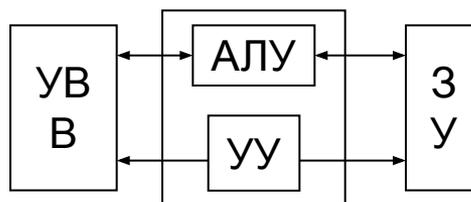


Архитектура ЭВМ

Архитектура ЭВМ – логическая организация и структура аппаратных ресурсов вычислительной системы и программного обеспечения (ПО).



Фон-Неймановская архитектура ЭВМ

УУ – узел управления;

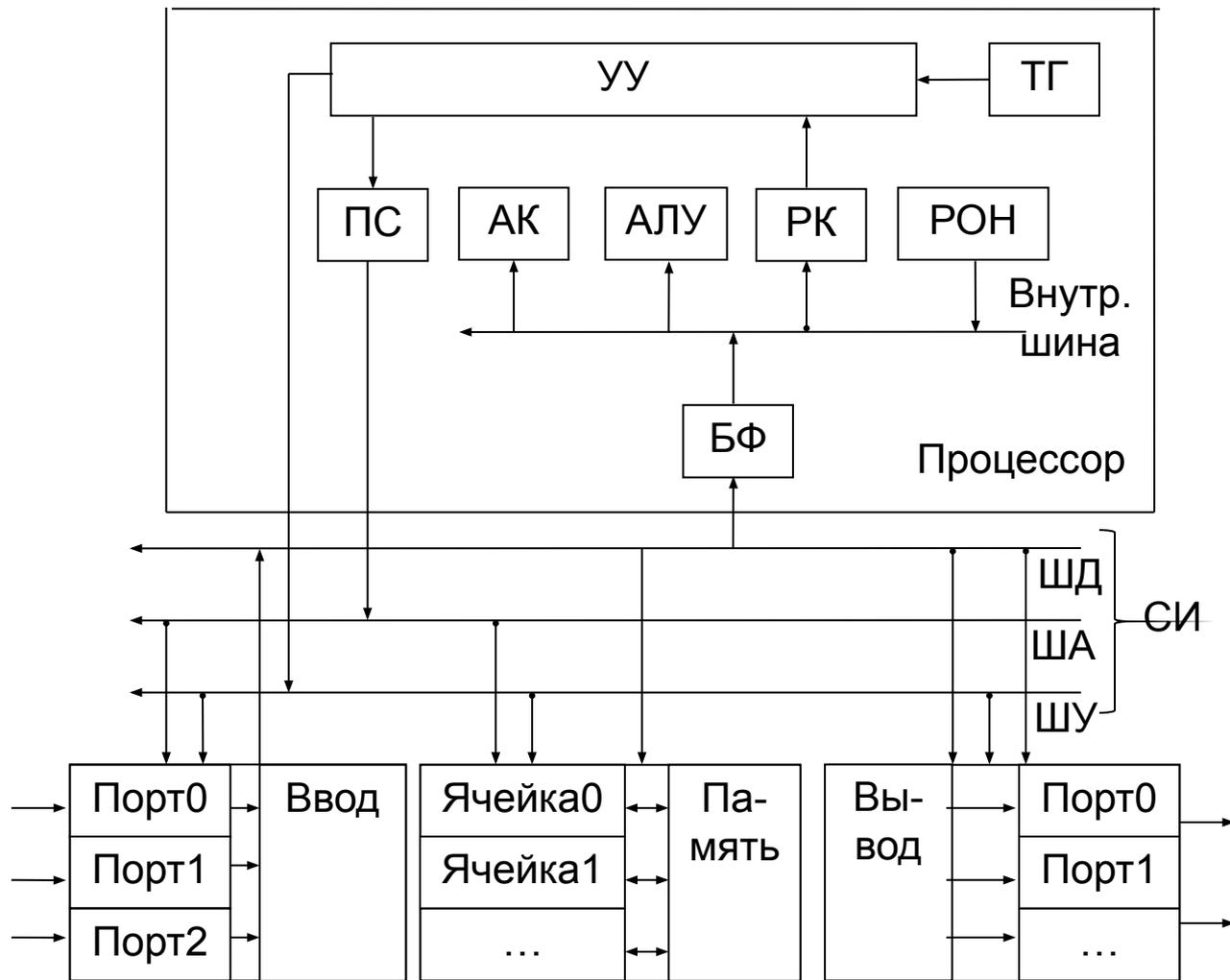
АЛУ – арифметико-логическое устройство;

УВВ – устройство ввода-вывода;

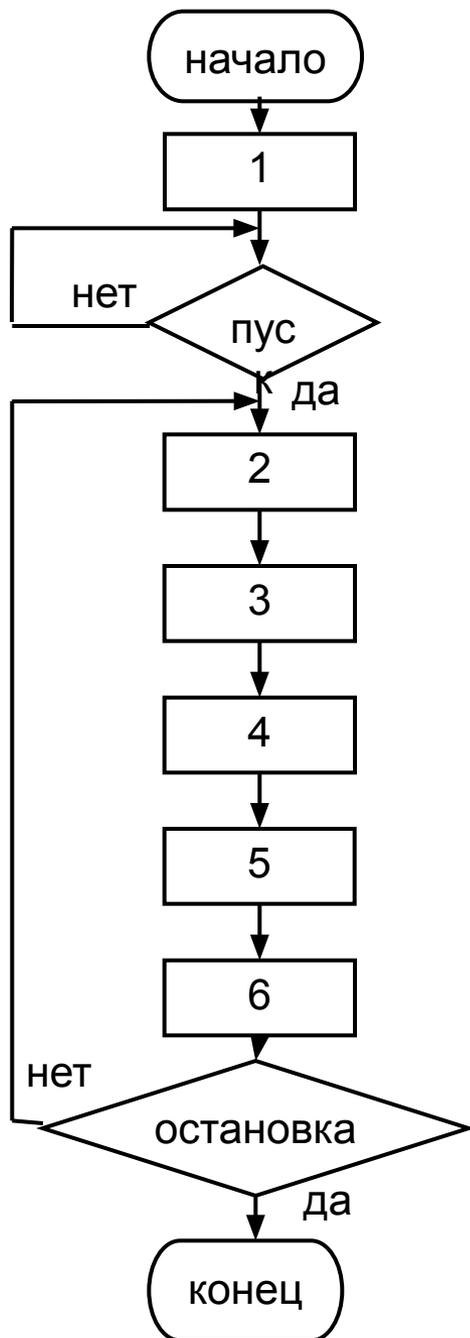
ЗУ – запоминающее устройство;

ЗУ состоит из памяти программ и памяти данных. Программы хранятся в постоянном ЗУ (ПЗУ), а данные – в оперативном ЗУ (ОЗУ).

Обобщённая структурная схема микроЭВМ



Алгоритм функционирования ЭВМ



Алгоритм отражает выполнение команд от момента включения до момента выключения.

1. Инициализация ЭВМ.

Здесь происходит загрузка программного счетчика. Как правило, в него заносится адрес 00. Далее проверяется аппаратный сигнал «Пуск».

2. Загрузка команды в регистр команд.

3. Дешифрация команды, формирование исполнительных адресов.

Исполнительный адрес – место нахождения операнда.

4. Выборка операнда.

5. Выполнение команды.

6. Формирование следующего адреса.

В процессе выполнения программы действия 2 – 6 повторяются, но с разными командами.

Процессор предназначен для обработки данных и управления режимами работы ЭВМ.

Память предназначена для хранения программы и данных.

Любая ЭВМ содержит узлы ввода и вывода:

Ввод предназначен для ввода данных.

Вывод предназначен для вывода результата.

Интерфейс – физическое и логическое согласование 2-х устройств вычислительной техники.

Системный интерфейс (СИ) – интерфейс, образующий вычислительную систему (машину).

СИ является связующим звеном узлов ЭВМ.

Память

Память состоит из ячеек. Каждая ячейка характеризуется разрядностью и имеет свой адрес.

Если память общая для программ и данных, то её называют принстонской, иначе – гарвардской.

Узлы ввода/вывода

Узлы ввода/вывода состоят из портов ввода/вывода. Каждый из портов имеет свой уникальный адрес.

Процессор (П)

УУ (устройство управления) предназначено для управления П и ЭВМ в целом. Исходной информацией для УУ является код команды, определяющий действие П и ЭВМ.

Команда выполняется по машинным тактам, из которых складываются машинные циклы. Время выполнения одного такта определяется частотой тактового генератора (ТГ). Минимальное время выполнения одной команды – один цикл. Минимальное время выполнения одного цикла – один такт.

ПС (программный счетчик) – регистр, служащий для временного хранения адреса команды.

АК (аккумулятор) – регистр, служащий для временного хранения данных.

АЛУ (арифметико-логическое устройство) служит для арифметической и логической обработки данных.

РК (регистр команд) предназначен для хранения кода исполняемой команды.

Все эти элементы связаны внутренней шиной. Внутренняя шина через буфер (Бф) связана с шиной данных (ШД). Бф служит для согласования внутренней шины и системного интерфейса (СИ).

Шина – совокупность проводников.

Есть шины данных, адреса (ША), управления (ШУ).

ШУ предназначена для передачи сигналов управления. Она показывает, что нужно делать с передаваемыми данными.

ША предназначена для передачи кода адреса от микропроцессора к устройствам, подключенным к ней.

ШД предназначена для обмена данными между различными устройствами. Шина данных обычно 8-разрядная.

Выполнение команды в микроЭВМ

Выполнения любой команды начинается с загрузки команды. Для загрузки команды процессор устанавливает на ША адрес команды. Это осуществляется посредством УУ и ПС. Из ячейки памяти код команды через ШД, Бф, внутреннюю шину поступает в РК. УУ дешифрует код команды и процессор осуществляет действия, соответствующие этой команде. Код команды оригинален для каждой последовательности действий данного процессора. За первой всегда должна следовать команда.

Команда сложения:

$A:=B+C$. Здесь имеются два операнда и один результат. В этой команде используются три адреса (откуда брать B, C и куда помещать A).

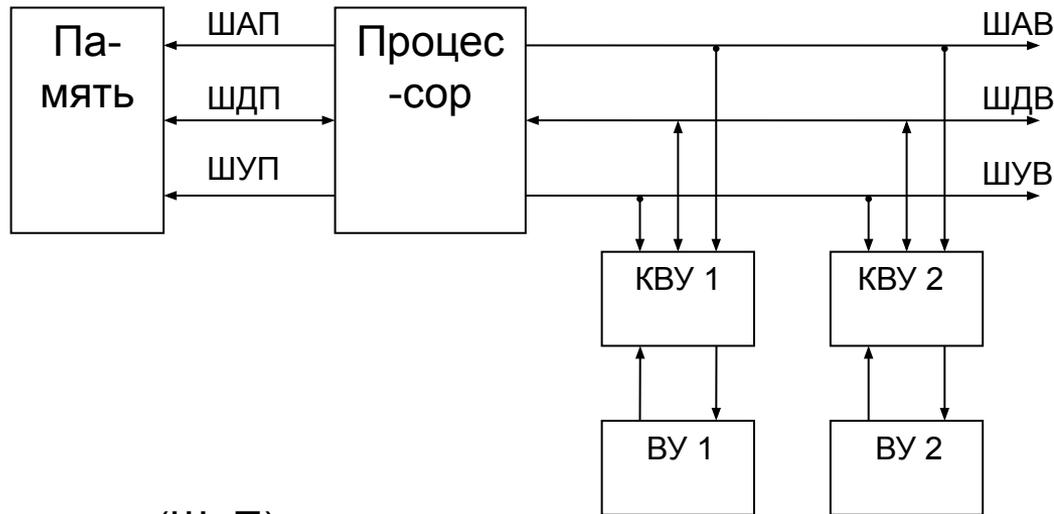
$A:=A+C$. Здесь число адресов уменьшается до двух. Это упрощает кодировку. Если заранее указано, где находится A, кодировка еще более упрощается. Результат отправляется в АК.

Код команды загружается в РК. Для проведения операции сложения первый операнд загружается из АК в АЛУ. Допустим, что второй операнд находится в ячейке памяти. Тогда процессор устанавливает адрес второго операнда и читает его из ячейки памяти. Операнд поступает в АЛУ, складывается, результат отправляется в АК.

На обращение к памяти всегда расходуется машинный цикл, на перенос операнда в АЛУ – машинный такт. Для экономии обращения к памяти в современных процессорах вводится блок РОН (регистр общего назначения). Доступ к РОН осуществляется за один такт. Существуют процессоры, где АК отсутствует, а все операнды хранятся в РОН. В таком процессоре $C:=C+A$ и $A:=A+C$ эквивалентны. Здесь имеют место большая гибкость и высокая производительность.

Структура микроЭВМ

1. Структура с отдельными шинами



Система шин памяти (Ш..П) связывает процессор и память.

Интерфейс ввода/вывода (Ш..В) предназначен для ввода/вывода данных. Контроллеры ввода/вывода (КВУ) служат для физического и логического согласования внешних устройств (ВУ) с внутренним интерфейсом. Между контроллером и ВУ действует интерфейс ВУ.

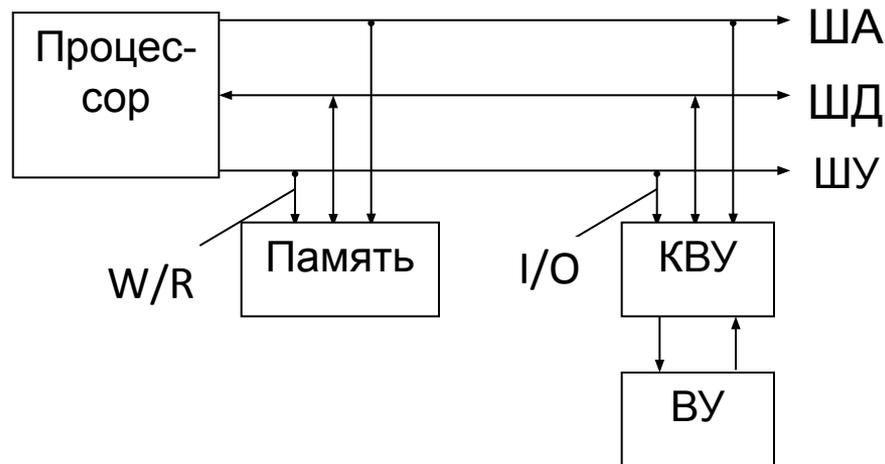
Достоинство: одновременно осуществляется обмен между памятью и процессором, между процессором и ВУ.

Недостаток: значительные аппаратные затраты.

Такая структура в ЭВМ не применяется.

2. Структура с изолированными шинами

При такой структуре память и контроллеры внешних устройств (КВУ) связаны с процессором одним системным интерфейсом (СИ).



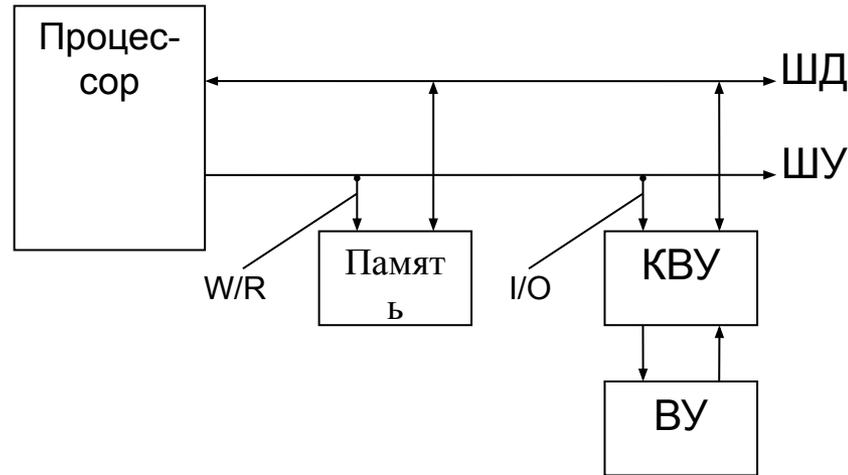
В такой структуре используются разные сигналы управления при обращении к памяти и внешнему устройству (ВУ), используются разные команды для работы с ними:

W/R – write/read.
I/O – input/ output.

Здесь имеются два изолированных адресных пространства (адреса памяти и адреса внешнего устройства), то есть адреса памяти и внешнего устройства могут пересекаться.

Достоинство: большое адресное пространство.

3. Структура с изолированными шинами и мультиплексированием линии адрес-данные.

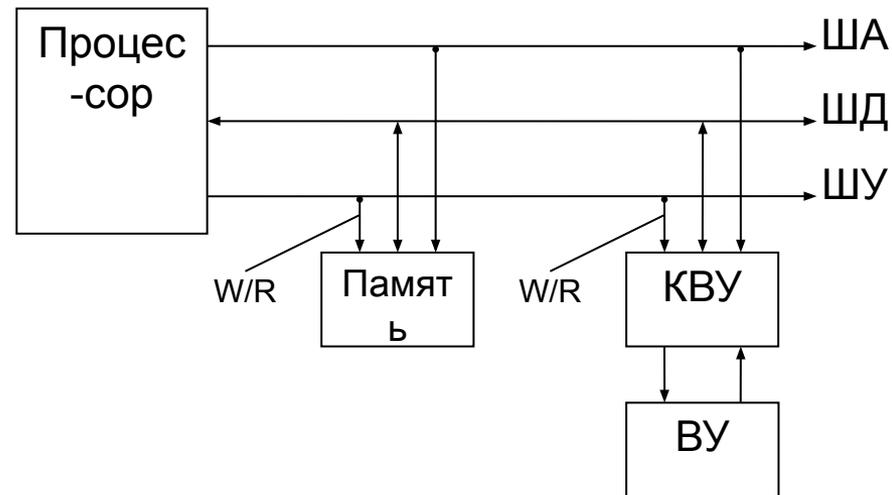


В такой структуре используются различные наборы команд для управления памятью и внешним устройством. Здесь имеются два адресных пространства. Для уменьшения линий СИ данные и адреса передаются по одной шине, причем сначала передается адрес, потом данные (это осуществляется за два такта).



Недостаток: низкое быстродействие.

4. Структура с общими шинами



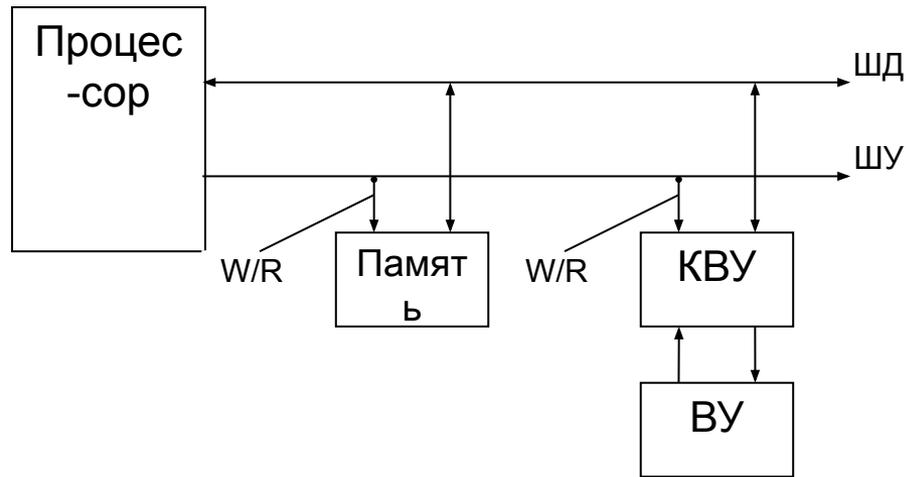
В такой структуре имеется единое адресное пространство для памяти и внешнего устройства, а также одни и те же команды для работы с памятью и внешним устройством, поскольку сигналы управления общие.

Достоинство: структура обеспечивает более гибкое программирование.

Недостаток: ограниченное адресное пространство (часть его занимает ВУ).

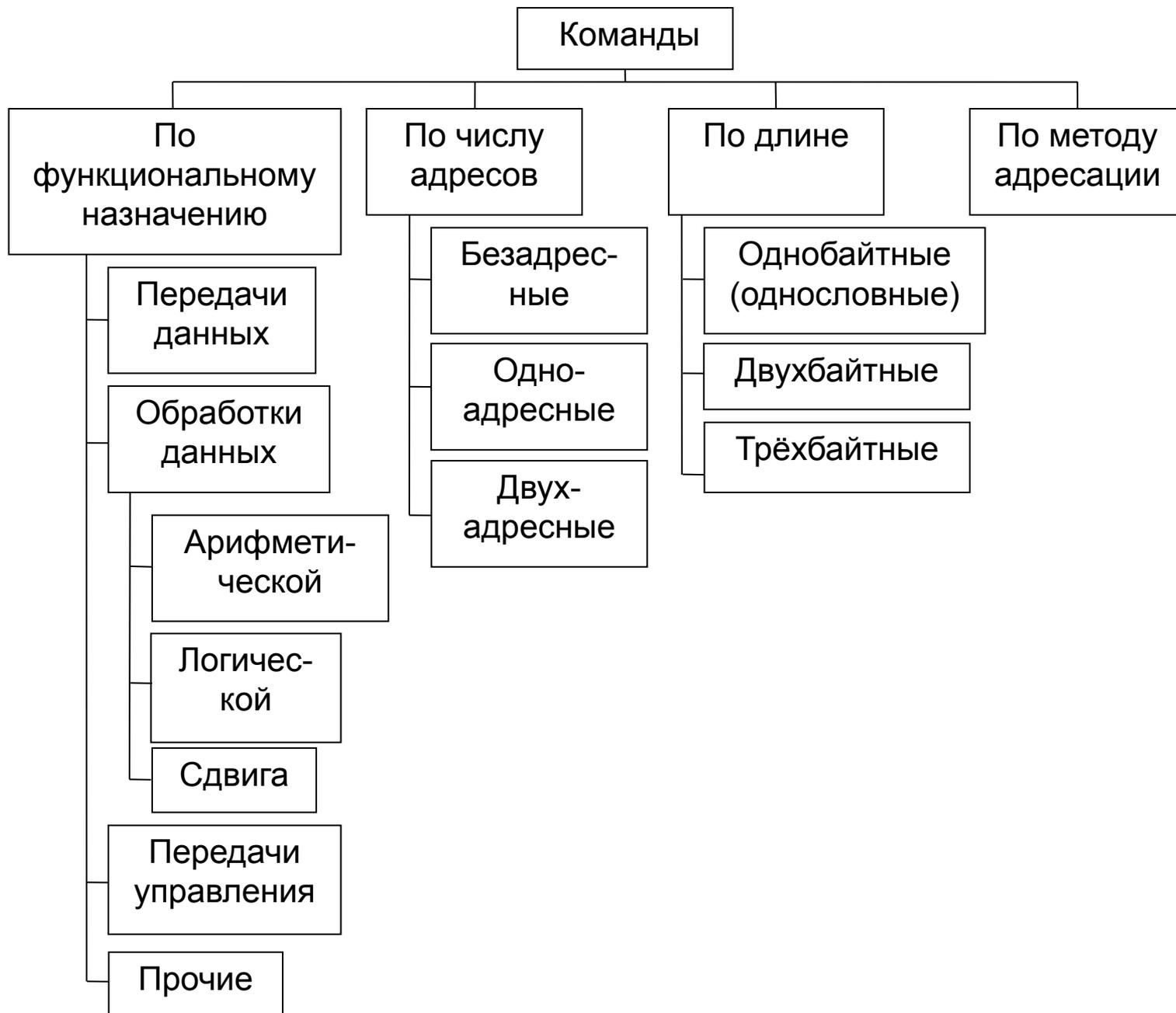
Такая структура применяется для простых ЭВМ.

5. Структура с общими шинами и мультиплексированием линии адрес-данные.



При такой структуре имеется общее адресное пространство. Здесь используется одна система команд для работы с ВУ и памятью. Данные и адреса передаются по одной шине. ЭВМ «электроникаб» выполнена по этой структуре.

Классификация команд ЭВМ



1. По функциональному назначению.

1.1. Команды передачи данных.

Эти команды составляют примерно 70% от числа всех команд.

Источниками информации являются память, регистры общего назначения, внешние устройства.

Примеры команд передачи данных:

MOV A,B – передать данные из B в A. Эта команда двухадресная, в ней имеется два операнда.

A:=B

A – приемник, B – источник.

LDA aa – загрузить A из источника с адресом aa.

STA aa – поместить A в память по адресу aa.

IN v – ввести данные с устройства v.

OUT v – вывести содержимое A по адресу v.

1.2. Команды обработки данных.

1.2.1. Команды арифметической обработки.

ADD B – команда сложения.

A:=A+B

SUB B – команда вычитания.

A:=A-B

В процессоре имеется специальный регистр, в котором хранятся признаки (флаги).

Содержимое этого регистра характеризуется словом состояния процессора PSW.

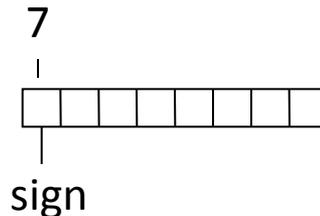
Для команд арифметической обработки данных используются следующие признаки слова состояния:

1) C (carry) – признак переноса, устанавливаемый в случае переноса 1 из старшего разряда регистра A (или в случае заема) при выполнении некоторых операций над данными. При отсутствии переноса в выполняющейся операции признак переноса сбрасывается.

ADC – команда сложения с переносом. Здесь содержимое признака C добавляется к результату сложения.

SBB – вычитание с заемом. Здесь содержимое признака C вычитается из результата вычитания.

2) S (sign) – признак знака, принимающий значение 1, если содержимое регистра A становится отрицательным после выполнения некоторой операции. Признак знака устанавливается по содержимому старшего разряда, так как старший разряд используется под знак. Если в старшем разряде 1, то $S=1$, если в старшем разряде 0, то $S=0$.sign7



3) Z (zero)– признак нулевого результата, принимающий значение 1, если число в регистре A равно 0; если же число в регистре A не равно 0, то признак Z сбрасывается.

1.2.2. Команды логической обработки данных.

ANA B – содержимое регистра B подвергается логическому И с содержимым аккумулятора.

XRA B – ИСКЛЮЧАЮЩЕЕ ИЛИ выполняется с содержимым регистра B и аккумулятора.

CMR B – команда сравнения. Содержимое регистра B вычитается из содержимого аккумулятора, но содержимое аккумулятора не изменяется, а по результату устанавливаются признаки.

1.2.3. Команды сдвига.

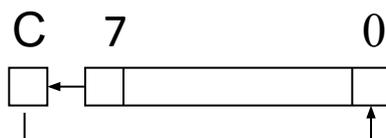
Различают следующие сдвиги:

правый и левый;

арифметический и циклический;

с переносом и без него.

Рассмотрим циклический сдвиг с переносом влево.



Содержимое старшего разряда переходит в содержимое переноса, а содержимое переноса – в содержимое младшего разряда.

1.3. Команды передачи управления.

Команды передачи управления бывают условные и безусловные.

Условные команды используют признаки слова состояния процессора, например:

JMP aa – перейти по адресу aa (безусловная);

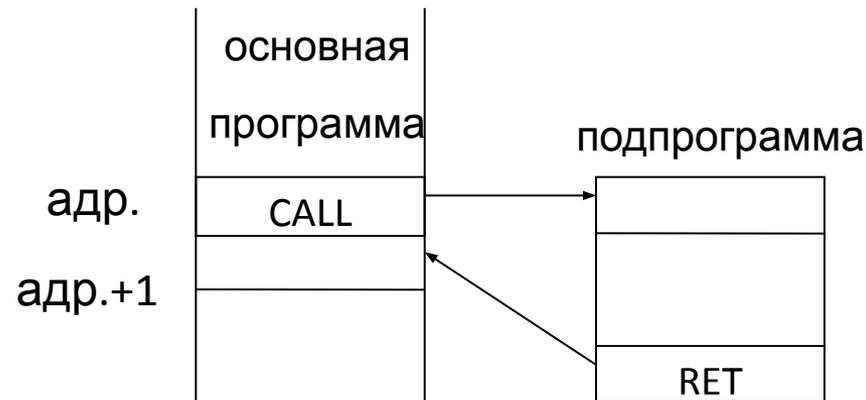
JZ aa – перейти по адресу aa, если ноль (условная).

Команды передачи управления предназначены для реализации разветвляющихся и циклических вычислительных процессов (алгоритмов).

К командам передачи управления относятся команды перехода к подпрограмме и возврата в программу.

Существует условный переход к подпрограмме и условный возврат из подпрограммы.

CALL aa – вызвать подпрограмму по адресу aa (команда безусловной передачи управления).



Для сохранения адреса возврата служит стек. Стек – это область оперативной памяти, доступ к которой организован по принципу «первым зашел – последним вышел».

Различают аппаратный и программный стек. Аппаратный стек реализован через указатель стека (SP). SP – специальный регистр, в котором храниться текущий адрес стека (вершина стека). При обращении к подпрограмме адрес возврата сохраняется в стеке, а при выходе из подпрограммы восстанавливается из него.

1.4. Прочие команды.

К этим командам относятся NOP и т.д.

NOP – нет операций.

2. По числу адресов.

2.0. Безадресные команды.

В таких командах отсутствуют указания на адрес операнда, например:

HLT – команда ОСТАНОВ.

CLR C – очистка признака C.



2.1. Одноадресные команды.

Здесь имеется указание на один операнд, например:

INR B – инкрементировать B (увеличить на 1).

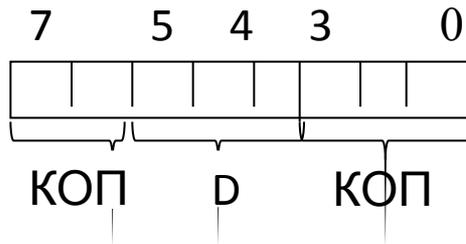
$V := V + 1$

B D закодирован регистр B.

2.2. Двухадресные команды.

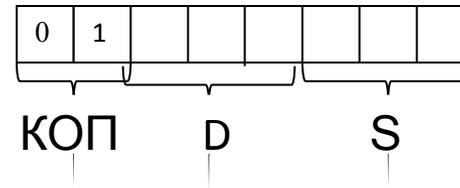
К двухадресным командам относится команда MOV B,C (передать данные из C в B).

$V := C$



$D(B) := S(C)$

B D закодирован приемник, в S – источник.



3. По длине.

3.1 а) Однобайтные команды (однословные).

Минимальная длина команды всегда равна размеру разрядной сетки процессора (1 байт).

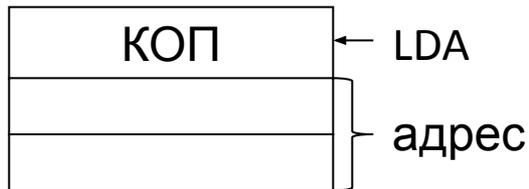
б) Двухбайтные команды.

К ним относится команда `MVI B, const` (передать непосредственно следующие данные в B). Здесь используется непосредственный метод адресации.



в) Трехбайтные команды.

К этим командам относится команда `LDA, aa` (загрузить A из источника с адресом aa). К трехбайтным командам относятся все команды передачи управления.



4. По методу адресации.

Методы адресации различны для разных процессоров, однако всех их можно разделить на:

методы прямой адресации (в команде указан адрес операнда);

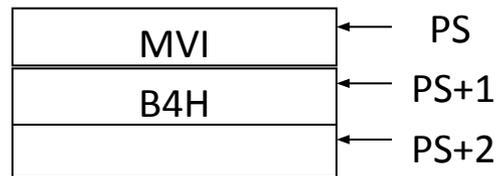
методы косвенной адресации (в команде указан адрес адреса операнда).

В качестве примера прямой адресации можно указать регистровую адресацию

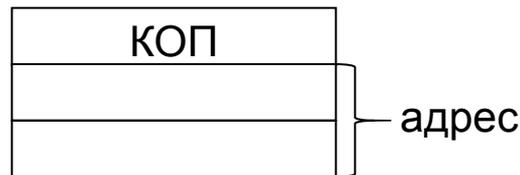
MOV A,B где A-приемник, B-источник. Команда **MOV M,A** отображает метод

косвенной адресации. В данном случае адрес ячейки M указан в регистре HL.

В ряде случаев адресация осуществляется через программный счетчик. К этому типу адресации можно отнести непосредственную адресацию **MVI B, #B4H**.



В данном примере константа B4 записывается в регистр B. Сначала программный счетчик PS указывает на код команды. Затем его содержимое увеличивается на 1. Так как константа находится непосредственно в команде, то адресация называется непосредственной. В команде сохранения и загрузки аккумулятора фигурирует так называемая абсолютная адресация.



В процессе считывания участвует программный счетчик PS, а так как в команде абсолютно указан адрес операнда, то адресация называется абсолютной.

Контроллеры внешних устройств.

Контроллер внешнего устройства (КВУ) – это узел ЭВМ, осуществляющий логическое и физическое согласование СИ и интерфейса ВУ.

Классификация интерфейсов внешних устройств.

По способу передачи данных интерфейсы ВУ делятся на:
 последовательные;
 параллельные.

Согласно этому можно подразделить контроллеры на:

последовательно ввода/вывода;

параллельного ввода/вывода.

При параллельном интерфейсе одновременно в линии связи действует байт информации.

При последовательном - информация передается побитно. В каждый момент времени действует 1 бит. В этом случае существенно упрощаются линии связи между внешним устройством ВУ и его контроллером КУ. Все современные сети построены на основе последовательного интерфейса.

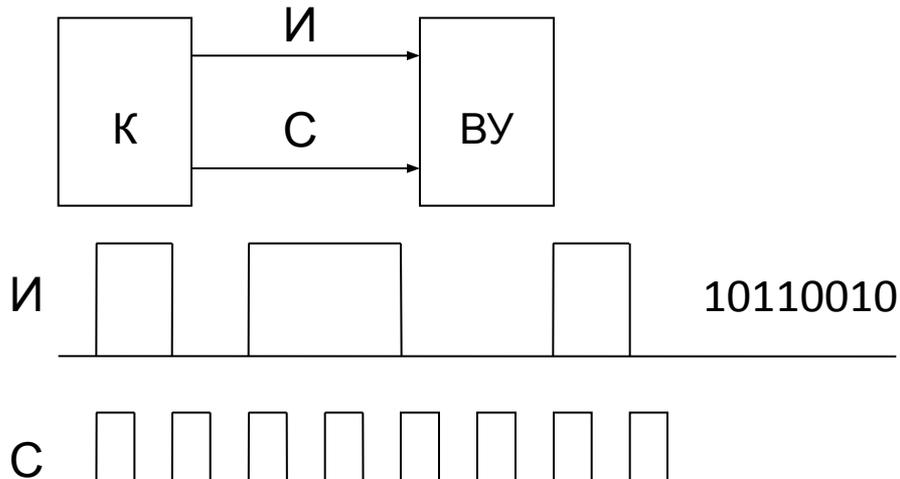
Интерфейсы последовательного обмена делятся на:

синхронные;

асинхронные.

Синхронный интерфейс.

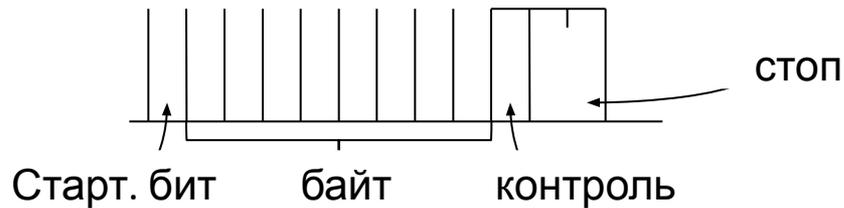
Имеются две линии, связывающие контроллер с ВУ: информационная и линия синхронизации.



За один период синхронизирующего сигнала передается 1 бит информации, что крайне неудобно (необходимо наличие двух линий).

Асинхронный интерфейс.

При асинхронном обмене между контроллером и внешним устройством имеется всего одна линия связи.



Байт информации обрамляется дополнительными битами и вместе с ними образует посылку. Стартовый бит показывает, что началась передача данных. После стартового бита следует байт информации, после которого идет завершение обмена, осуществляемое одним битом контроля и одним/двумя стоповыми битами. Бит контроля предназначен для контроля верности передачи, который осуществляется либо по четности, либо по нечетности. Контроль по четности заключается в том, что число единиц в байте информации и бите контроля четное. Контроль по нечетности противоположен контролю по четности. Обмен между контроллером и интерфейсом тоже может быть синхронным или асинхронным, но всегда параллельным.

По способу размещения контроллеры делятся на:

А) Контроллеры, выполненные на одной плате с процессором;

Недостатком такого метода является то, что при отсутствии контроллера на какое-либо устройство, его нельзя будет установить.



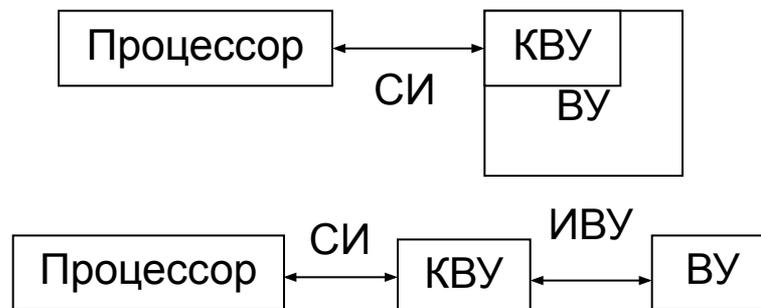
Б) Контроллер и ВУ – единый конструктив.

Достоинство: можно подключить любое ВУ к процессору.

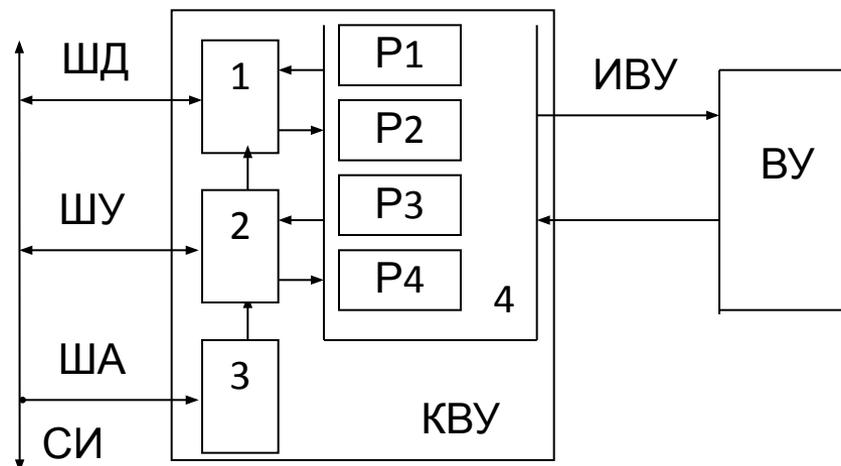
Недостаток: длина проводов должна быть минимальной, поэтому практически не применяется.

В) Контроллер внешнего устройства является отдельным конструктивным элементом.

Достоинство: гибкость построения вычислительной машины.



Типовая структурная схема КВУ.



КВУ включает в себя следующие основные узлы:

приемопередатчики шины данных ШД;

приемопередатчики шины управления ШУ, логика управления;

приемники шины адреса ША, дешифратор адресов;

основной узел контроллера.

Узел 4 решает вопросы синхронизации ВУ с КВУ, преобразования и усиления сигналов контроллера. В его состав входят следующие регистры:

P1 - регистр данных вывода (временное хранение данных);

P2 - регистр данных ввода (временное хранение данных);

P3-регистр состояния (отражает состояние контроллера и ВУ);

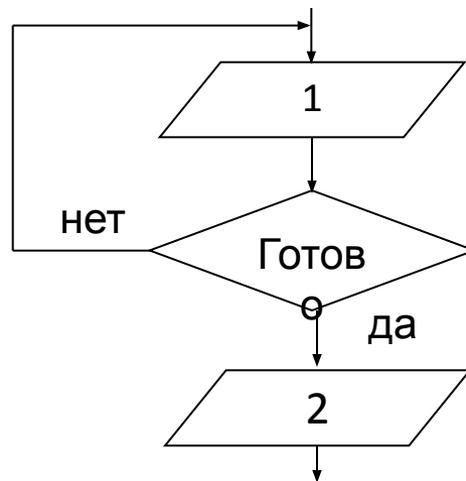
P4-регистр команд.

Регистры P3 и P4 часто объединяют в регистр команд состояния (PKC).

Алгоритм асинхронного обмена СИ с контроллером.

Обмен данными может осуществляться в 2-ух режимах: синхронном (данные вводятся/выводятся и нет информации об их истинности), асинхронный (по алгоритму опроса готовности).

Алгоритм опроса готовности выглядит следующим образом:

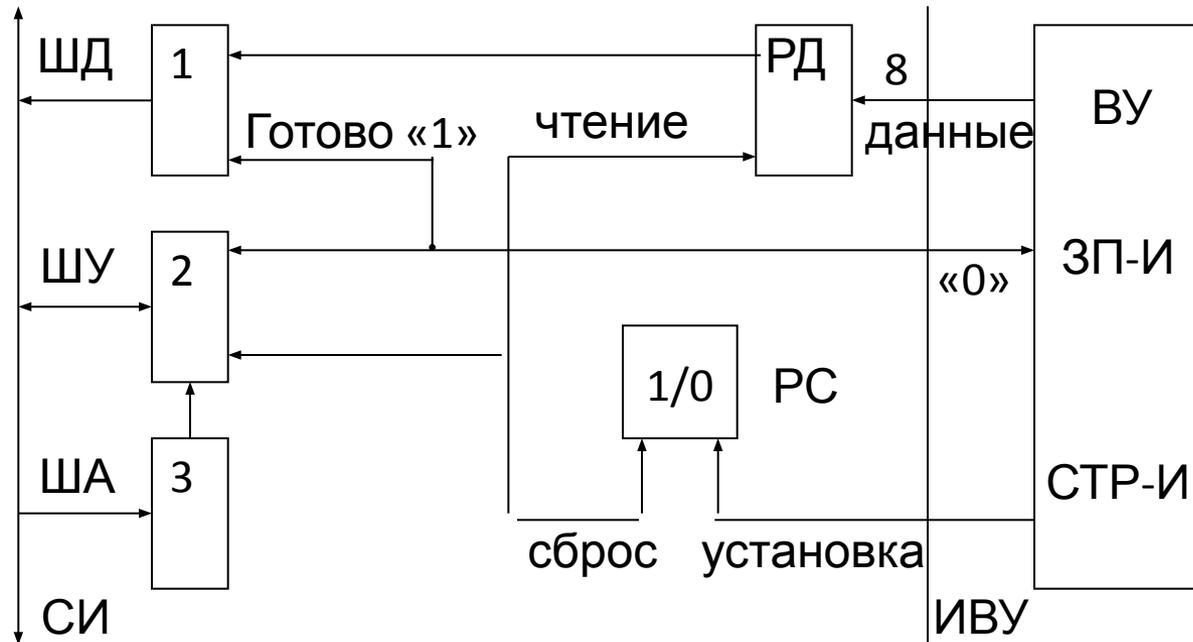


1 – чтение регистра состояния;

2 – ввод/вывод данных в контроллер.

Контроллер параллельного ввода.

Контроллер предназначен для ввода байта данных от ВУ в СИ. При этом решается задача синхронизации СИ с контроллером и контроллера с ВУ.



ИВУ-интерфейс внешнего устройства.

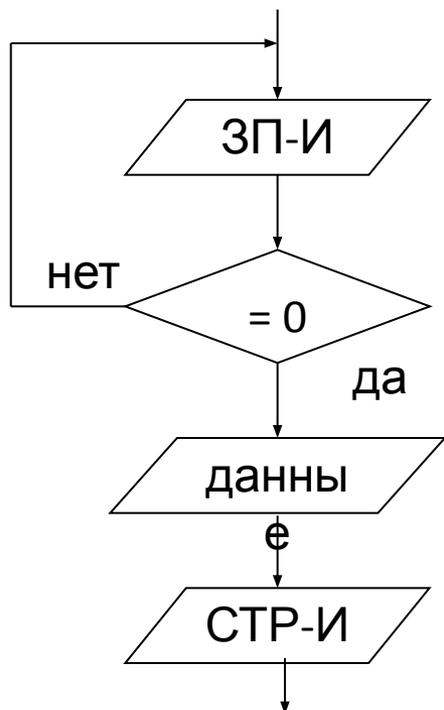
Для синхронизации контроллера и ВУ ИВУ имеет два синхронизирующих сигнала – запрос источнику (ЗП-И) и строб источника (СТР-И).

Стробирующий сигнал показывает, что данные установлены и истинны.

Этот сигнал сопровождает байт данных.

Синхронизация ВУ и контроллера.

Для решения этой задачи ВУ должно функционировать по следующему алгоритму:



ВУ анализирует сигнал ЗП-И. Если этот сигнал не равен 0, то устройство ждет, когда он станет нулевым, т.е. активным. В случае, если ЗП-И активного уровня, контроллер готов принять данные. ВУ устанавливает данные и сопровождает их сигналом СТР-И.

Синхронизация СИ с контроллером.

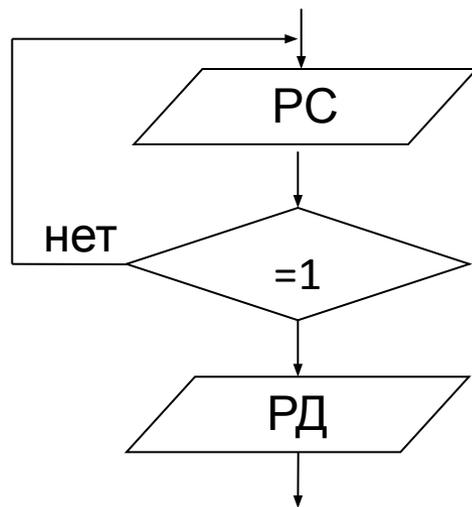
Синхронизация производится по алгоритму опроса готовности (см. выше). Внутренняя часть контроллера параллельного ввода (см. рис) связана с СИ посредством 3 узлов:

- 1 – передатчики ШД;
- 2 – приемо-передатчики ШУ, логика управления;
- 3 – приемники ША, дешифратор адресов.

РД – регистр данных для временного хранения вводимых данных.

РС – одноразрядный регистр состояния контроллера и ВУ.

Алгоритм синхронизации СИ и контроллера выглядит так:



В контроллере происходит совместное выполнение двух алгоритмов: указанного выше и опроса готовности. ВУ, получив активный сигнал ЗП-И, устанавливает данные и сопровождает их сигналом СТР-И. Сигнал СТР-И устанавливает регистр состояния. Процессор, прочитав готовность контроллера (РС установлен), вводит данные из регистра данных РД. При этом внутренний сигнал ЧТЕНИЕ открывает РД и сбрасывает РС. Контроллер снова готов принять следующий байт данных. Переключение РС обеспечивает синхронизацию СИ с контроллером и контроллера с ВУ.

Контроллер параллельного вывода.

Этот контроллер предназначен для вывода байта данных из СИ во ВУ.

При этом решается задача синхронизации СИ с контроллером и контроллера с ВУ.

Эта задача решается путем опроса готовности контроллера.

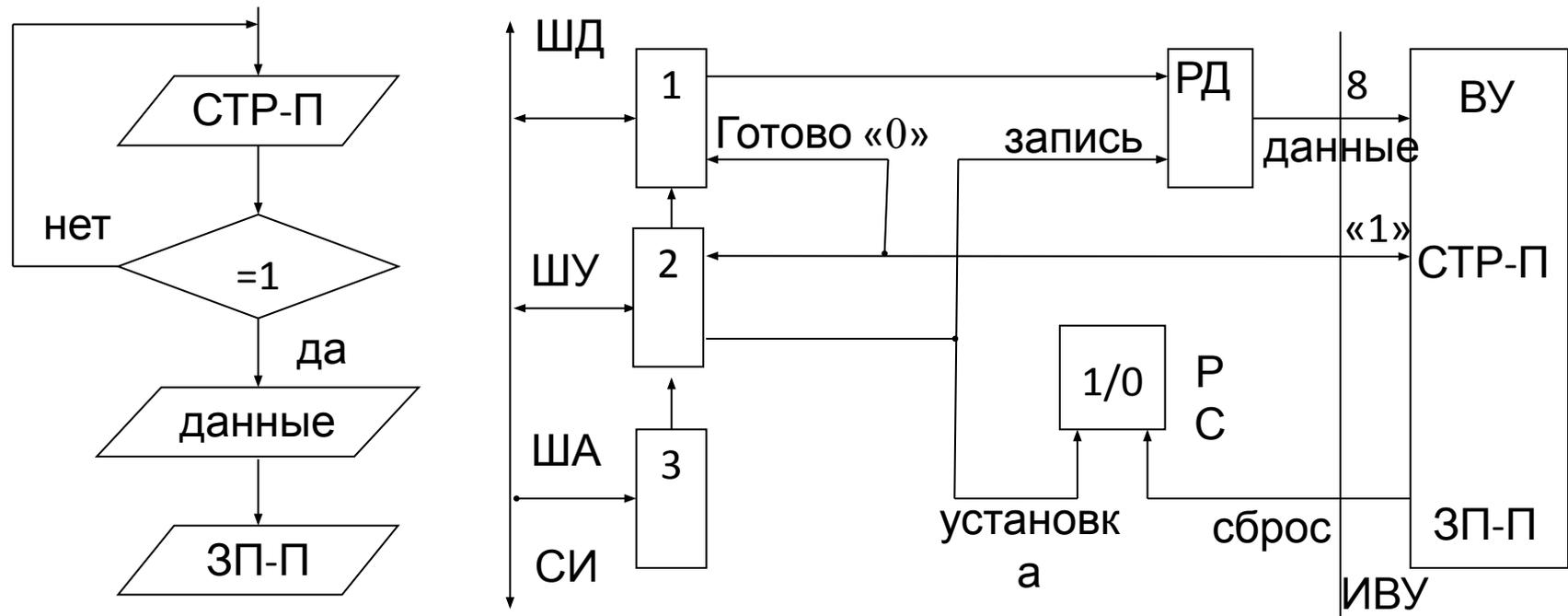
На рис. цифрами указаны следующие блоки:

1 – приемо-передатчики ШД;

2 – приемо-передатчики ШУ, логика управления;

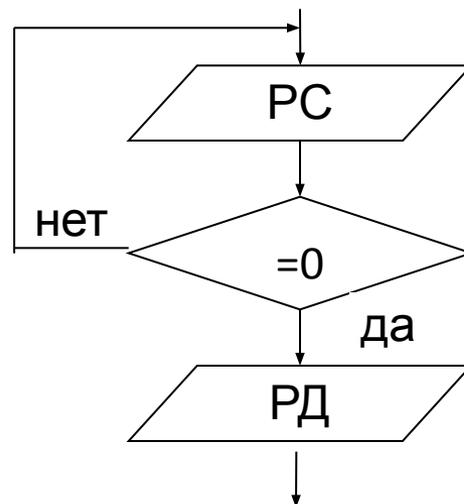
3 – приемники ША, дешифратор адресов.

Как и в контроллере параллельного ввода в контроллере параллельного вывода имеются два регистра: регистр данных РД и регистр состояния РС.



Алгоритм синхронизации ВУ и контроллера представлен на следующем рисунке

Синхронизация СИ и контроллера производится по следующему алгоритму



Между СИ и контроллером осуществляется асинхронный обмен, т.е. процессор опрашивает РС (готовность контроллера). Если контроллер готов, т.е. в РС находится «0», то производится вывод данных в РД.

Со стороны ВУ в ИВУ действуют следующие сигналы:

8 разрядов данных;

2 синхронных сигнала СТР-П (строб приемнику – обозначает истинность передаваемых данных);

ЗП-П (запрос от приемника указывает на готовность приемника принять информацию).

Контроллер установил данные и сопроводил их сигналом СТР-П; при этом ВУ получило данные и установило сигнал ЗП-П. Сигнал ЗП-П сбрасывает РС. Когда сбрасывается СТР-П повторный ввод данных ВУ невозможен. Одновременно устанавливается готовность контроллера (ГОТОВО «0»).

Процессор, убедившись в готовности контроллера выводит очередной байт данных. При этом внутренний сигнал ЗАПИСЬ записывает данные в РД и устанавливает РС, тем самым сбрасывается готовность контроллера, повторный вывод данных невозможен, и устанавливается сигнал СТР-П. ВУ может забрать следующий байт данных.

Во всех случаях переключение РС не дает возможности передавать данные с большой скоростью.

Интерфейс ИРПР.

ИРПР – интерфейс радиальный параллельный. Интерфейс ИРПР обеспечивает параллельный байтовый обмен данных между контроллером и ВУ. Максимальная длина 10м. Используется для подключения принтеров, терминалов и др.



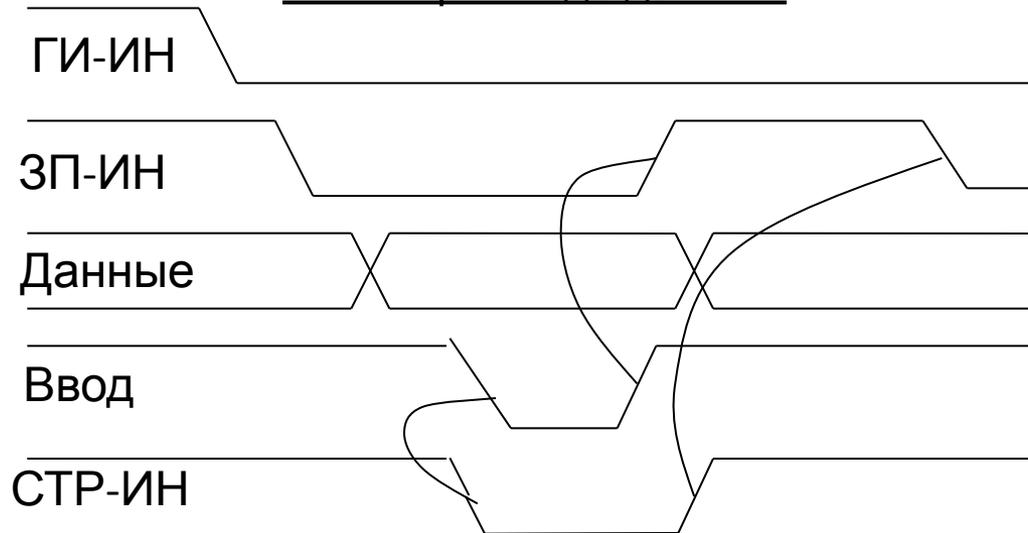
Описываемый интерфейс содержит 3 синхросигнала:

ГИ-ИН – готовность источника. Активный уровень – низкий. Сигнал свидетельствует о включенном и исправном состоянии ВУ;

ЗП-ИН – запрос источнику. Этот сигнал указывает на готовность контроллера принять данные от ВУ;

СТР-ИН – строб источника сопровождает передаваемые данные.

ИРПР при вводе данных.



Контроллер устанавливает сигнал запроса, в ответ ВУ выдает байт данных и сопровождает строб-сигналом. Получив строб-сигнал, контроллер получает данные и сбрасывает сигнал ЗП-ИН. После завершения ввода данных вновь устанавливает сигнал запроса.

Стрелками на временной диаграмме показано взаимодействие синхросигналов. В регистре состояния минимум два разряда. Разряд ОШИБКА – старший.

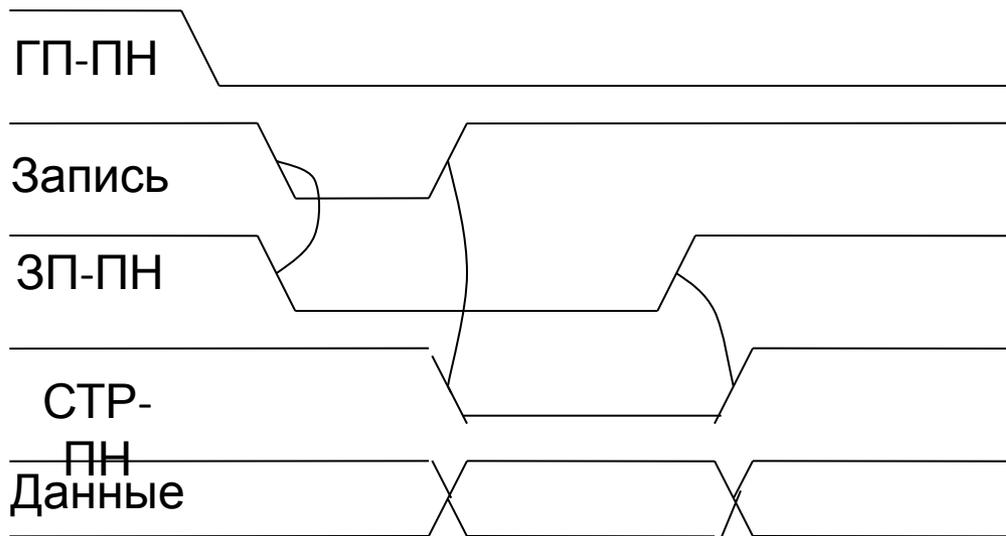


Если ГИ-ИН находится в активном состоянии, то в старшем разряде записана «1».

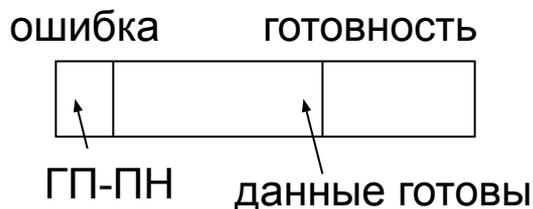
ИРПР при выводе данных.



ГП-ПН – сигнал готовности приемника от приемника, указывает на включенное и исправное состояние ВУ. Активный – низкий уровень. ЗП-ПН – запрос приемника. СТР-ПН – строб приемнику.



Регистр состояния будет выглядеть следующим образом:



Последовательный ввод-вывод.

Последовательный обмен может быть синхронным и асинхронным.

Синхронный обмен. Контроллер последовательного синхронного приёма.

Контроллер ПСП решает задачу преобразования последовательного байта данных в параллельный код и ввод этого кода в системный интерфейс.

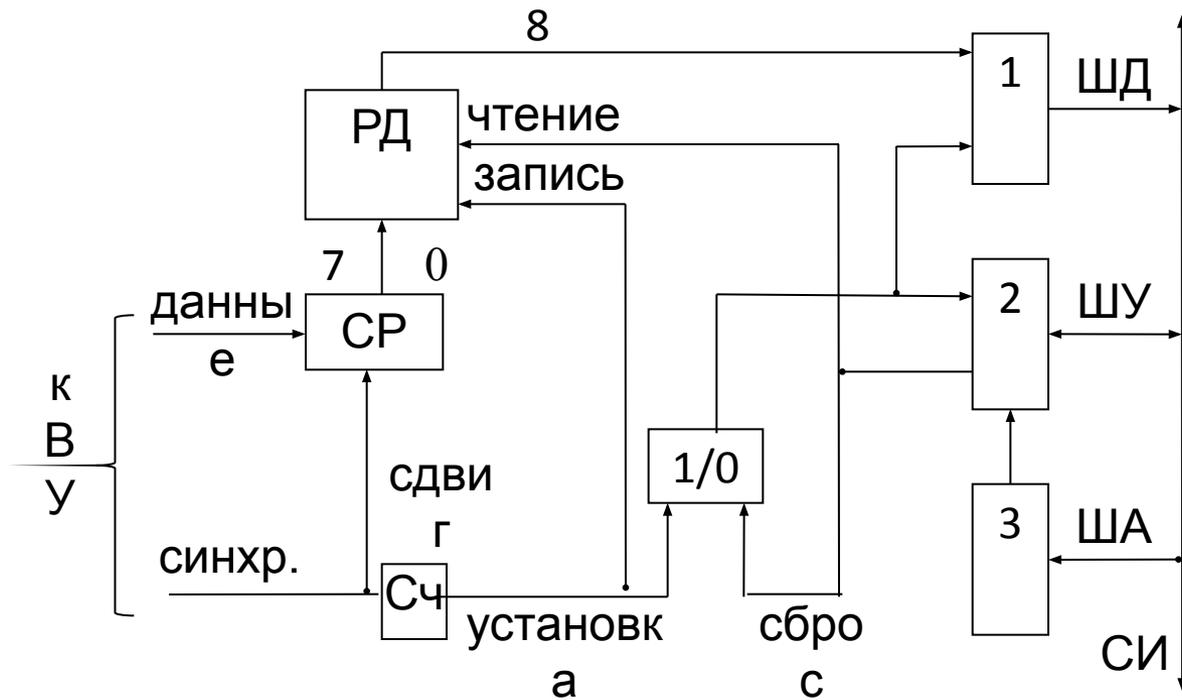
Контроллер включает в себя:

- 1 – приемо-передатчики ШД;
- 2 – приемо-передатчики ШУ, логика управления;
- 3 – приемники ША.

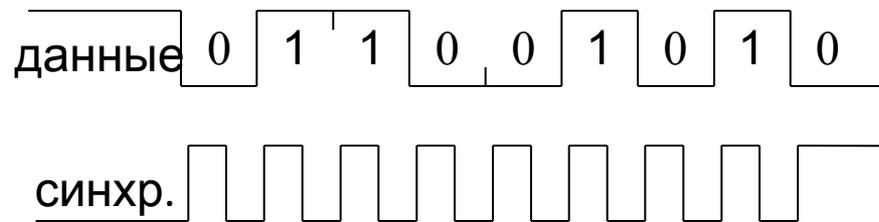
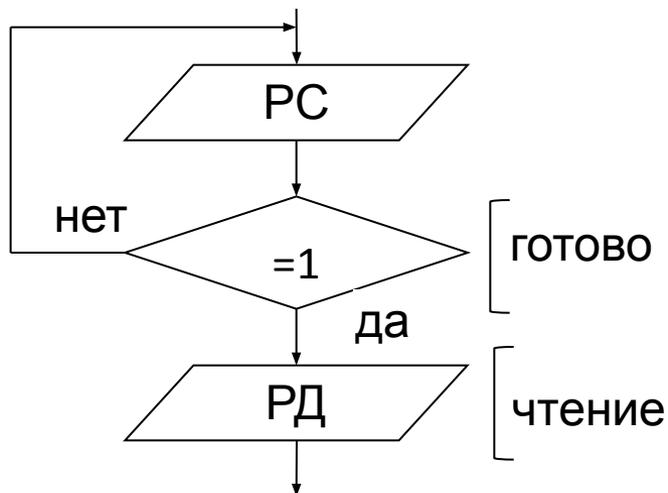
СР – 8-разрядный сдвиговый регистр для преобразования последовательного кода в параллельный.

Сч – счетчик – делитель на 8, для подсчета принятых бит.

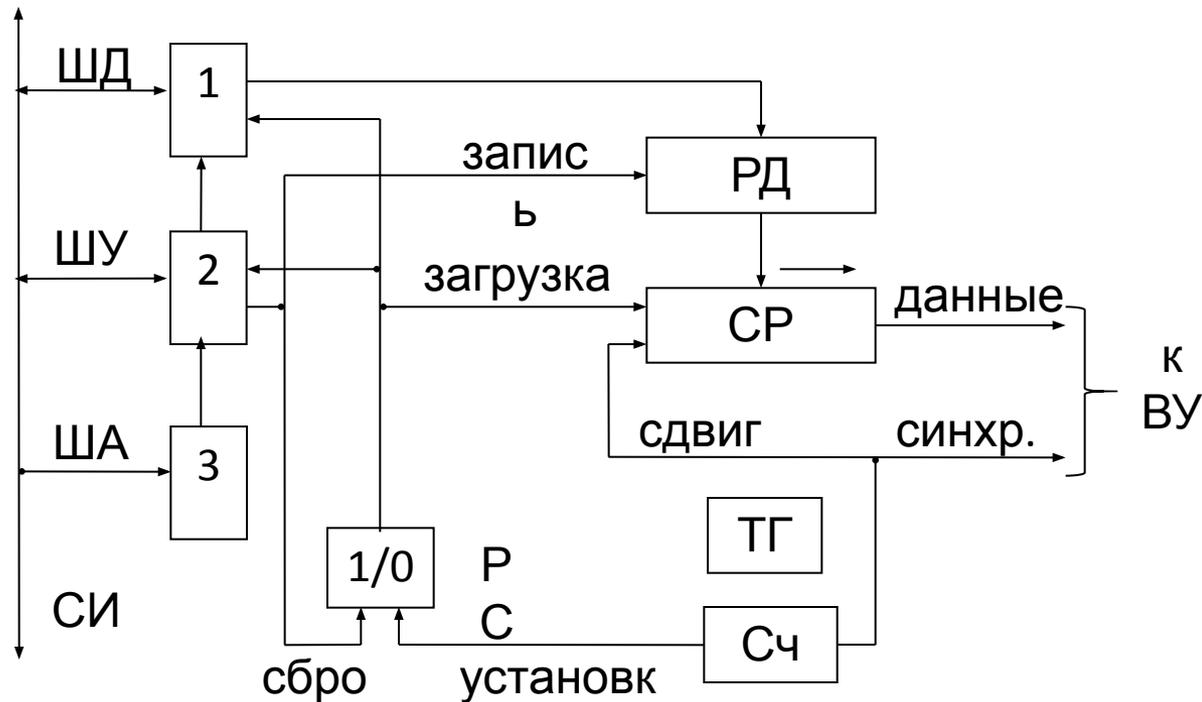
На вход данных сдвигового регистра поступает информация, на вход СДВИГ – синхроимпульсы. При этом за 1 период синхросигнала будет принят и сдвинут 1 бит информации. Поскольку мы принимаем 1 байт данных, то таких сдвигов должно быть 8, поэтому в схеме имеется счетчик-делитель на 8. При завершении 8-и сдвигов на выходе счетчика формируется внутренний сигнал ЗАПИСЬ, который переписывает принятый байт из СР в РД. Одновременно устанавливается регистр состояния РС и возникает состояние готовности контроллера.



Алгоритм синхронизации СИ и контроллера



Контроллер последовательной синхронной передачи.



Этот контроллер предназначен для синхронной передачи данных в линию, связывающую контроллер с ВУ. Назначение контроллера состоит в синхронизации обмена СИ с контроллером и ВУ, причем контроллер с СИ обмениваются данными асинхронно, а с ВУ – синхронно, по двум линиям (по линии синхронизации и линии данных).

Контроллер содержит три узла, связывающих СИ, а именно:

- приемо-передатчики ШД;
- приемо-передатчики ШУ, логика управления;
- приемники ША, дешифратор адресов.

Также в контроллере имеются восьмиразрядный регистр данных (РД) и одnorазрядный регистр состояния (РС). Восьмиразрядный сдвиговый регистр (СР) служит для преобразования параллельного кода в последовательный. Этот последовательный код будет побитно передан в линию связи. В схеме содержится тактовый генератор (ТГ). Его частота определяет скорость обмена. В данном случае особые требования к стабильности ТГ не предъявляются, так как синхронизирующие импульсы передаются вместе с данными при последовательном синхронном обмене. Для подсчета числа переданных бит имеется счетчик-делитель на 8.

Рассмотрим работу контроллера. Синхронизирующие импульсы ТГ поступают на вход «сдвиг» сдвигового регистра (СР), при этом за один период тактовой частоты будет передан в линию один бит данных. Одновременно счетчик (Сч) считает число переданных бит. После восьми Операций сдвига в линию будет передан байт данных. На выходе счетчика–делителя на 8 формируется сигнал «установка», который установит РС, тем самым будет произведена загрузка в СР следующего байта данных, а процессор, получив информацию о готовности контроллера к приему следующего байта данных, запишет его в РД, при этом РС будет сброшен.

Алгоритм работы процессора с контроллером, то есть синхронизации СИ и контроллера полностью идентичен предыдущим вариантам контроллеров (а именно контроллеру параллельного синхронного вывода).

Бит контроля четности в данном случае не формируется.

Данный контроллер содержит следующие узлы связи и синхронизации контроллера с СИ: приемо-передатчики ШД;

приемо-передатчики ШУ, логика управления;

приемники ША, дешифратор адресов.

В контроллере имеются РД, РС, а также дополнительные элементы, связанные с формированием посылки: одиннадцатиразрядные СР (так как в посылке 11 бит), высокостабильный ТГ, частота которого стабилизирована кварцевым резонатором.

Для уменьшения фазового сдвига между передатчиком и приемником применяется счетчик-делитель на 16. Счетчик-делитель на 10 служит для подсчета числа бит в посылке. Схема И формирует внутренний сигнал «загрузка».

Схема работает следующим образом: Частота импульсов ТГ делится на 16 и подается на вход «сдвиг» сдвигового регистра СР, при этом за каждый период этой частоты будет осуществлен сдвиг на 1 бит и будет передан 1 бит в линию данных. Сначала в линию передается стартовый бит, потом 8 информационных бит, затем 2 стоповых. После передачи десятого бита счетчик - делитель на 10 формирует сигнал «установка» для РС.

РС устанавливается, тем самым блокируется счетчик- делитель на 16, тем самым дальнейшая передача в линию данных прекращается (будет установлен одиннадцатый стоповый бит на линии данных). Одновременно с этим подготавливается схема И.

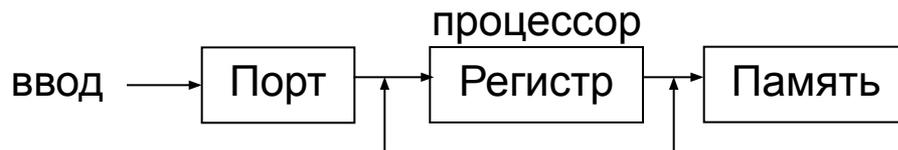
После того, как процессор определит состояние готовности контроллера, производится запись нового байта в РД, при этом сбрасывается РС, а также срабатывает схема И, формируется сигнал «загрузка» и новый байт данных записывается в СР совместно со стартовым и двумя стоповыми битами, формируя посылку. Поскольку РС сбрасывается, то счетчик-делитель на 16 разблокируется и через каждые 16 импульсов тактового генератора (ТГ) будет осуществляться сдвиг.

Узлы 1, 2, 3, РС, РД аналогичны соответствующим узлам предыдущих контроллеров. Первым на вход контроллера стартовый бит. Этот бит с помощью схемы И разблокирует Сч16. Счетчик делит частоту тактового генератора на 16 и каждый 16-й импульс тактового генератора сдвигает один бит сдвиговым регистром. После 9-ти операций сдвига стартовый бит будет сдвинут из СР, на вход СР поступит стоповый бит. Выход Сч9 с помощью схемы И блокирует Сч16. Дальнейший прием бит прекратится. При этом РС будет установлен и полученный байт переписан в РД. После чтения полученного байта процессором РС сбрасывается и схема принимает следующий байт.

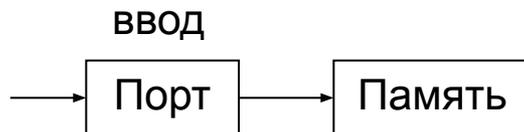
Режим прямого доступа к памяти.

Режим прямого доступа к памяти применяется для ввода и вывода больших объемов информации, например, все программы в персональный компьютер загружаются в режиме прямого доступа к памяти. В режиме прямого доступа контроллер работает с памятью непосредственно, минуя процессор.

Ввод одного байта информации (без применения режима прямого доступа к памяти) осуществляется следующим образом: информация из порта ввода загружается в регистр процессора, далее она переносится в память.

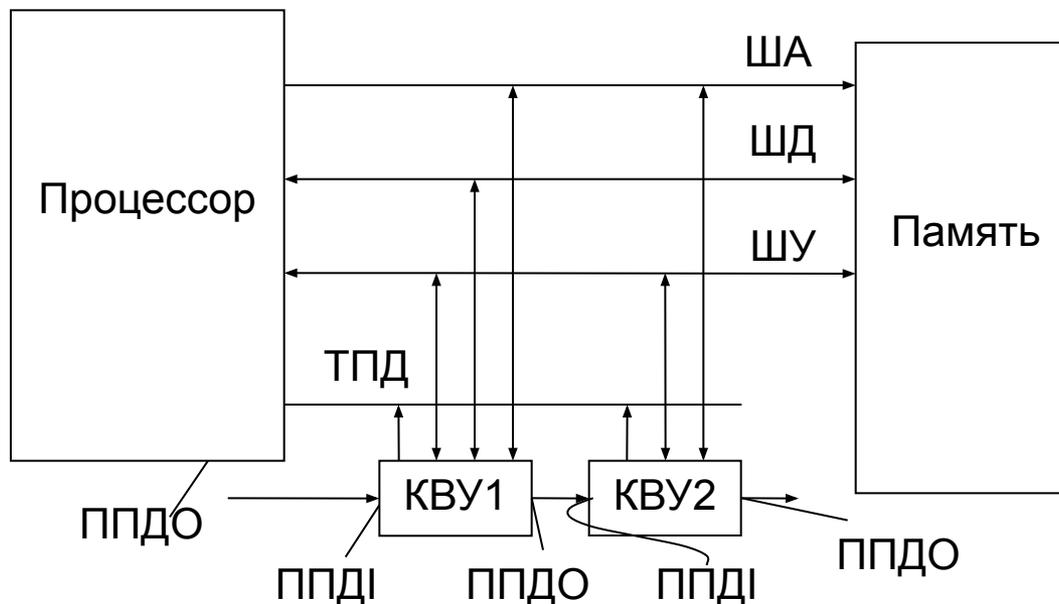


Здесь участвуют две команды, а это означает, что для загрузки информации необходимо как минимум 5 циклов. В этом случае при запоминании 1-го байта необходимо как минимум 5 циклов. В режиме прямого доступа к памяти порт напрямую связывается с памятью. Здесь нет необходимости читать команду.



Таким образом, режим прямого доступа является эффективным средством передачи данных. В этом и состоит основное достоинство режима прямого доступа к памяти. Недостатком такого режима являются дополнительные аппаратные затраты. Контроллер ВУ должен выполнять управление СИ.

Схема режима прямого доступа к памяти.

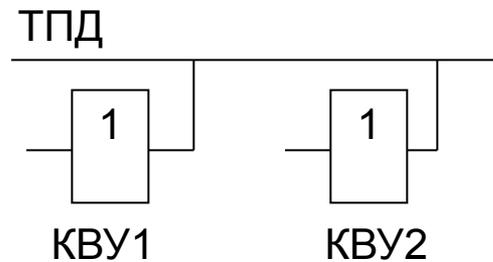


В схеме имеется два контроллера, которые могут работать в режиме прямого доступа (КВУ1, КВУ2). Кроме этого здесь показаны ША, ШД, ШУ и два дополнительных сигнала, которые обеспечивают логику работы в режиме прямого доступа: ТПД (требование прямого доступа), ППД (подтверждение прямого доступа).

ТПД – DMR;

ППД – DMC;

Для реализации прямого доступа контроллер запрашивает прямой доступ путем установки сигнала ТПД. Активный уровень сигнала – низкий, так как все контроллеры к линии сигнала ТПД подключены по схеме монтажного ИЛИ (если хотя бы в одном КВУ ноль, то во всей линии ноль).



Получив ТПД, процессор завершает выполнение очередной команды, отключает ША, ШД, ШУ (переводит их в третье состояние) и устанавливает сигнал ППД.

Существует 3 типа состояния: ввод, вывод, разрыв (состояние высокого сопротивления). Управление СИ передается КВУ. Контроллер устанавливает адрес памяти, вводит данные и сопровождает эти операции сигналом управления. Так как на схеме два контроллера, то возникает два вопроса:

приоритетность устройств;

идентификация устройства прямого доступа.

Оба вопроса решаются последовательным прохождением сигнала ППД через КВУ.

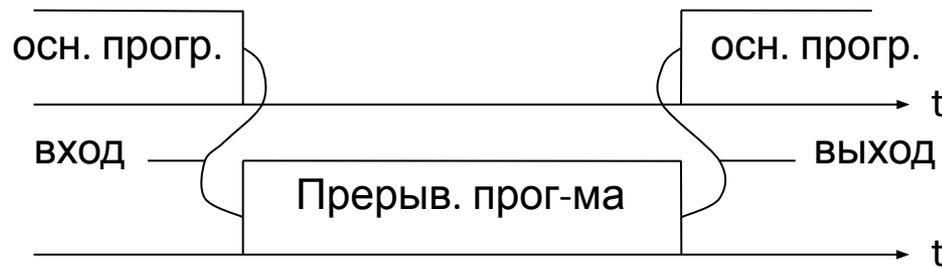
Допустим, одновременно запросили режим прямого доступа оба контроллера, тогда КВУ1 блокирует КВУ2. Если же КВУ2 запросил режим прямого доступа, а КВУ1 – нет, то сигнал ППД пройдет через КВУ2.

Таким образом, контроллер, находящийся ближе к процессору будет обладать наивысшим приоритетом. Последовательное прохождение сигнала через контроллеры называется дейзицепью.

Триггер разрешения прямого доступа, который входит в состав процессора, может устанавливаться или сбрасываться программно, управляя прямым доступом.

Режим работы по прерыванию.

Рассмотрим временную диаграмму прерывания.



Основная программа и прерывающая работают асинхронно, следовательно, очень важно правильно задать вход и выход прерывающей программы. Для любого вычислительного процесса важнейшими являются два значения:

содержание программного счетчика;

слово состояния процессора.

Эти две компоненты задают вектор. Различают вектор текущего процесса и вектор прерывания. Для того чтобы правильно начать прерывание, нужно знать адрес, с которого начинается прерывание и слово состояния, таким образом, чтобы правильно зайти в прерывающую программу, надо загрузить вектор прерывания. Вектор текущего процесса сохраняется при входе в прерывание и восстанавливается при выходе из него. Этот вектор сохраняется в стеке, и вместо него загружается вектор прерывания, при выходе из прерывания вектор текущего процесса восстанавливается из стека.

Достоинство: при использовании режима работы по прерыванию величивается производительность ЭВМ за счет исключения опроса готовности.

Недостаток: для реализации работы режима по обслуживанию прерывания требуются дополнительные аппаратные затраты.

Приоритеты прерывания.

Прерывания подразделяются на внутренние и внешние. Внутренние прерывания контролируют внутренние события вычислительной машины. Внешние прерывания контролируют события, возникающие в ВУ.

Типовые внутренние прерывания:

Прерывание по контролю напряжения сети.

В большинстве ЭВМ, особенно в управляющих, для надежного функционирования программы контролируется состояние источника питания. В случае отключения питания процесс должен корректно прерваться, а при подаче напряжения – продолжиться с прерванного места. Это достигается двумя путями:

- а) энергонезависимой памятью, куда сохраняются все необходимые параметры;
- б) использованием механизма прерывания по напряжению питания.

2) Прерывание по несуществующей команде.

В ряде случаев контролируются коды команд, и если встретится несуществующий код команды, то возникнет ситуация прерывания.

3) Обращение по несуществующему адресу.

Контролируется наличие устройств, имеющих определенные адреса. Если устройство отсутствует, то возникает прерывание.

К внутренним прерываниям можно отнести программные прерывания.

Внешние прерывания.

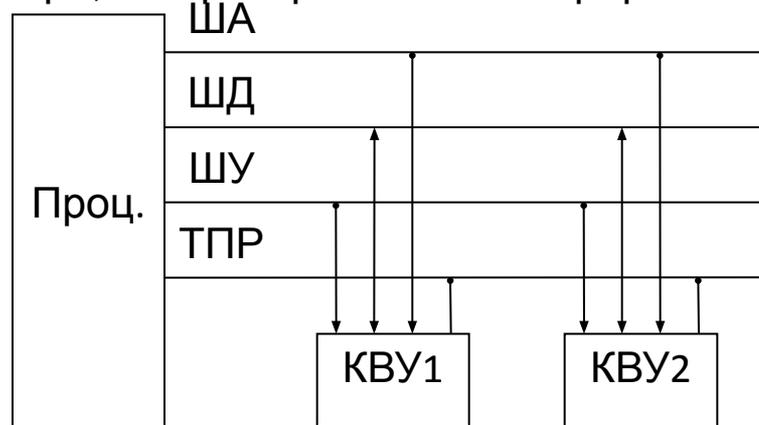
Внешние прерывания связаны с конкретными контроллерами ВУ и предназначены для обработки событий, возникающих во ВУ (например, клавиатура, дисплей, таймер). Таймер используется для фиксации астрономического времени. В качестве примера таймера можно привести внутренние часы.

Говоря о прерываниях, следует сказать, что внутренние прерывания более приоритетны по отношению к внешним.

Схема программного полинга

Под полингом понимаются определение приоритета прерывания и идентификация.

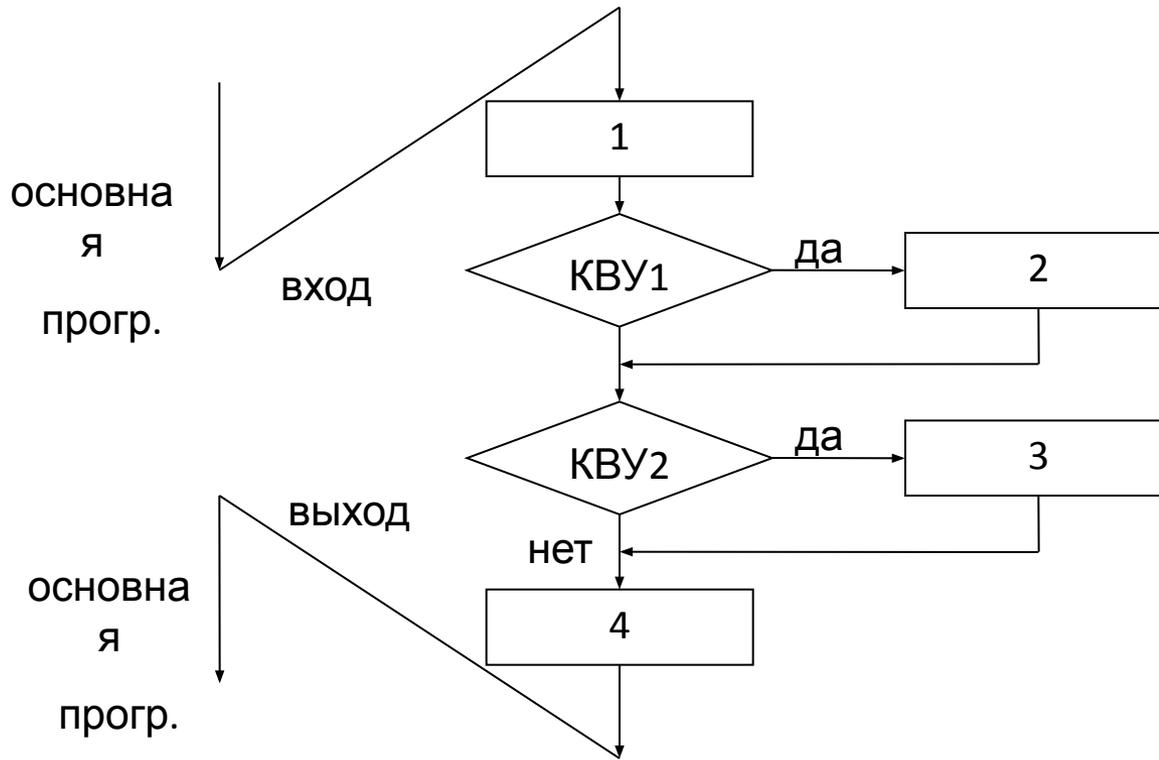
Рассмотрим два контроллера, которые работают по прерыванию:



В аппаратной части СИ содержит дополнительную линию ТПР (требование прерывания).

К этой линии по схеме монтажного ИЛИ подключены два контроллера КВУ1 и КВУ2.

Процессор, получив сигнал ТПР, начинает выполнять алгоритм программного полинга.



Этот алгоритм включает:

Сохранение в стеке внутренних регистров процессора, проверка готовности КВУ1, КВУ2;

Обработка прерывания КВУ1;

Обработка прерывания КВУ2;

Восстановление из стека внутренних регистров.

Приоритетность задается программно. В данном случае, как это видно из алгоритма, приоритет у КВУ1.

Достоинство: простота аппаратной реализации.

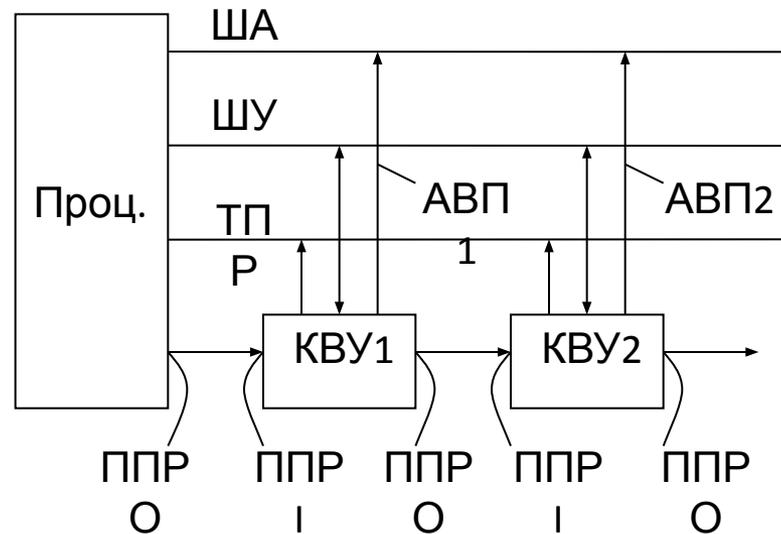
Недостаток: дополнительное время для идентификации устройства, запросившего прерывание. Этот недостаток устраняется в схеме аппаратного полинга.

Схема аппаратного полинга.

Здесь задача полинга решается аппаратно.

Рассмотрим, как осуществляется выдача адреса вектора прерывания (АВП). Под АВП понимается адрес ячейки памяти, где хранится вектор прерывания. Как правило, АВП размещается в младших адресах. Процессор, получив сигнал ТПР (требование прерывания), завершает выполнение очередной команды и проверяет триггер (флаг) разрешения прерывания. Если прерывания разрешены, то устанавливается сигнал ППР (подтверждение прерывания). Получив сигнал ППР, КВУ устанавливает на ША адрес вектора прерывания. Процессор вводит адрес вектора прерывания и продолжает обрабатывать прерывание.

В данном случае КВУ1 имеет больший приоритет, так как к нему сигнал ППР придет первым.



ТПР – VIRQ;

ППР – IAKO (IAKI).

Ситуация прерывания возникает в контроллере в случае, если прерывания разрешены и данные готовы. С помощью схемы И1 формируется сигнал требования прерывания. Этот сигнал передается на ШУ, а также с помощью инвертора 1 и схемы И2 блокирует прохождение сигнала ППР через контроллер.

При поступлении сигнала ППР схема И3 открывает регистр АВП (РАВП) и его состояние устанавливается на ША. АВП набирается перемычками или переключателями на входе РАВП. В ряде случаев АВП жестко назначен в большой интегральной схеме.

Система памяти микро-ЭВМ.

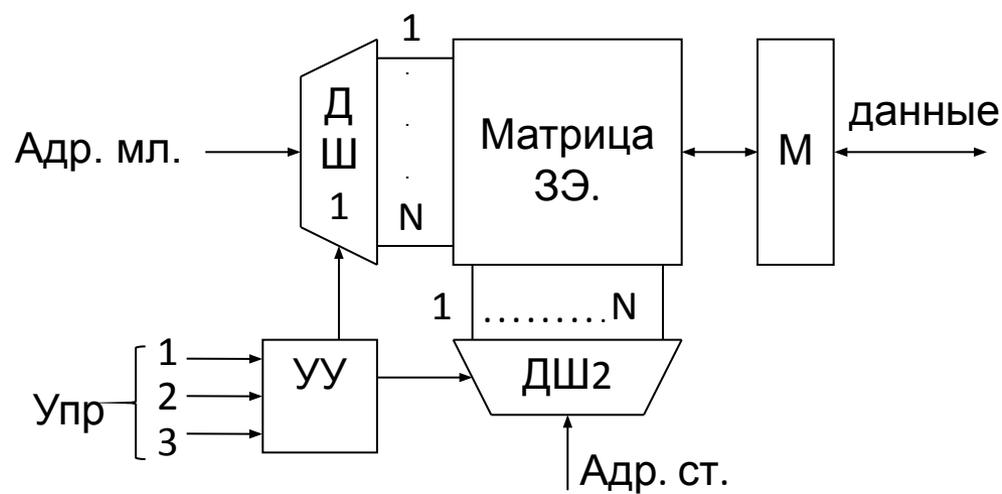
Память подразделяется на внутреннюю и внешнюю. К внешней памяти относятся различные накопители информации. Внутренняя память подключается непосредственно к СИ. В свою очередь внутренняя память подразделяется на постоянное запоминающее устройство ПЗУ и оперативное запоминающее устройство ОЗУ.

ПЗУ предназначено для хранения программ и констант. ОЗУ – для временного хранения программных данных.

Характеристики памяти:

- 1) Объем.
- 2) Организация. По способу организации различают байтовую и словную память.
- 3) Быстродействие. Основная характеристика быстродействия – время выборки, т.е. время от установки адреса до появления информации на выходе памяти.
- 4) Энергопотребление. Измеряется в мВт/Бт.
- 5) Стоимость.

Основу организации памяти составляет матрица запоминающих элементов.



М – мультиплексор, обеспечивающий считывание сразу с нескольких запоминающих элементов.

УУ – узел управления.

ДШ1 и ДШ2 – дешифраторы строки и столбца.

На узел управления УУ подаются три сигнала:

- 1) CS (chip select) – выбор кристалла.
- 2) CE (chip enable) или OE (output enable) – управление выходом микросхемы данных.
- 3) W(write)/R(read) – запись/чтение. Этот сигнал переключает выходы микросхемы из состояния ввода в состояние вывода. Характерен только для ОЗУ.

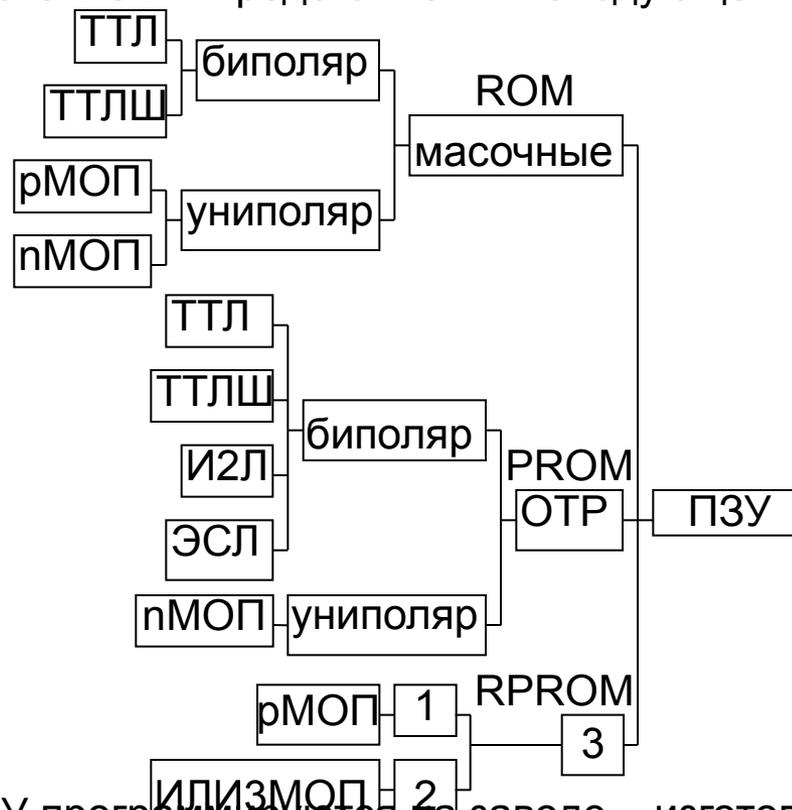
Микросхемы ПЗУ.

В ПЗУ информация хранится длительно и сохраняется при отключении питания. Все микросхемы памяти могут быть выполнены по биполярной или униполярной технологии.

По принципу записи и хранения информации:

- масочные;
- однократно программируемые;
- перепрограммируемые.

Типы ПЗУ и технологии изготовления представлены в следующей схеме



Масочные микросхемы ПЗУ программируются на заводе – изготовителе на завершающей стадии производства. Применяются в массовом и крупносерийном производстве.

Однопрограммируемые (OTP – One Time Programming) микросхемы ПЗУ.

Программирование таких микросхем производится 1 раз на заводе – изготовителе средств вычислительной техники. Для программирования используются два способа:

- выжигание перемычек в матрице 3Э;
- смещение заряда в МОП – структуре.

ОТР применяются в серийном и массовом производстве.

Перепрограммируемые микросхемы ПЗУ. Такие микросхемы позволяют многократно стирать и записывать информацию. Различают перепрограммируемые МС ПЗУ двух типов:

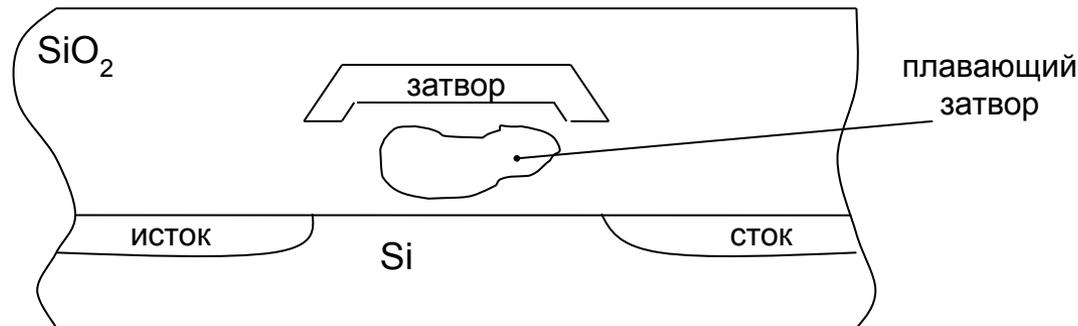
- 1) основаны на электрическом стирании и электрическом программировании;
- 2) основаны на ультрафиолетовом стирании и электрическом программировании.

В обоих случаях программирование осуществляется смещением заряда в МОП – структуре.

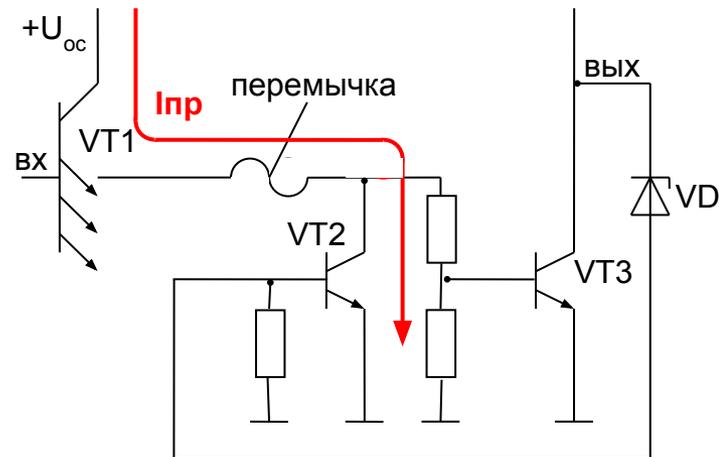
Принцип записи и хранения в МОП – структуре.

МОП – металл, окисел, полупроводник.

В область затвора искусственно помещают заряд, получая так называемый плавающий затвор, который будет экранировать действие основного затвора. Для ПЗУ необходимо вводить заряд относительно большой величины, слой окисла также должен быть большим – это позволяет получать большое время жизни заряда, т. е. информация в ПЗУ может храниться достаточно долго.



Способ записи и хранения информации путем выжигания перемычек.



В схеме имеется перемычка из материала высокого сопротивления и малого сечения.

Если имеется перемычка, то входной сигнал открывает основной выходной транзистор $VT3$, если нет - $VT3$ остается в закрытом состоянии при приложении входного сигнала. Суть программирования заключается в выжигании перемычки. Для этого подается входной сигнал, а на выход напряжение, большее, чем порог пробоя VD . Вспомогательный программируемый транзистор $VT2$ открывается, по цепи течет программируемый ток $I_{пр}$, перемычка выгорает.

Недостаток этого метода заключается в том, что около 10% всех перемычек со временем восстанавливаются.

Микросхемы ОЗУ.

ОЗУ предназначено для временного хранения программ и данных.

В микросхеме статического ОЗУ информация сохраняется длительно без каких-либо дополнительных действий. Основой микросхем статического ОЗУ является триггер.

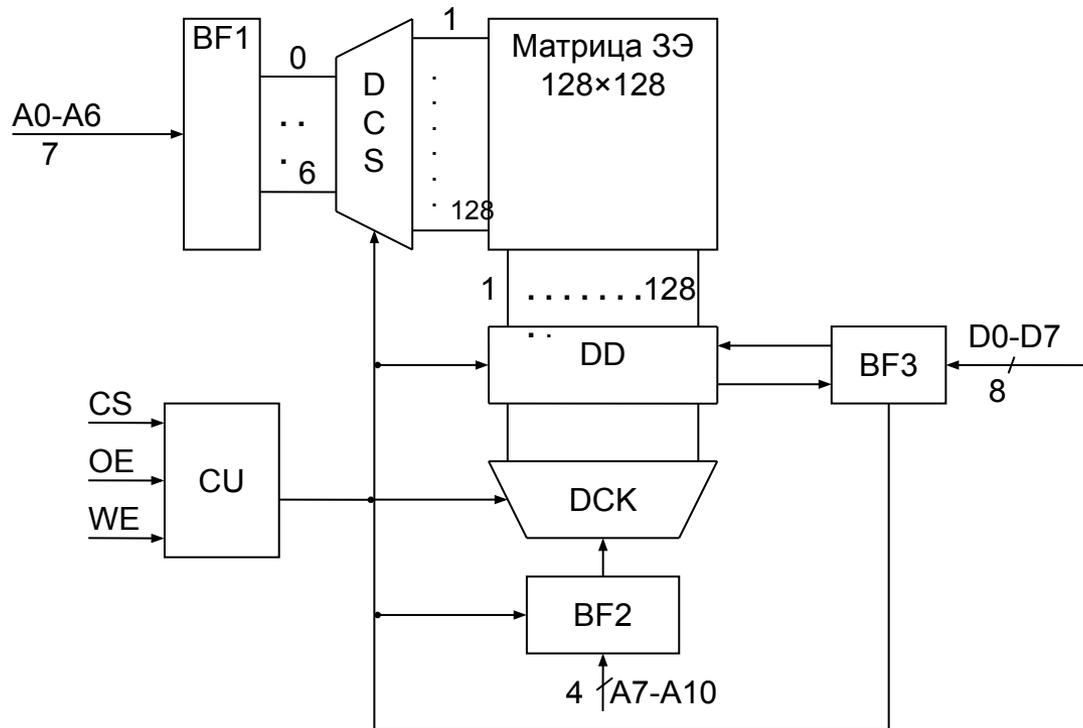
В микросхемах динамического ОЗУ элементом хранения является МОП – структура.

Для записи информации используется перемещение (накачка) заряда. Величина заряда для ОЗУ невелика, поэтому в течение времени он рассасывается, т.е. информация повреждается. Для поддержания заряда используется периодическая регенерация содержимого ОЗУ. Динамическая ОЗУ характеризуется большой емкостью на одном кристалле.

Все многообразие микросхем ОЗУ можно представить в виде следующей схемы



Структура микросхемы статической ОЗУ на примере МС КР537РУ10 емкостью 2048×8 бит.



Типовым элементом хранения является триггер на 6 МОП транзисторах. Основу микросхемы составляет матрица 128×128.

На схеме обозначены:

DCS – дешифратор номера строки;

DCK – дешифратор номера столбца;

DD – мультиплексор;

BF1, BF2, BF3 – три буфера для согласования уровней сигнала;

CU – узел управления, на который подаются 3 управляющих сигнала:

CS – выбор кристалла;

OE – управление выходом микросхемы памяти;

WE – запись/чтение.

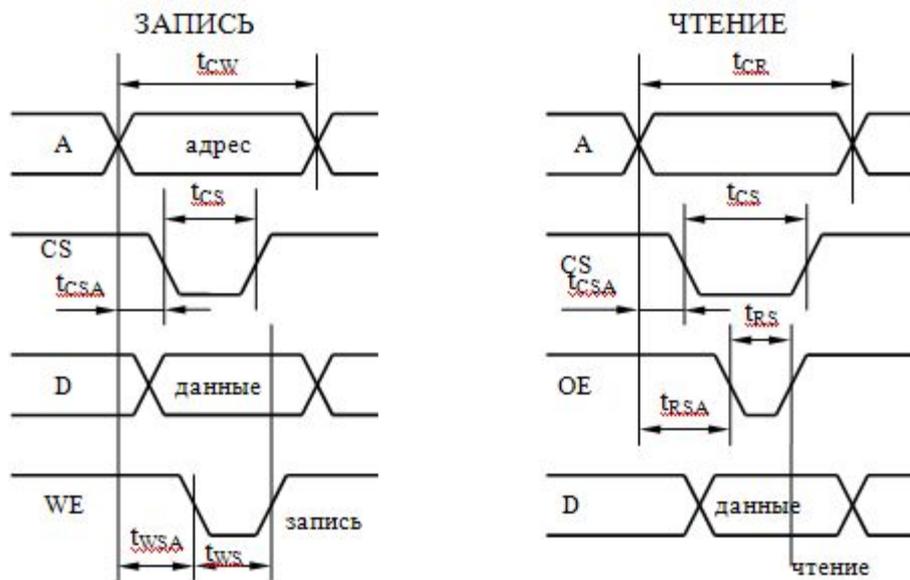
CS	WE	OE	D	Режим
1	*	*	*	Хранение
0	0	*	D1	Запись
0	1	1	Z	Чтение 1
0	1	0	D0	Чтение

Если сигнал CS равен 1, то микросхема находится в режиме хранения. Когда сигнал CS активного уровня, то возможна запись при WE низкого уровня – входные данные поступают на ячейки матрицы ЗЭ.

При CS=0 и WE=1 микросхема может находиться в режиме «Чтение» и «Чтение1».

Режим «Чтение» характеризуется выдачей данных на выход микросхемы. При работе микросхемы в режиме «Чтение1» происходит чтение без выдачи данных.

Ниже представлены временные диаграммы, поясняющие режимы «Запись» и «Чтение»



Запись производится по фронту импульса WE, t_{CW} – время цикла записи.

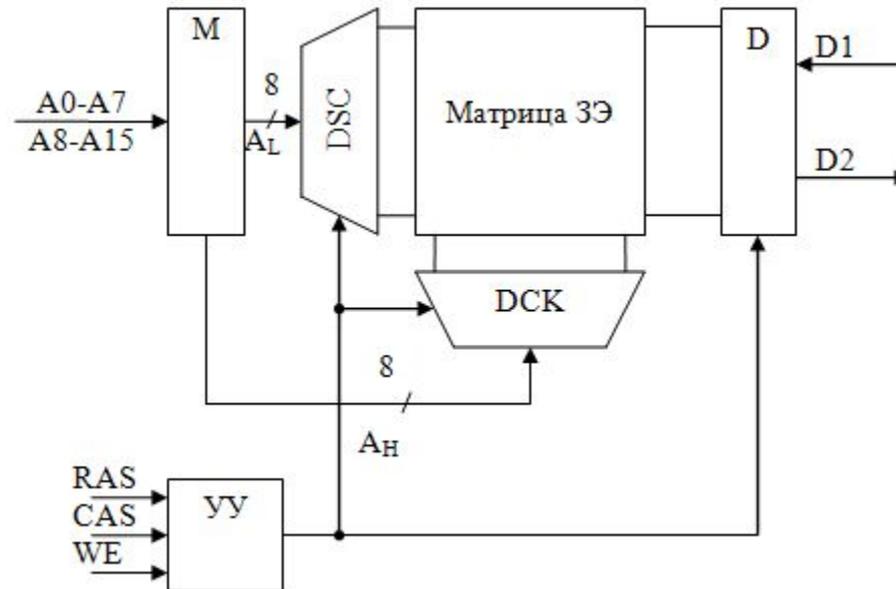
Чтение производится по фронту импульса OE, t_{CR} – время цикла чтения.

Микросхемы динамического ОЗУ.

Микросхемы динамического ОЗУ обладают значительно большей емкостью, нежели микросхемы статического ОЗУ. Основная сложность изготовления этих микросхем – увеличение числа выводов при сохранении ее малых габаритов. Решением этой задачи является мультиплексирование шины адреса. Адрес передается за 2 приема (2 такта).



В качестве примера динамического ОЗУ рассмотрим микросхему КР565РУ5 объемом 64×1кбай



Основа микросхемы – матрица 3Э. Организация битовая. В микросхеме имеются 2 дешифратора DCS и DCK. На дешифратор строки подается младшая часть адреса, на дешифратор столбца – старшая. Для переключения используется демультиплексор М. Входы и выходы данных снабжены буферами данных D. Имеется также узел управления CU, на который подаются три сигнала управления:

- RAS – сопровождает младшую часть адреса и все операции с памятью. Если он неактивного уровня («1»), то микросхема находится в режиме хранения и никаких действий с ней не производится;
- CAS – сопровождает старшую часть адреса;
- WE – запись/чтение. Когда этот сигнал активного уровня («0»), то происходит запись D1; если же WE неактивного уровня, то происходит чтение, т.е. на выходе будут данные D0.

Состояние микросхемы отражено в следующей таблице

RAS	CAS	WE	D1	D0	Режим
1	*	*	*	Z	Хранение
0	0	0	D1	Z	Запись
0	0	1	*	D0	Чтение
0	1	*	*	Z	Регенерация

Ниже приведена диаграмма цикла чтения.

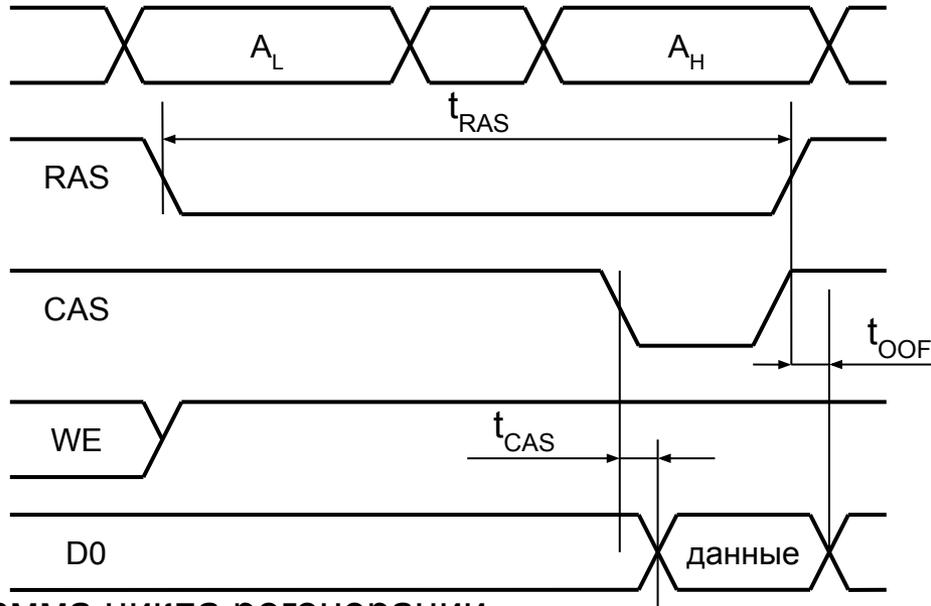
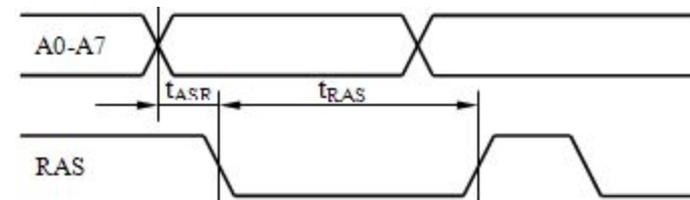


Диаграмма цикла регенерации

t_{CAS} – время задержки.



Для разных микросхем время регенерации различно, но не превосходит 20 мс.

Динамические ОЗУ строятся с помощью контроллера динамической памяти (для регенерации ОЗУ).

Контроллеры динамического ОЗУ.

Контроллеры динамического ОЗУ можно классифицировать следующим образом:

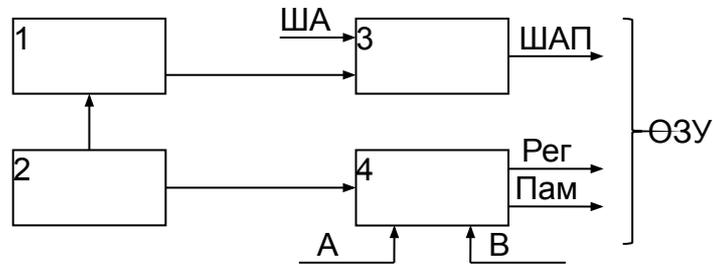
1. Контроллеры асинхронной регенерации. Процесс регенерации и обращения к памяти осуществляется асинхронно. Для реализации используется регулятор регенерации.

2. Контроллеры синхронной регенерации. Процесс регенерации и процесс обращения к памяти синхронизированы тактовой частотой процессора. При этом возникают конфликтные ситуации, когда необходимо проводить регенерацию и в то же время выводить, либо вводить, информацию в процессор.

3. Контроллеры полусинхронной регенерации. Обращение к памяти синхронизируется фронтом тактового импульса, а регенерация – его спадом.

Для контроллеров динамической памяти характерно увязывание их действий с действиями процессора.

Схему контроллера динамического ОЗУ.



Контроллер включает в себя:

- счетчик сток регенерации;
- генератор запрета регенерации;
- мультиплексор (коммутатор) адреса;
- узел управления.

ШАП – шина адреса памяти.

Контроллер формирует два сигнала: РЕГЕНЕРАЦИЯ и ПАМЯТЬ.
Регенерация будет осуществляться с частотой генератора.

Наращивание объемов памяти.

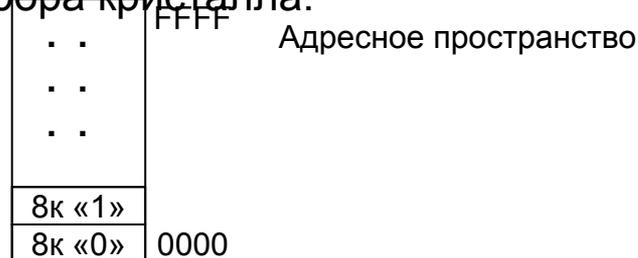
Задача наращивания объемов памяти возникает в двух случаях:

- когда емкости одной микросхемы памяти недостаточно;
- когда требуется объединить ОЗУ и ПЗУ.

Наращивание происходит в двух направлениях:

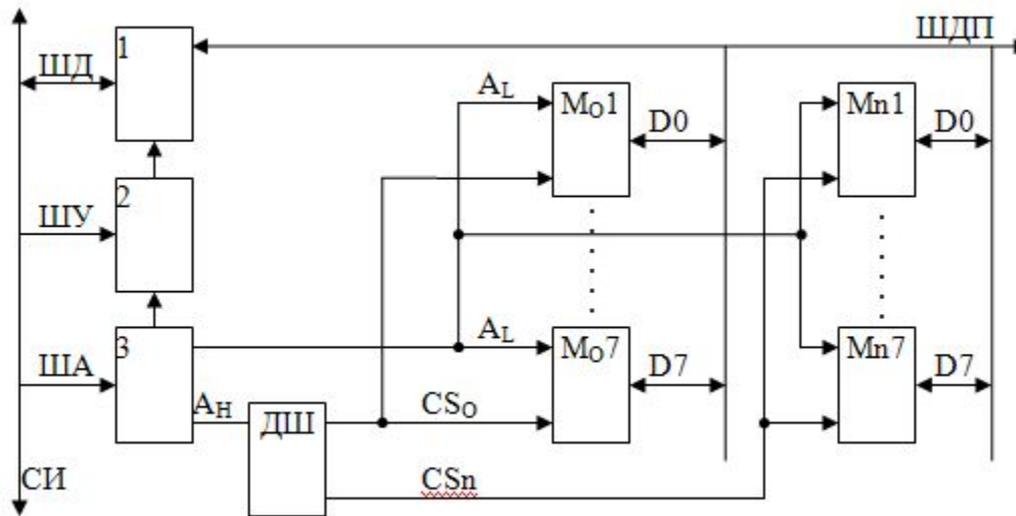
- по числу разрядов – задача возникает в случае, когда микросхемы имеют битовую и 4-ехразрядную организацию;
- по числу адресов – если емкости отдельной микросхемы недостаточно.

Наращивание объема по числу разрядов осуществляется параллельным соединением всех выводов микросхемы кроме линии данных; наращивание по числу адресов – параллельным соединением всех выводов микросхемы кроме линии выбора кристалла.



Адресное пространство можно условно разбить на банки размером, например, по 8 кбайт. Тогда будет 8 банок. Нужно 8 микросхем для заполнения адресного пространства. У каждой микросхемы свой сигнал выбора кристалла CS. Этот сигнал формируется путем дешифрирования старшей части адреса (3 старших разряда, т.к. 8 комбинаций определяется 3 разрядами адреса). Каждая микросхема имеет битовую организацию. Для организации байтовой системы нужно 8 микросхем. Наращивание по числу разрядов (по вертикали). Все выводы микросхем включаются параллельно, кроме линии данных. Наращивание по числу адресов (по горизонтали).

Пример наращивания как по числу разрядов, так и по числу адресов.

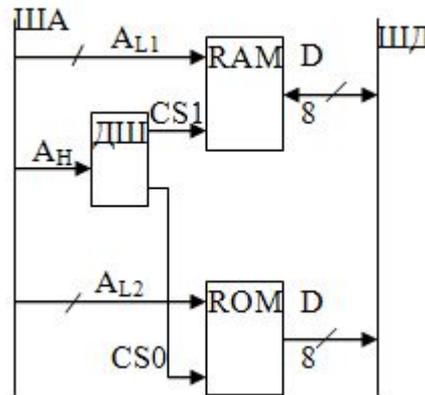


На схеме:

- приемо-передатчики ШД;
- передатчики ШУ;
- передатчики ША.

Объединение ОЗУ и ПЗУ в единое адресное пространство.

Для управляющих ЭВМ характерен большой объем ПЗУ и небольшой объем ОЗУ. Разграничение ОЗУ и ПЗУ осуществляется путем дешифрации старшей части адреса. Линии данных младшей части адреса соединены параллельно и образуют единое адресное пространство. Сигналы управления формируются в соответствии с логикой работы конкретных микросхем и могут использоваться отдельно.

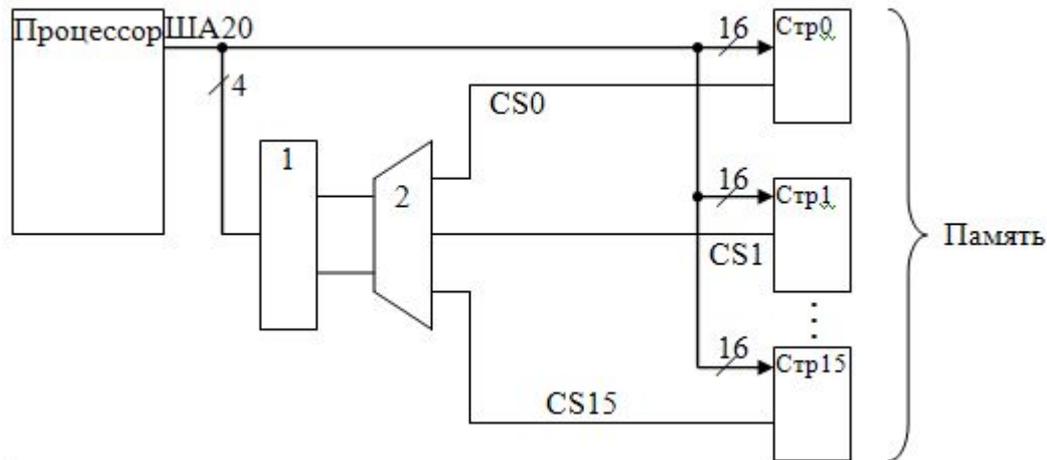


RAM - память произвольного доступа, обеспечивает ввод/вывод данных по произвольно установленному адресу. Альтернативой выступает память последовательного доступа ROM

Страничная организация памяти.

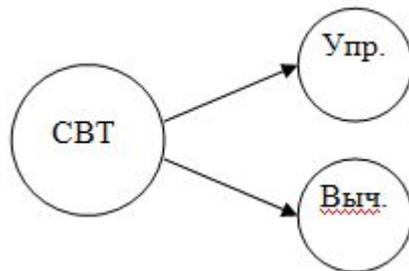
Применяется в том случае, когда объем адресного пространства превышает адрес, записанный в команде.

Процессор имеет 20-разрядную шину адреса. 16 страниц по 64 кбайта соединены параллельно, кроме линии выбора кристалла, которая формируется из старшей части адреса. Для этого используется регистр номера страницы 1 и дешифратор номера страницы 2.



Микроконтроллеры.

Средства вычислительной техники (СВТ) используются для управления и для вычислений.



ОЭМ – однокристальные микро-ЭВМ (отечественное название микроконтроллеров). Микроконтроллер – микропроцессор, снабженный ОЗУ, ПЗУ и ВУ (портами ввода/вывода), причем все это выполнено на одном кристалле.

- Микропроцессор.
- RAM.
- ROM.
- Порты ввода/вывода.

Одна из первых такое устройство как микроконтроллер предложила фирма Intel. Этот микроконтроллер был назван 8051 или 80С51. Современные микроконтроллеры, как правило, 8-разрядные, реже используются 16-разрядные микроконтроллеры. Существуют также 32-разрядные, но они почти не применяются.

К особой ветви микроконтроллеров можно отнести цифровые сигнальные процессоры (DSP). Они обладают большой разрядностью и высоким быстродействием.

Наибольшее распространение получили микроконтроллеры с системой команд 51-го кристалла. Условно это семейство микроконтроллеров называют MCS-51.

Базовая структура (ядро MCS-51).

- Процессор с аккумулятором (A).
- (256/128) RAM.
- ROM ПЗУ имеет разные объемы памяти (от 0 до 64 кбайт). Типы ПЗУ также разнообразны: однократно программируемые, перепрограммируемые.
- 4 8-разрядных порта ввода/вывода с возможностью побитовой адресации. Часть ввода/вывода портов используется для специализированных целей:
 - 2 входа прерывания;
 - 2 таймера-счетчика;

UART – последовательный порт ввода/вывода. UART обеспечивает асинхронный обмен данными по двум линиям (ввода и вывода) с формированием посылки.

Время выполнения команды – 8 мкс, частота – до 12 МГц.

Усовершенствованный микропроцессор C517 имеет следующие отличия:

- АЦП (ADS);
- устройство быстрой арифметики с высокой разрядностью;
- увеличенное число входов прерывания;
- формирование ШИП сигналов (сигналов широтно-импульсной регуляции PWM).
- устройство сравнения и захвата (CCU). Ряд задач возлагается на эти специализированные узлы, частично разгружается процессор.

Фирма Atmel специализируется на микроконтроллерах с flash-памятью. Эти микроконтроллеры отличаются большим объемом памяти на кристалле, причем вся память flash-типа. Для этого микроконтроллера характерно наличие синхронных интерфейсов I²C, SPI. Они реализованы аппаратно, позволяют включать дополнительную память, АЦП, ЦАП с последовательным вводом и выводом данных.

Фирма Atmel одна из первых предложила технологию ISP, преимущество которой в том, что без снятия микросхемы с устройства можно изменить программу в ней.

Дальнейшее развитие микроконтроллеров было связано с разработкой микропроцессоров с RISC-структурой. Для них характерна конвейерная обработка команд. Они позволяют обрабатывать любую команду за 1 такт.

AVR – микроконтроллер, выполненный по RISC-технологии. Он может осуществлять 16 млн. операций в секунду.

Фирма Signal объединила наиболее развитую периферию и на RISC-ядре выполнила систему команд 51-го кристалла. Внедрение RISC-структуры увеличило скорость до 20-25 млн. операций в секунду при соответствующей тактовой частоте 20-25МГц.

Семейство микроконтроллеров MSC-51.

Организация памяти.

Для микроконтроллеров семейства MSC-51 характерна структура с отдельной памятью программ и памятью данных.

Различают два вида памяти:

- внутренняя (на кристалле);
- внешняя.

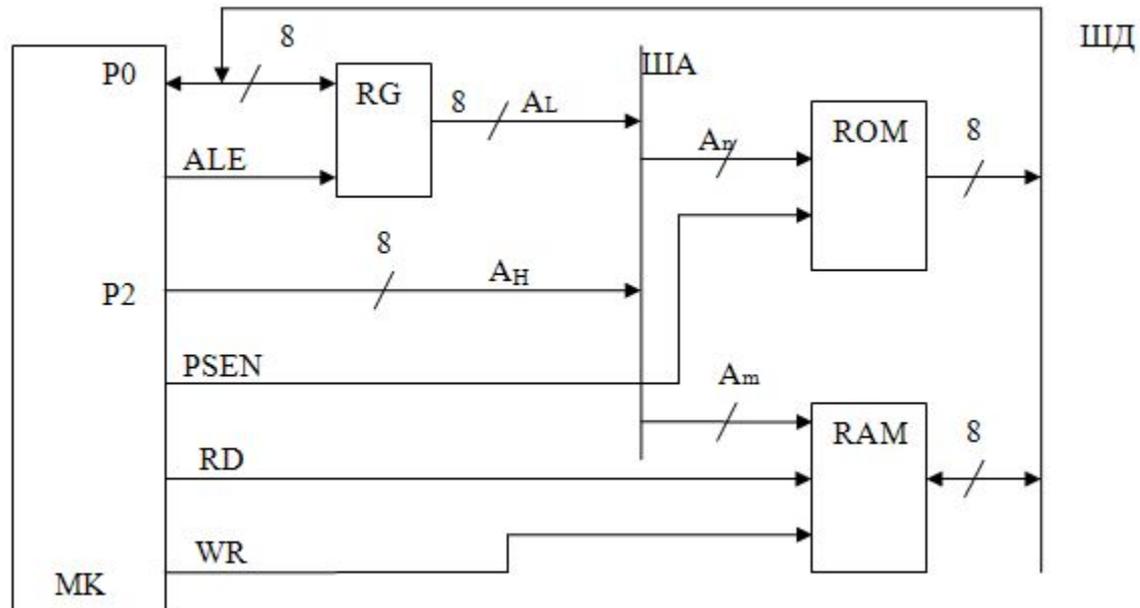
Микроконтроллеры семейства MCS-51 имеют следующие характеристики:

- максимальный объем памяти – 64 кБт;
- 32 порта ввода/вывода.

Для того чтобы подключить внешнюю память программ и внешнюю память данных по линии порта P0 передается младшая часть адреса и данные в мультиплексном режиме, по линии порта P2 передается старшая часть адреса. Две линии порта P3 используются для передачи сигналов управления RD и WR (чтение и запись).

Подключение внешней памяти к микроконтроллеру семейства MCS-51.

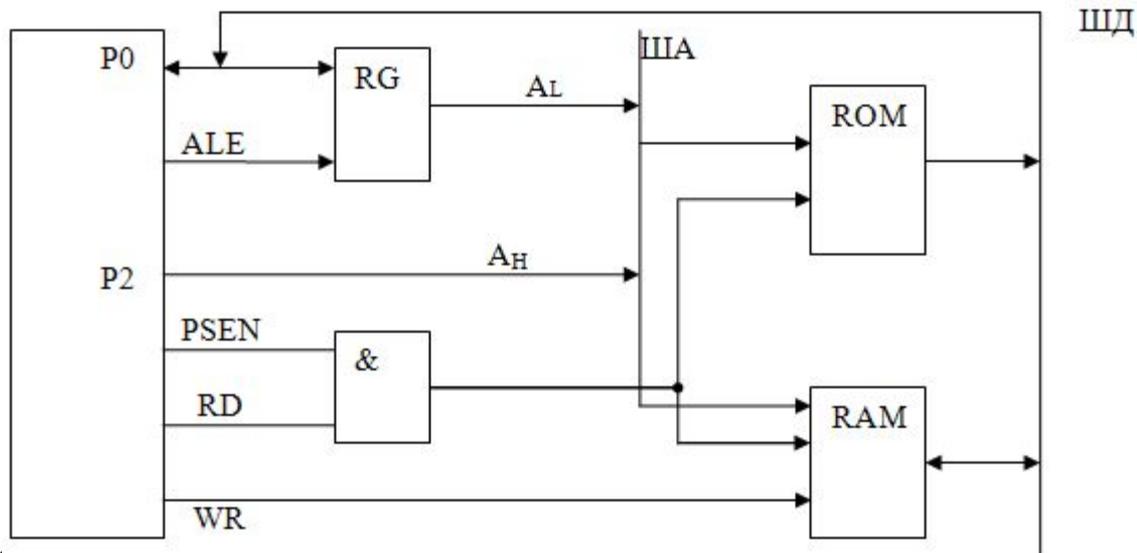
Для разделения адреса и данных используется регистр-защелка. Сигнал ALE сопровождает передачу данных по линии порта P0. Для обращения к внешней памяти программ микроконтроллер использует сигнал PSEN. Этот сигнал используется для того, чтобы разделить память программ и память данных. 3 порта микроконтроллера используются для подключения внешней памяти программ и внешней памяти данных. При этом порт P0 в мультиплексном режиме передает младшую часть адреса и данные. Для выделения младшей части адреса используется регистр-защелка. Порт P2 передает старшую часть адреса.. По порту P3 передаются сигналы запись и чтение. Эти сигналы формируются микроконтроллером при обращении к внешней памяти при помощи определенных команд.



Объединение адресного пространства для MCS-51.

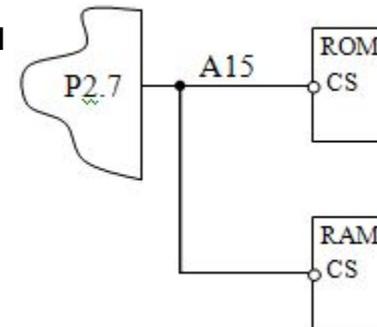
При объединении памяти программ и памяти данных увеличивается гибкость программирования, появляется возможность загружать в ОЗУ и отлаживать программы, однако, уменьшается максимально доступный объем памяти.

Такое объединение осуществляется следующим образом:



Отличие заключается в том, что после формирования результирующий сигнал подается на обе микросхемы. Адресное пространство должно быть поделено между ROM и RAM.

Наиболее просто это деление можно осуществить так как показано на рис. ниже. При этом память равномерно распределяется между ОЗУ и



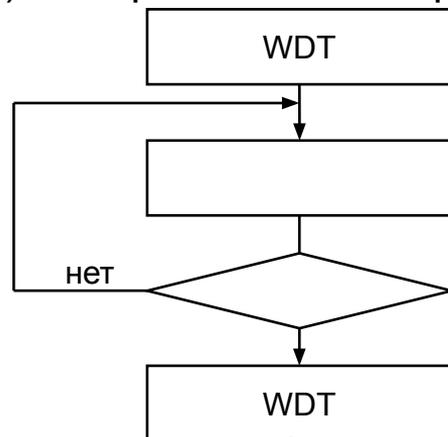
Структурная схема микроконтроллера семейства MCS-51.

Рассмотрим структурную схему микроконтроллера AT89S8252 фирмы «Atmel». Микроконтроллер имеет встроенную Flash-память, в нем также реализована ISP-технология.

На схеме указаны:

- ACC – аккумулятор;
- ALU – арифметико-логическое устройство с двумя регистрами временного хранения данных: TMP1 и TMP2;
- PSW – регистр слова состояния процессора;
- SP – указатель стека;
- RA1, RA2 – два регистра адреса внутренней памяти данных;
- RAM – внутренняя память данных (128 байт), в которой предусмотрено два банка регистров по 8 регистров в каждом, область регистров специальных функций, где хранятся настройки данной микросхемы, область с битовой организацией;
- CU – узел управления, в который входит регистр команд RC, где временно хранится код выполняемой программы. На узел управления подаются 4 управляющих сигнала:
 - PSEN – сигнал, сопровождающий операцию чтения кода команды;
 - ALE – сопровождает младшую часть адреса на линиях порта P0;
 - EA – включение/отключение внутренней памяти программ;
 - RST – для сброса внутренних регистров процессора в исходное состояние. После этого сигнала микроконтроллер начинает выполнять программу с нулевого адреса.
- BUF – буфер для внутреннего согласования отдельных элементов;
- PC – программный счетчик;
- PCI – инкрементатор программного счетчика – устройство для изменения адреса;
- DPTR – регистр смещения; в микроконтроллерах семейства MCS-51 имеются команды, где исполнительный адрес формируется сложением адреса со смещением;
- FLASH – Flash-память с электрическим программированием и электрическим стиранием.

- А – узел прерываний. В микроконтроллере имеются два прерывания, которые могут маскироваться. Также имеются два таймера-счетчика, которые программируются в различных режимах работы
- WDT(watch dog timer) – сторожевой таймер, который позволяет избежать зацикливаний

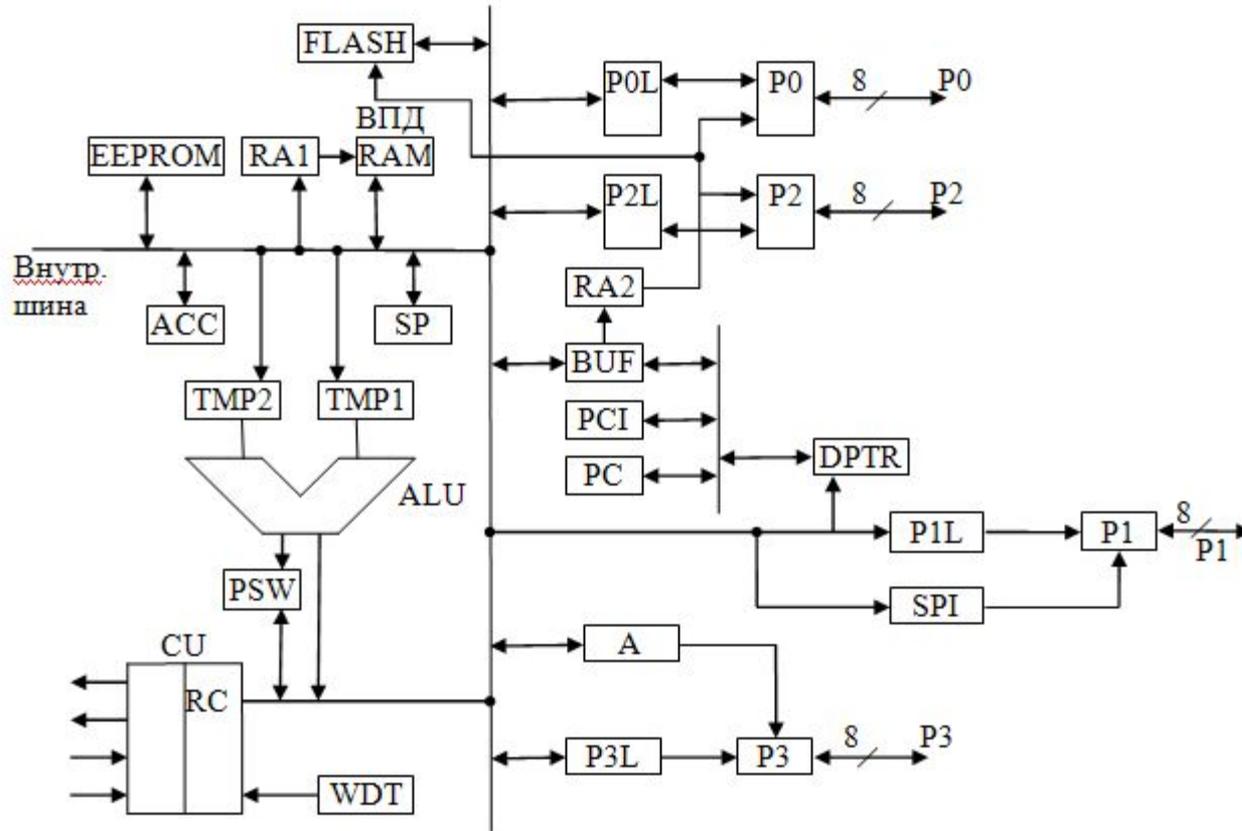


Данный микроконтроллер имеет 4 порта. Все они снабжены регистрами-защелками P0L, P1L, P2L, P3L. Выходная информация может быть записана в эти регистры и сохранена до следующей перезаписи. Каждый порт снабжен выходным буфером, в которых согласуются внешние ТТЛ-сигналы с внутренними сигналами микроконтроллера, прежде всего по нагрузочной способности.

P0, P1, P2, P3 – выходные буферы. Через P0 и P2 передается адрес в режиме работы с внешней памятью данных.

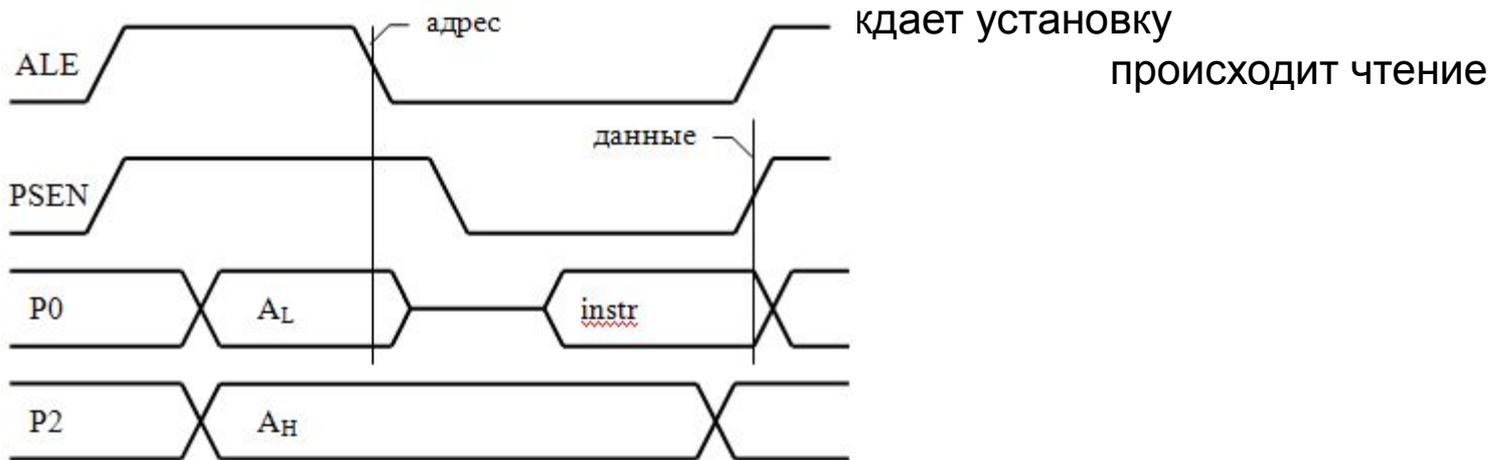
- EEPROM – внутренняя перепрограммируемая память.

Структурная схема микроконтроллера AT89S8252 фирмы «Atmel».



Временные диаграммы работы контроллера с внешней памятью.

Временные диаграммы чтения внешней памяти программ:

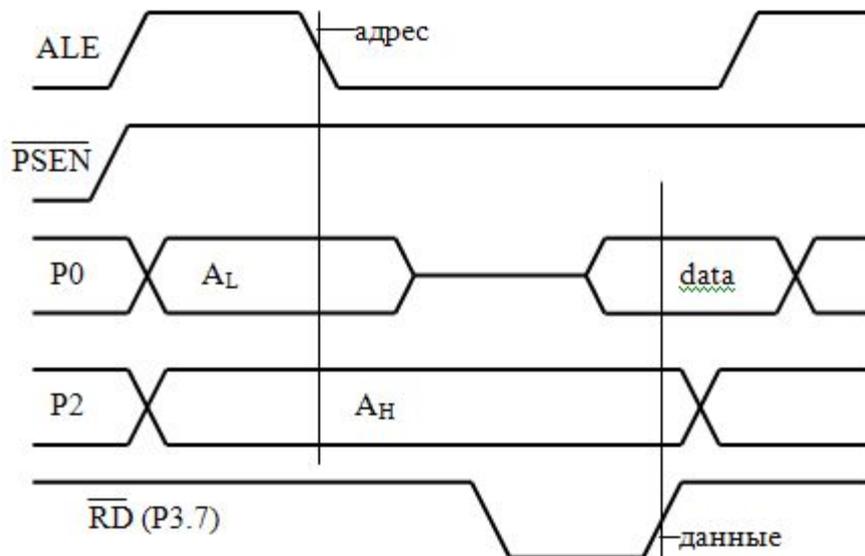


Временные диаграммы чтения внешней памяти данных:

Здесь сигнал PSEN не участвует, для

этого используется сигнал \overline{RD} по

фронту которого



Временные диаграммы записи во внешнюю память данных

