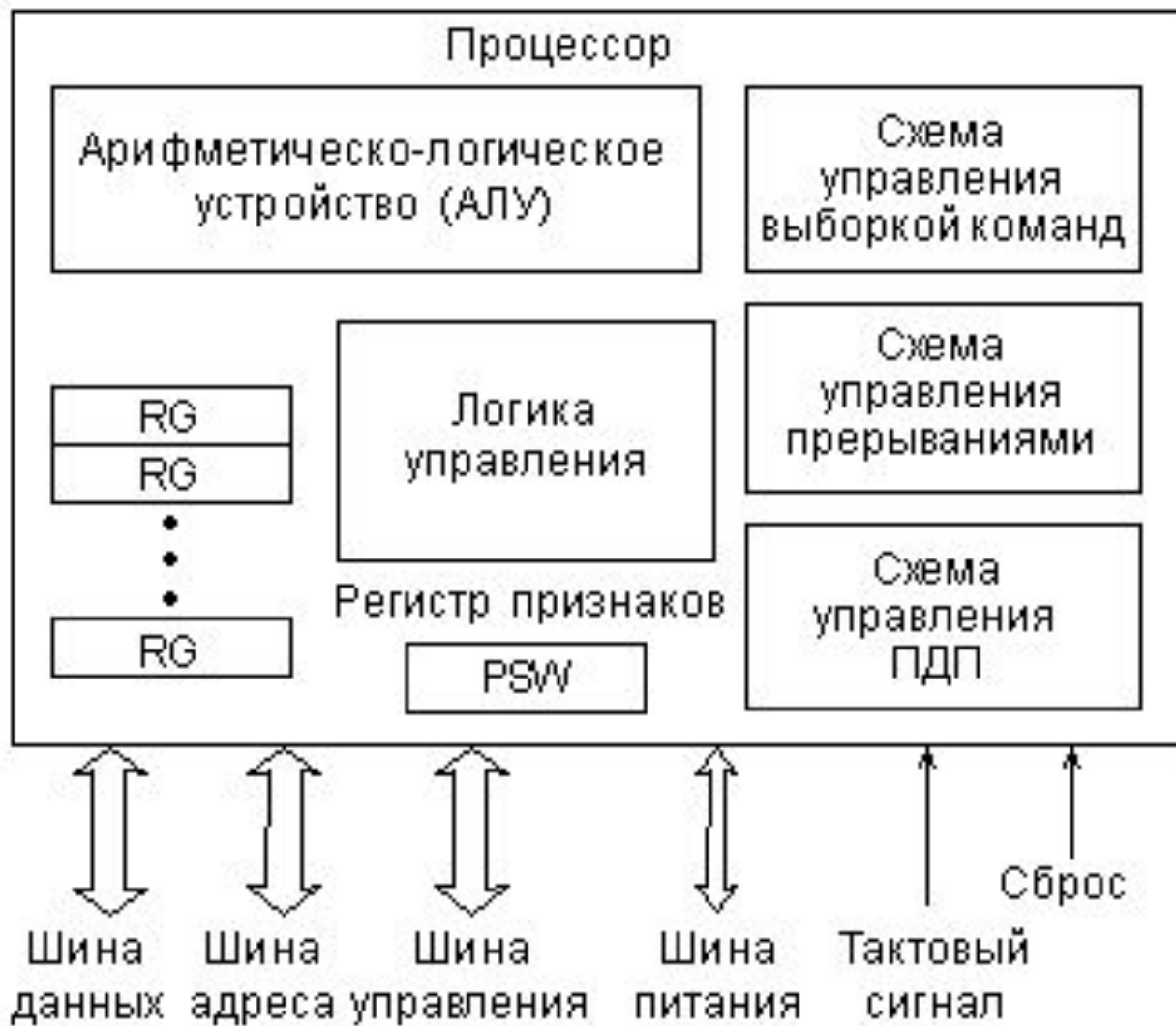


Рис. 2. Внутренняя структура микропроцессора.

Процессор **обязательно имеет выходы** трех шин:

шины адреса, **шины данных** и **шины управления**

Важнейшие характеристики процессора — это количество разрядов его **шины данных**, количество разрядов его **шины адреса** и количество управляющих сигналов в **шине управления**.

Разрядность шины данных определяет **скорость** работы системы.

Разрядность шины адреса определяет **допустимую сложность** системы.

Количество линий управления определяет **разнообразие режимов обмена** и **эффективность обмена** процессора с другими устройствами системы.

Чем больше тактовая частота процессора, **тем он быстрее** выполняет команды. **Быстродействие процессора** определяется не только тактовой частотой, но и **особенностями его структуры** (архитектуры)

Основные характеристики определяющие производительность микропроцессора:

1. **Набор регистров** для хранения промежуточных данных
2. **Система команд** процессора
3. **Способы адресации операндов** в пространстве памяти
4. **Организация процессов выборки** и использования команд

Основные функции любого микропроцессора:

1. **выборка (чтение)** выполняемых команд;
2. **ввод (чтение)** данных из памяти или устройства ввода/вывода;
3. **вывод (запись)** данных в память или в устройства ввода/вывода;
4. **обработка данных** (операндов), в том числе арифметические и/или логические операции над ними;
5. **адресация памяти**, то есть задание номера ячейки памяти, с которой будет производиться обмен;
6. Организация дополнительных способов взаимодействия с памятью или устройствами ввода/вывода – **режим обработки прерываний и режим прямого доступа к памяти.**

Архитектурой микропроцессора называется комплекс его аппаратных и программных средств, предоставляемых пользователю.

В это общее понятие входит 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 и функционирования процессора обычно используется его представление в виде совокупности программно-доступных регистров, образующих регистровую или программную модель.

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.).

С точки зрения системы команд и способов адресации операндов в основном различают **две архитектуры процессоров**

RISC (*Reduced Instruction Set Computer*) **процессоры** (процессоры с сокращённым набором команд)

CISC (*Complicated Instruction Set Computer*) **процессоры** (процессоры с полным набором команд)

Система команд может быть:

1. **Ортогональная** – все команды фиксированной длины, имеют одинаковое время исполнения (преимущественно за один цикл (такт) синхронизации, равноправное использование всех регистров процессора)
2. **Неортогональная** – не все команды могут использовать весь набор возможных способов адресации применительно к любому из регистров процессора

В микропроцессорах с полным набором команд (**CISC** микропроцессоры) используется уровень микропрограммирования для того, чтобы декодировать и выполнить команду микропроцессора (т.е. используются **БМУ** – **Б**лок **М**икропрограммного **У**правления и **УП** - **У**правляющая **П**амять).

Достоинства

- **Формат команды не зависит от конструкции (аппаратуры) процессора.**
- **На одной и той же аппаратуре при смене микропрограммы могут быть реализованы различные микропроцессоры.** С точки зрения пользователя у микропроцессора только увеличивается производительность, снижается потребление энергии, уменьшаются габариты устройств.
- **Изменение конструкции (аппаратуры) никак не влияет на программное обеспечение микропроцессора.**

Недостаток

- Производители микросхем стараются увеличить количество команд, которые может выполнять микропроцессор, тем самым увеличивая сложность микропрограммы и **замедляя выполнение каждой команды в целом.**

В микропроцессорах с сокращённым набором команд (**RISC** процессоры) **декодирование и исполнение команды производится аппаратно**, поэтому **количество команд ограничено минимальным набором**. Для реализации более сложных операций приходится **комбинировать команды**

В RISC микропроцессорах команда и микрокоманда **совпадают**.

Преимущество

Достаточно сложная команда, например, умножение с накоплением может быть в принципе выполнена за один такт (не требуется выполнение микропрограммы), так как выполняется специализированным устройством.

Недостаток

Для выполнения сложной команды требуется выполнение большого набора микрокоманд реализующих отдельную микропрограмму.

В большинстве случаев быстродействие RISC процессоров выше чем CISC процессоров

При выборе процессора нужно учитывать все параметры в целом, т.к. тактовая частота RISC процессора может оказаться **значительно ниже по сравнению с CISC процессором** (особенно если в нём применяются специальные меры по повышению производительности), разрядность команды может оказаться выше чем у CISC процессора (что чаще всего и бывает). В результате общий **объём исполняемой программы для RISC процессора превысит объём** подобной программы для CISC процессора.

Фон-Неймановская (Принстонская) архитектура организации процессора

Принципы фон Неймана (1946 г.)

1. Принцип программного управления.

Программа состоит из набора команд, которые выполняются процессором друг за другом в определенной последовательности.

2. Принцип однородности памяти.

Как программы (команды), так и данные хранятся в одной и той же памяти (и кодируются в одной и той же системе счисления – чаще всего двоичной). Над командами можно выполнять такие же действия, как и над данными.

3. Принцип адресуемости памяти.

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

Единая шина для данных и команд. В составе системы присутствует одна общая память, как для данных, так и для команд



Рис. 3. Архитектура с общей шиной команд и данных

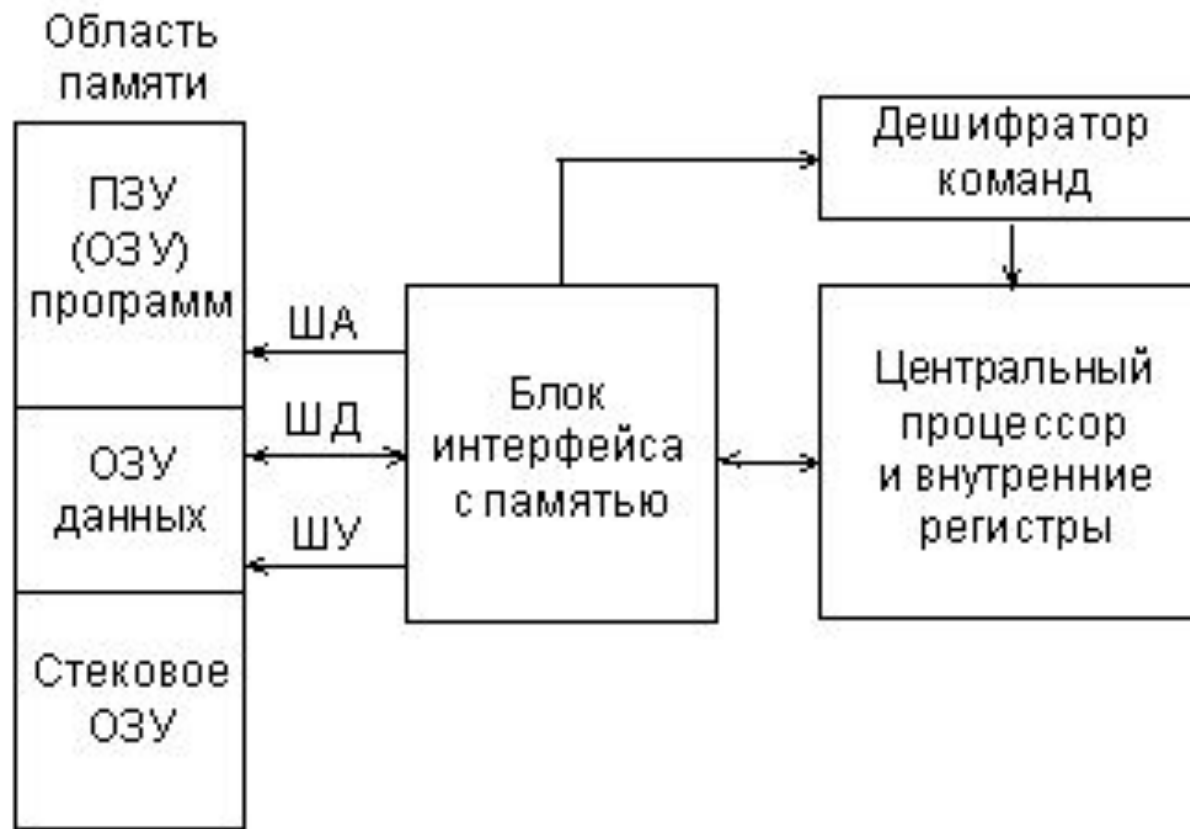


Рис. 4. Структура МПС с фон-Неймановской архитектурой

Основное преимущество архитектуры Фон-Неймана – **упрощение устройства МПС**, так как реализуется обращение только к одной общей памяти. Кроме того, использование единой области памяти **позволяет оперативно перераспределять ресурсы между областями программ и данных**, что увеличивает гибкость МПС с точки зрения разработчика программного обеспечения.

Размещение **стека** в общей памяти облегчает доступ к его содержимому.

Гарвардская архитектура организации процессора

Гарвардская архитектура (разработана Говардом Эйкеном в конце 1930-х) предполагает наличие в системе **отдельной памяти для данных** и **отдельной памяти для команд**. Обмен процессора с каждым из двух типов памяти происходит по своей индивидуальной шине.

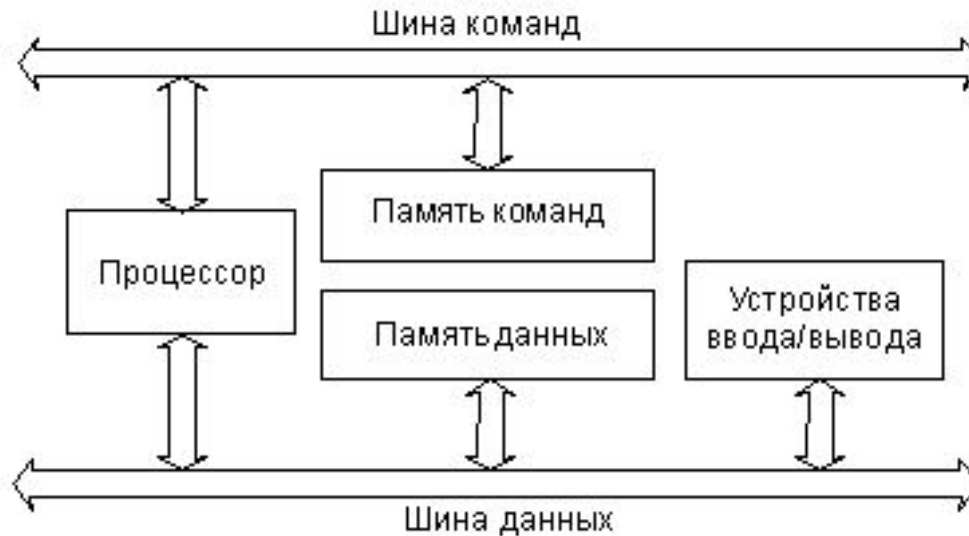


Рис. 5. Архитектура с отдельными шинами данных и команд

Недостатки. В Гарвардской архитектуре невозможны многие методы программирования (например, программа не может во время выполнения менять свой код; невозможно динамически перераспределять память между программным кодом и данными),

Преимущества. Гарвардская архитектура позволяет более эффективно выполнять работу в случае ограниченных ресурсов, поэтому она часто применяется в микроконтроллерах (во встраиваемых системах).

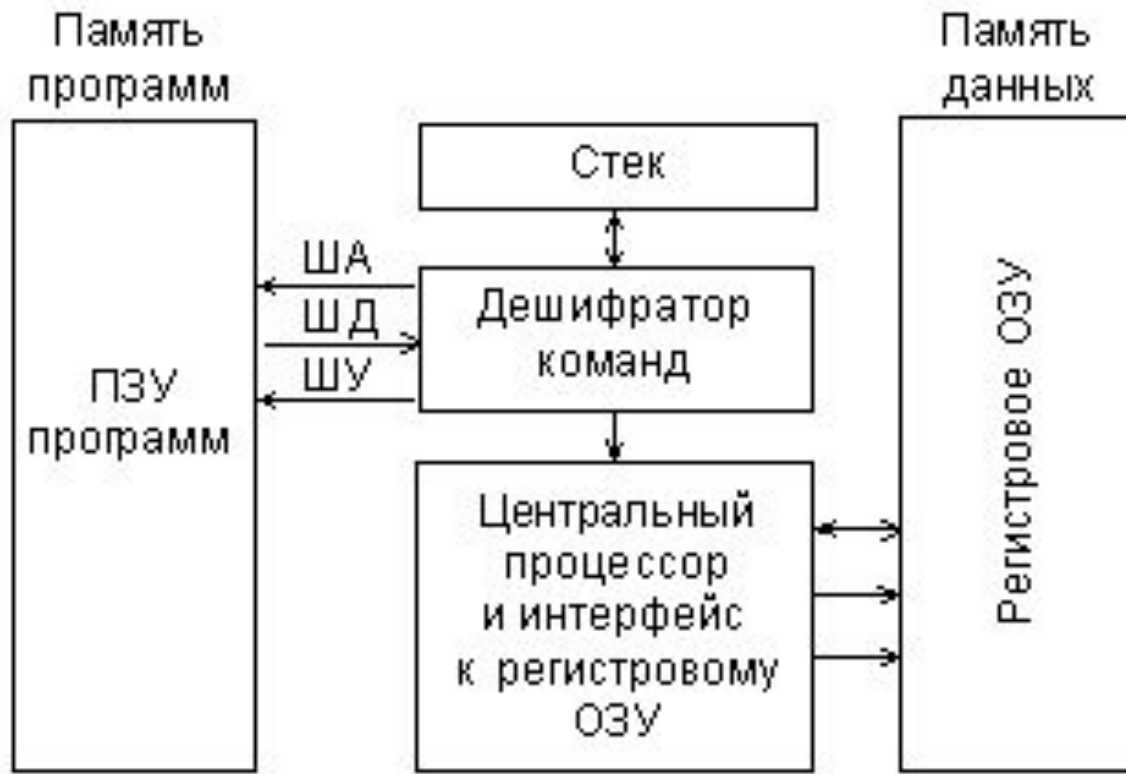


Рис. 6. Структура МПС с гарвардской архитектурой

Гарвардская архитектура дает определенные преимущества разработчикам **автономных систем управления** перед фон-Неймановской, т.к. как гибкость и универсальность последней не имеют большого значения в данном классе задач.

Гарвардская архитектура **обеспечивает потенциально более высокую скорость выполнения программы** по сравнению с фон-Неймановской за счет возможности реализации параллельных операций.

Особенности структуры микропроцессора для цифровой обработки сигналов

В отличие от микропроцессоров с универсальной структурой микропроцессоры предназначенные для **цифровой обработки сигналов** (ЦОС, **DSP – Digital Signal Processing**) обладают определённой структурной специализацией, обусловленной характером решаемых ими задач в составе вычислительных средств первичной обработки информации.

В настоящий момент имеются два основных структурных направления развития быстродействующих микропроцессоров этой группы:

1. **Специализированные микропроцессоры для ЦОС**, структура которых ориентирована на решение относительно широкого класса задач ЦОС, множество которых ограничено только допустимым частотным диапазоном обработки сигналов в реальном времени
2. **Микросхемы с узкоспециализированной структурой**, обеспечивающей достижение наибольшей производительности при решении конкретной задачи.

Особенности архитектуры ЦПОС

Особенностью программ цифровой обработки сигналов

- программы выполняются, как правило, в реальном масштабе времени - по мере поступления входного сигнала, что придает критическую важность вопросам повышения быстродействия;
- программы содержат много логических и особенно арифметических операций и практически не содержат программ перехода;
- происходит постоянный и быстрый ввод вывод данных, зачастую в аналоговой форме
- программы относительно короткие и достаточно редко изменяются, зачастую остаются неизменными на протяжении всего срока эксплуатации процессора.

Особенности архитектуры

- часто используется Гарвардская архитектура;
- большая (иногда нестандартная) разрядность обрабатываемых данных – 24, 32, 48, 64, 128, что позволяет увеличить диапазон обрабатываемых чисел без применения формата с плавающей запятой или обрабатывать по несколько чисел одновременно;
- аппаратные блоки, предназначенные для ускорения выполнения команды умножения – **сдвиговые регистры, матричные умножители**;
- память команд и данных расположены на самом кристалле процессора;
- возможность параллельного выполнения нескольких операций одновременно, например, ввода вывода и арифметических команд;
- все команды имеют одинаковую длину и выполняются за одинаковое время, что позволяет использовать счетчик команд для отсчета временных интервалов.
- ортогональная система команд.

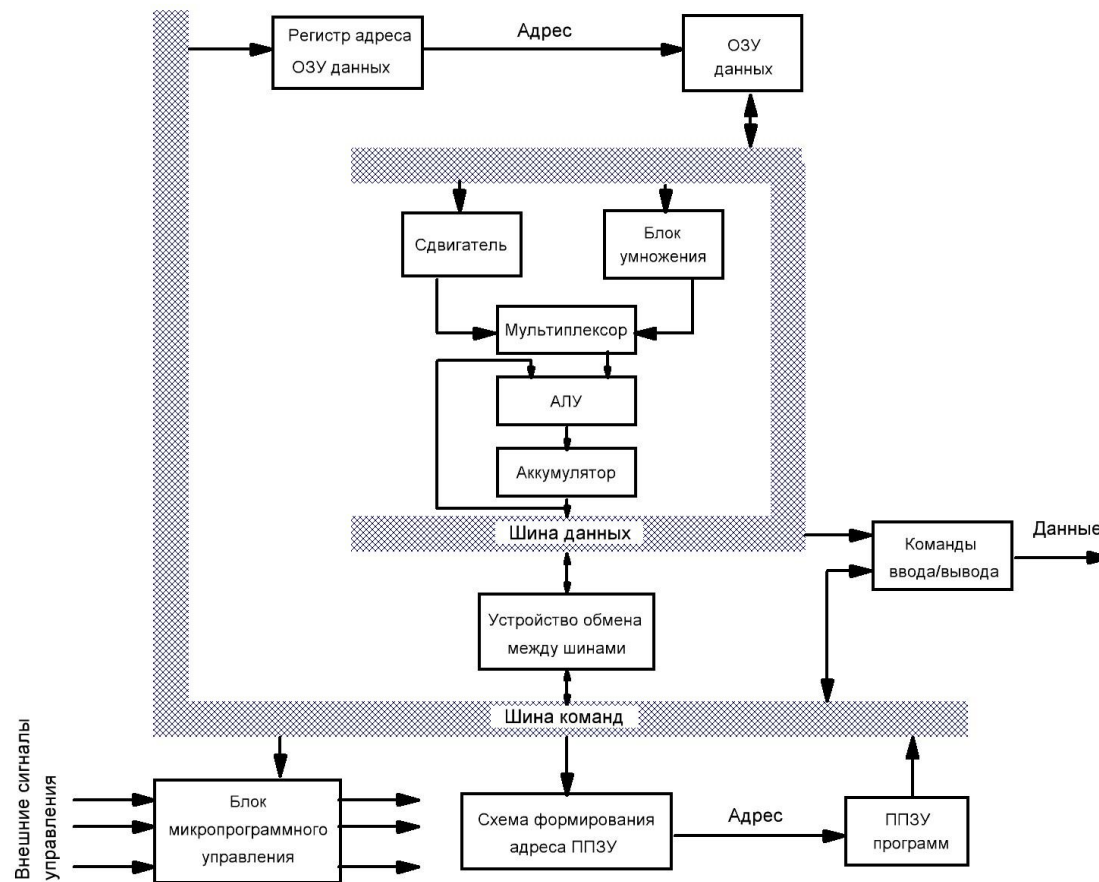


Рис. 7. Типовая структура процессора цифровой обработки сигналов

Особенностью БИС с полной структурной специализацией является их функциональная ориентация на выполнение **фиксированной базовой операции**, специфичной для алгоритмов ЦОС, что даёт возможность наиболее полно использовать их естественный параллелизм.

Основная операция ЦОС обеспечивает выполнение **операции умножения с накоплением**: $Z_i = Z_{i-1} + X_i \cdot Y_i$

Использование ПЛИС для ЦОС

Недостаток микропроцессоров для ЦОС состоит в их **структурной ограниченности** направленной на получение предельной производительности при выполнении строго определённых вычислений, что приводит к **ограничению числа одновременно задействованных операционных устройств** и поэтому не позволяющего эффективно использовать естественный параллелизм алгоритмов ЦОС. Поэтому важное значение имеет **второе направление** предполагает **максимальное распараллеливание процесса обработки информации** с целью получения наибольшего выигрыша в производительности при решении конкретной задачи.

Такой подход реализован на основе **универсальных вычислительных ячеек (УВЯ)** на базе **программируемых логических матриц (ПЛМ или ПЛИС)**.

В случае использования УВЯ для каждой вычислительной процедуры используется своя группа таких вычислительных ячеек (так называемый **настраиваемый процессор**).

Программа загружается один раз перед началом решения задачи, а обработка потоков информации производится без промежуточного запоминания результатов, т.е. **работает по принципу конвейера**, который на одном конце с каждым тактом загружается информацией, а на другом конце конвейера производится выгрузка результата обработки информации.

Общая структура вычислительной системы обеспечивает параллельную организацию множества конвейеров, в том числе охваченных обратными связями.

Особенностью такой вычислительной **системы** является **отсутствие устройств управления, ОЗУ, магистрали данных, а также наличие большого количества информационных входов и выходов.**

Подобные структуры вычислительных систем ЦОС на однородных средах называются **систолическими структурами**.

Функции памяти

Память микропроцессорной системы выполняет функцию **временного** или **постоянного** хранения данных и команд.

Объем памяти определяет допустимую сложность выполняемых системой алгоритмов, а также в некоторой степени и скорость работы системы в целом.

Информация в памяти хранится в ячейках, количество разрядов которых равно количеству разрядов шины данных процессора. Обычно оно **кратно 8** (например, 8, 16, 32, 64).

Допустимое количество ячеек памяти определяется количеством разрядов шины адреса как 2^N , где N — количество разрядов шины адреса.

Чаще всего объем памяти измеряется в байтах независимо от разрядности ячейки памяти.

килобайт — 2^{10} или 1024 байта (обозначается Кбайт),

мегабайт — 2^{20} или 1 048 576 байт (обозначается Мбайт),

гигабайт — 2^{30} байт (обозначается Гбайт),

терабайт — 2^{40} (обозначается Тбайт)

Совокупность ячеек памяти называется обычно **пространством памяти** системы.

Для подключения модуля памяти к системной магистрали используются **блоки сопряжения**, которые включают в себя:

селектор (дешифратор) адреса,

схему обработки управляющих сигналов магистрали и

буфер данных

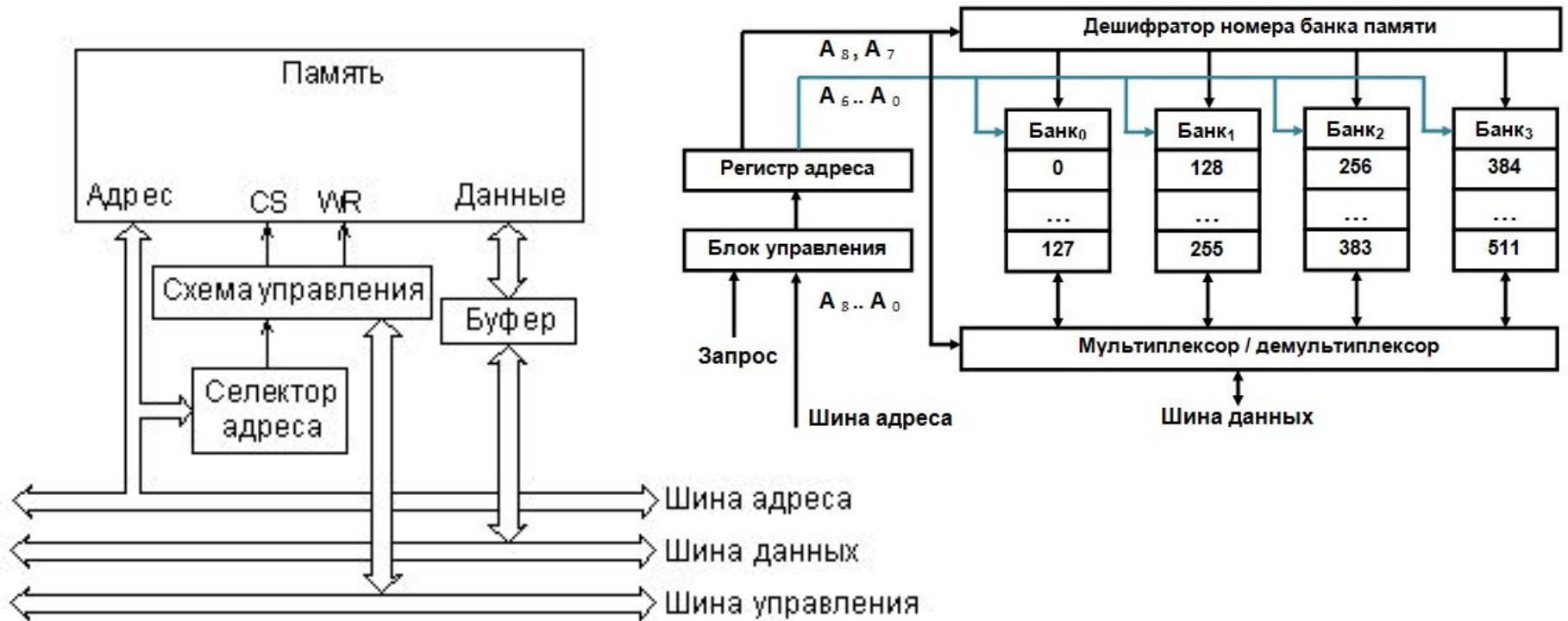


Рис. 8. Структура модуля памяти

Специальные области памяти

1. Стек

Память для стека или **стек** (*Stack*) — это часть оперативной памяти, предназначенная для временного хранения данных в режиме **LIFO** (*Last Input — First Output*) или **памятью магазинного типа** (например, в магазине автомата патрон, установленный последним, будет извлечен первым).

Особенность стека по сравнению с другой оперативной памятью — это **заданный и неизменяемый способ адресации**.

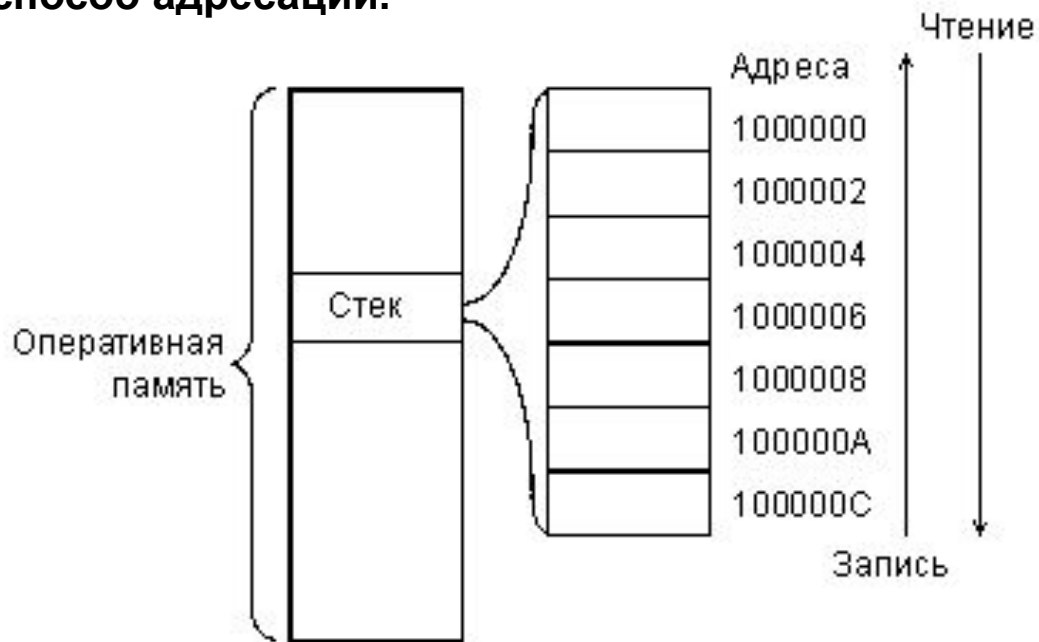


Рис. 9. Принцип работы стека

Необходимость такой адресации становится очевидной в случае многократно вложенных подпрограмм.

2. Таблица векторов прерываний

Под прерыванием (*interrupt*) в общем случае понимается не только обслуживание запроса внешнего устройства, но и **любое нарушение последовательной работы процессора.**

Любое прерывание обрабатывается через **таблицу векторов** (указателей) **прерываний.**

В таблице векторов прерываний в простейшем случае **находятся адреса** начала программ обработки прерываний, которые и называются **векторами.**

Длина таблицы может быть значительной (до нескольких сот элементов).

Обычно **таблица векторов прерываний располагается в начале пространства области памяти** (в ячейках памяти с малыми адресами).

Адрес каждого вектора (или адрес начального элемента каждого вектора) представляет собой номер прерывания.



Рис. 10. Упрощенный алгоритм обработки прерывания

В случае **аппаратных прерываний** номер прерывания или задается устройством, запросившим прерывание (при векторных прерываниях), или же задается номером линии запроса прерываний (при радиальных прерываниях).

Программное прерывание тоже обслуживается через таблицу векторов прерываний, но номер прерывания указывается в составе команды, вызывающей прерывание.

В конце программы обработки прерываний **обязательно должна располагаться команда выхода** из прерывания, выполнив которую, процессор возвращается к выполнению прерванной основной программы.

3. Память устройств (ввода/вывода), подключенных к системной шине.

Процессор получает возможность **обращаться к внутренней памяти устройств ввода/вывода** или каких-то еще подключенных к системной шине устройств, **как к своей собственной системной памяти**. Обычно окно в пространстве памяти, выделяемое для этого, не слишком большое.

Распределение адресных пространств памяти процессора (системной памяти) и адресов устройств ввода/вывода.

Существует два основных подхода :

1. **выделение** в общем адресном пространстве системы специальной области адресов для устройств ввода/вывода;
 - при обращении к устройствам ввода/вывода процессор может использовать те же команды, которые служат для взаимодействия с памятью. Но адресное пространство памяти должно быть уменьшено на величину адресного пространства устройств ввода/вывода.
2. **полное разделение** адресных пространств памяти и устройств ввода/вывода.
 - память занимает все адресное пространство микропроцессорной системы. Для общения с устройствами ввода/вывода применяются специальные команды и специальные стробы обмена на магистрали.

Функции устройств ввода/вывода

Устройства ввода/вывода обмениваются информацией с магистралью по тем же принципам, что и память.

Устройства ввода/вывода взаимодействуют еще и с внешними устройствами, цифровыми или аналоговыми.

Быстродействие устройств ввода/вывода может значительно отличаться от быстродействия остальной микропроцессорной системы.

Разнообразие устройств ввода/вывода неизмеримо больше, чем модулей памяти.

Устройств ввода/вывода могут иметь другие названия:

**устройства сопряжения,
контроллеры,
карты расширения,
интерфейсные модули и т.д.**

Объединяют все устройства ввода/вывода общие принципы обмена с магистралью и, соответственно, общие принципы организации узлов, которые осуществляют сопряжение с магистралью.



Рис. 11. Структура простейшего устройства ввода/вывода

Самые **простейшие устройства ввода/вывода** выдают на внешнее устройство код данных в **параллельном формате** и принимают из внешнего устройства код данных тоже в параллельном формате. Их называют **параллельными портами ввода/вывода**. Они **наиболее универсальны** и их часто вводят в состав микропроцессорной системы. Параллельные порты обычно имеются в составе микроконтроллеров. **Именно через параллельные порты микроконтроллер связывается с внешним миром.**

Более сложные устройства ввода/вывода (устройства сопряжения) **имеют** в своем составе **внутреннюю буферную оперативную память** и даже могут иметь **микроконтроллер**, на который возложено выполнение функций обмена с внешним устройством.

Каждому устройству ввода/вывода отводится свой адрес в адресном пространстве микропроцессорной системы. **Дублирование адресов должно быть исключено!** за этим должны следить разработчик и пользователь микропроцессорной системы.

Устройства ввода/вывода помимо программного обмена могут также поддерживать **режим обмена по прерываниям**. В этом случае они преобразуют поступающий от внешнего устройства сигнал запроса на прерывание в сигнал запроса прерывания, необходимый для данной магистрали (или в последовательность сигналов при векторном прерывании).

Если нужно использовать **режим ПДП**, устройство ввода/вывода должно выдать сигнал запроса ПДП на магистраль и обеспечить работу в циклах ПДП, принятых для данной магистрали.

В составе микропроцессорных систем, как правило, выделяются четыре специальные группы устройств ввода/вывода:

1. устройства интерфейса пользователя

а. ввода информации пользователем

контроллеры клавиатуры,
тумблеры,
отдельные кнопки,
мыши, трекбол,
джойстика и т.д.

б. вывода информации для пользователя;

контроллеры светодиодных индикаторов,
табло жидкокристаллических, плазменных и электронно-лучевых экранов

2. устройства ввода/вывода для длительного хранения информации;

а. дисководы (компакт-дисков или магнитных дисков), а также с накопителями на магнитной ленте.

3. таймерные устройства.

а. Эти устройства предназначены для того, чтобы микропроцессорная система могла выдерживать заданные временные интервалы, следить за реальным временем, считать импульсы и т.д.

4. устройства для подключения к информационным сетям (локальным и глобальным).