



**Процессор и**

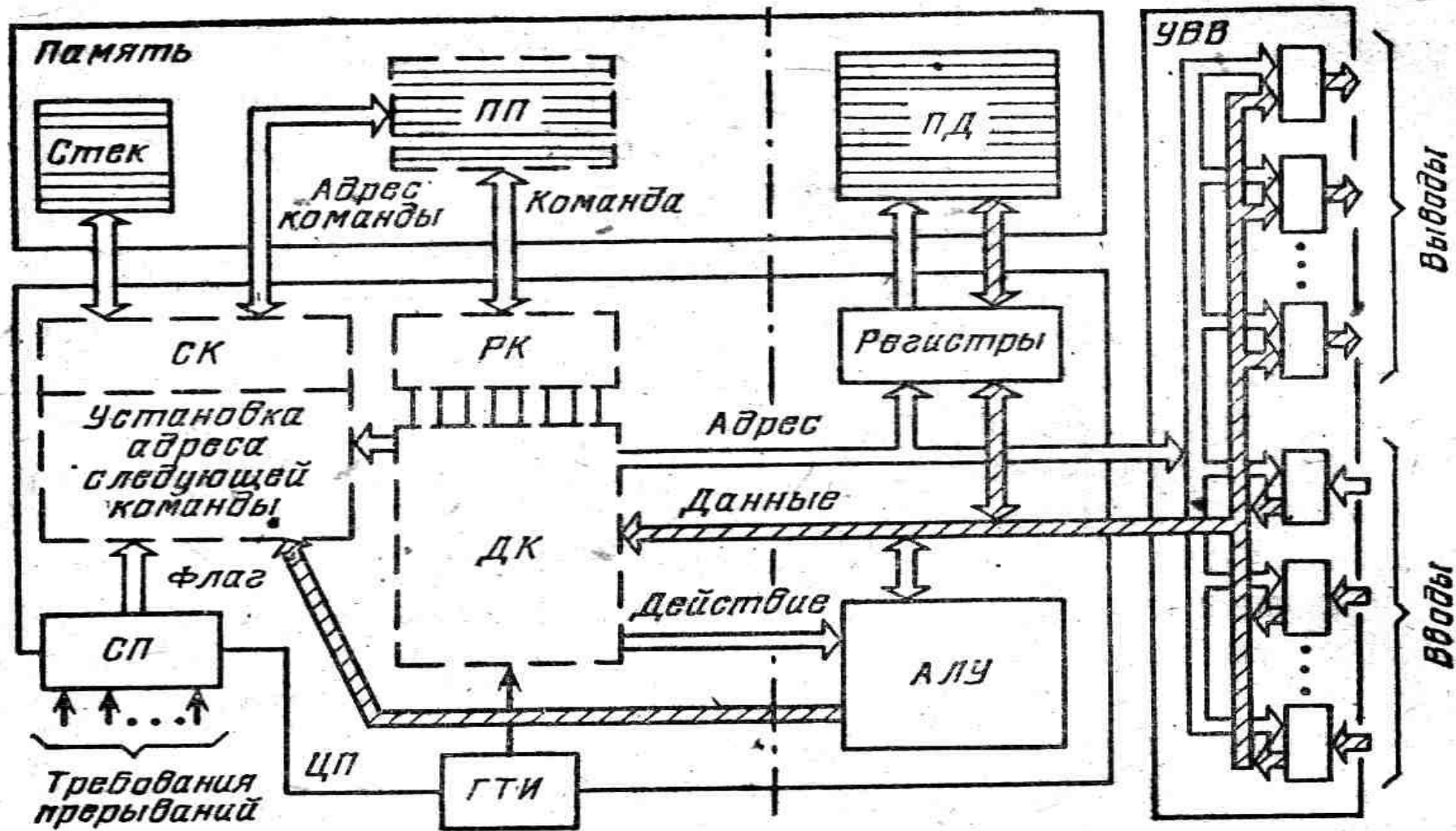
TELKOM-3

**память**

**(продолжение)**

«ЭКСПРЕСС-АМ5»

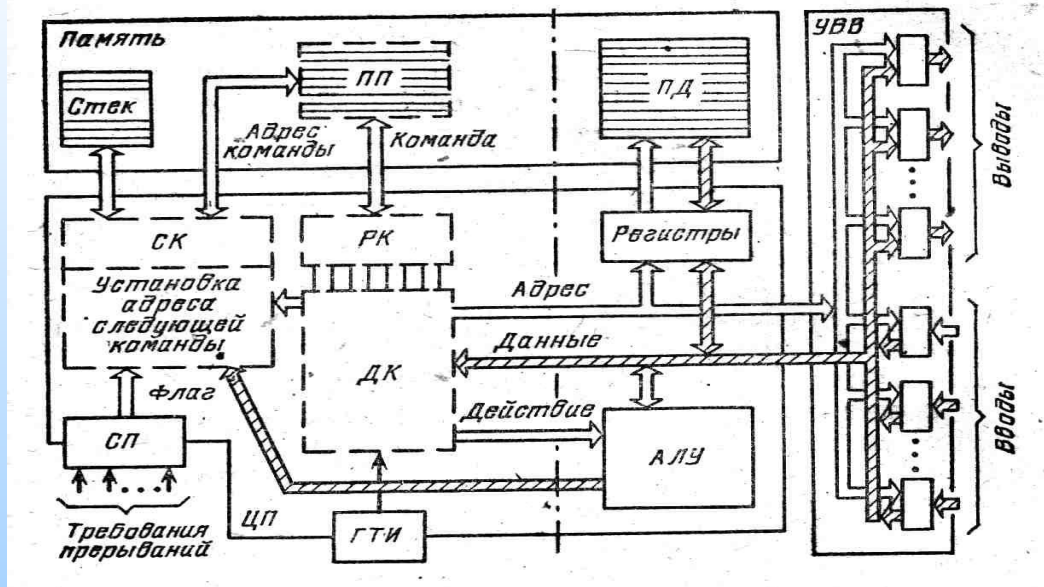
# Функциональная схемы микропроцессора



# Функциональная схема микропроцессора

Перед вами самый простой микропроцессор.

Рассмотрим его состав. Мы видим 3 прямоугольника: память, ЦП и УВВ

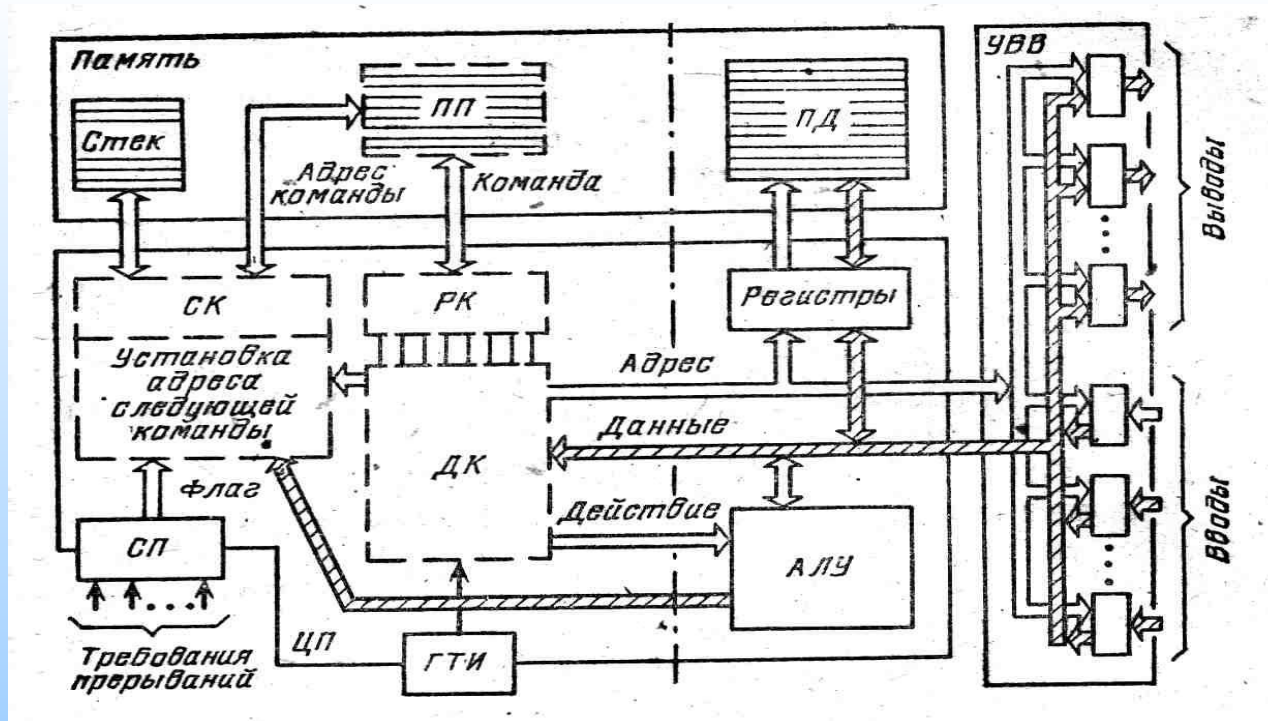


Управляющая часть (пунктирные линии)  
запускается тактовыми импульсами  
Управляемая часть (сплошные линии), в  
соответствии со считываемой из  
программной памяти  
последовательностью команд  
побуждается к выполнению  
определенных операций обработки и  
передачи информации между АЛУ,  
программной памятью и порта-ВВ.



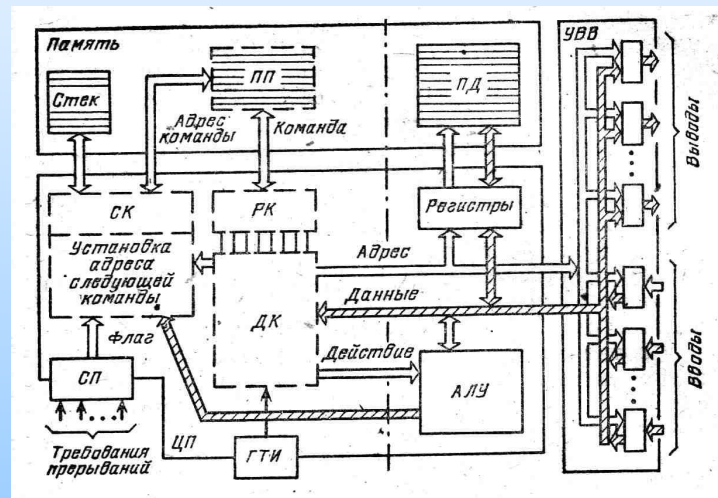
# Функциональная схема микропроцессора

Пунктиром показаны РК, ДК, СК, память программ и схема установки адреса следующей команды.



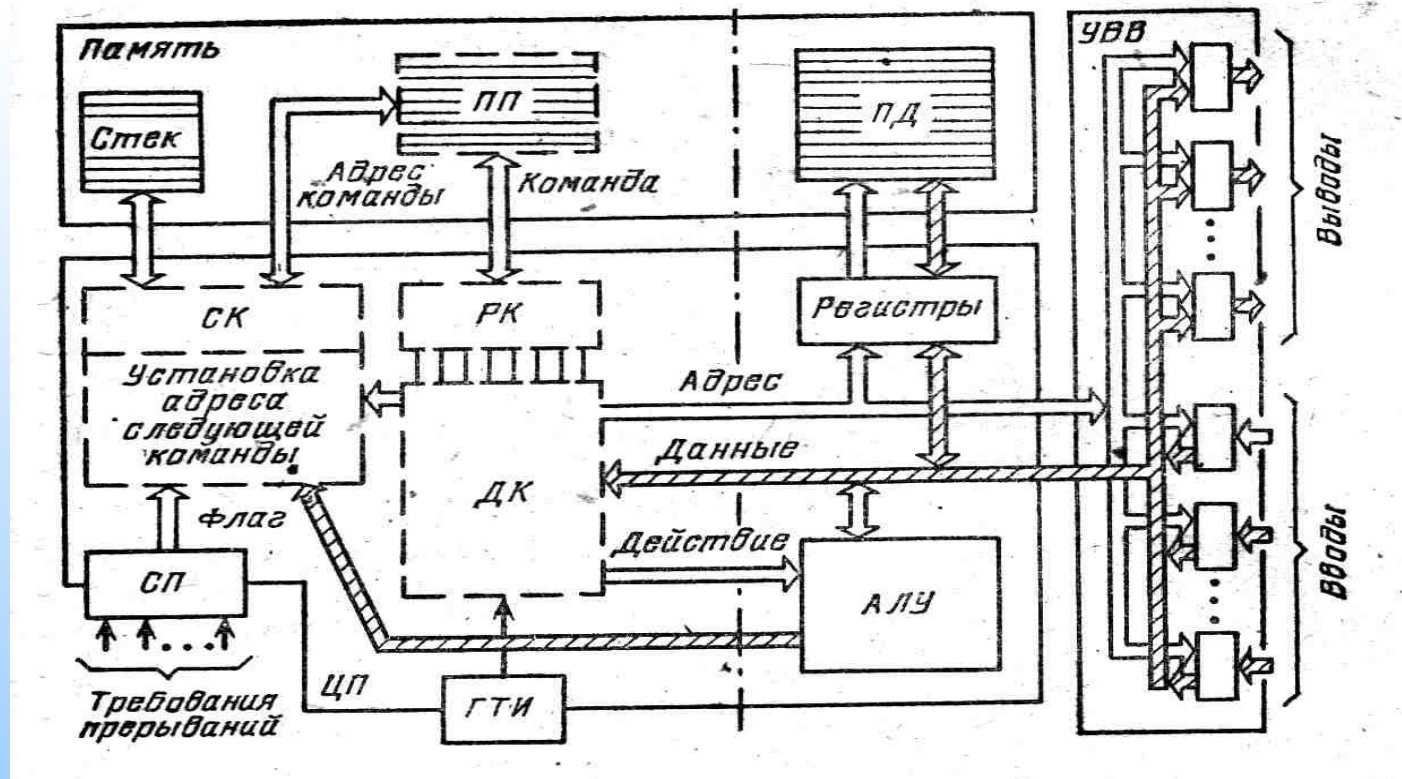
## Функциональная схема микропроцессора

РК - регистр команд хранит в течение цикла выполняемую команду  
ДК – дешифратора команд, который вырабатывает управляющие сигналы для прочих модулей ЭВМ в соответствии с тактовыми импульсами  
СК - счетчик команд служит для хранения ранее подготовленного адреса программной памяти



# Функциональная схема микропроцессора

Сплошной линией показаны память данных, АЛУ, регистры, стек, устройства ввода-вывода



Стек (памяти состояния) - сохраняет содержимое счетчика команд и возможную дополнительную информацию о состоянии ЦП к моменту прерывания

АЛУ - арифметическо-логическое устройство является ядром ЭВМ. Управляющим сигналом, посылаемым дешифратором команд, в этом модуле выполняется по необходимости логическая или арифметическая операция с одним или двумя операндами, результатом которых является новое двоичное слово. Обычно это происходит за один период тактовых импульсов.



ПД – память данных – сохраняет двоичные слова (данные, операнды) для обработки в АЛУ и результат вычислений.

Регистры и КЭШ-память – сверхоперативная память данных, расположена рядом с АЛУ для ускорения доступа к данным

Порты ВВ являются устройствами связи между ЭВМ и внешними устройствами.

Связь между управляющей и управляемой частью микропроцессора:

Связь между этими частями осуществляется главным образом по сигналам системы управления, кроме двух случаев.

- Для выполнения условных переходов необходима обратная связь правой (управляемой) части с модулем, определяющим адрес следующей команды, чтобы передать признаки (выставить флаг) последнего результата.
- Кроме того, для управляющей части должна иметься возможность вернуться к содержимому регистров.

**ГТИ** - генератор тактовых импульсов  
вырабатывает тактовые импульсы

**СП** - система прерываний воспринимает  
требование прерывания и после того, как  
выполнится текущая команда,  
содержимое счетчика команд сохраняется  
и устанавливается начальный адрес  
обслуживания прерывания



Приведенный рисунок функциональной схемы поясняет прежде всего возможные функциональные связи между модулями ЭВМ. Практическая их реализация может быть несколько иной. Так, различные типы памяти могут объединяться в блок памяти и соответственно этому подсоединяться совместно на общую магистраль связи. Как правило, в более сложных процессорах, существует специальный процессор для организации работы памяти, и специальный процессор для организации ввода-вывода.

Двухадресные процессоры всегда запоминают результат на месте первого операнда. Таким образом, в команде достаточно указать лишь два адреса, но сообщаться будет первый, операнд.

Одноадресные процессоры это такие процессоры, в которых с самого начала установлено определенное место в памяти (регистр). Этот регистр еще называют аккумулятором (регистром результата).

Следовательно, в команде необходимо задать только один адрес для второго операнда, но перед обработкой надо прежде всего передать первый операнд в аккумулятор.

Рассмотрим систему команд микропроцессора

Имеется 4 типа команд микропроцессора

- **Команды обработки**
- **Команды передачи данных**
- **Команды ветвления программ (команды перехода, команды вызова)**
- **Команды управления**

## Команды обработки (одноадресный процессор)

Т а б л и ц а      Эталонный состав команд обработки данных

Арифметическо-логические операции с одним операндом

- ИНКР      Повысить операнд на 1 (инкрементировать)
- ДЕКР      Понизить операнд на 1 (декрементировать)
- НЕ      Логическое побитное отрицание операнда

Арифметическо-логические операции с двумя операндами

- СЛОЖ      Сложить операнд с содержимым аккумулятора
- ВЫЧ      Вычесть операнд из содержимого аккумулятора
- И      Логическая побитная операция И операнда с содержимым аккумулятора
- ИЛИ      Логическая побитная операция ИЛИ операнда с содержимым аккумулятора

Операция	Адрес операнда
----------	----------------

- Регистр *N*.      Адрес регистра *N*, содержащего операнд (прямая адресация)
- (Регистр *N*).      Адрес регистра *N*, содержащего адрес ячейки памяти, в которой находится операнд (косвенная адресация)



## Команды передачи данных

ПЕРЕ- ДАТЬ	Адрес назначе- ния	Адрес источни- ка
---------------	--------------------------	-------------------------

Операция передачи данных. Содержимое указанного адресом источника передается в указанный адрес назначения. После передачи по обоим адресам находится одинаковое содержимое. Источником и назначением могут быть как регистры, так и ячейки памяти

ЗАГРУ- ЗИТЬ	Регистр $N$	Констан- та
----------------	-------------	----------------

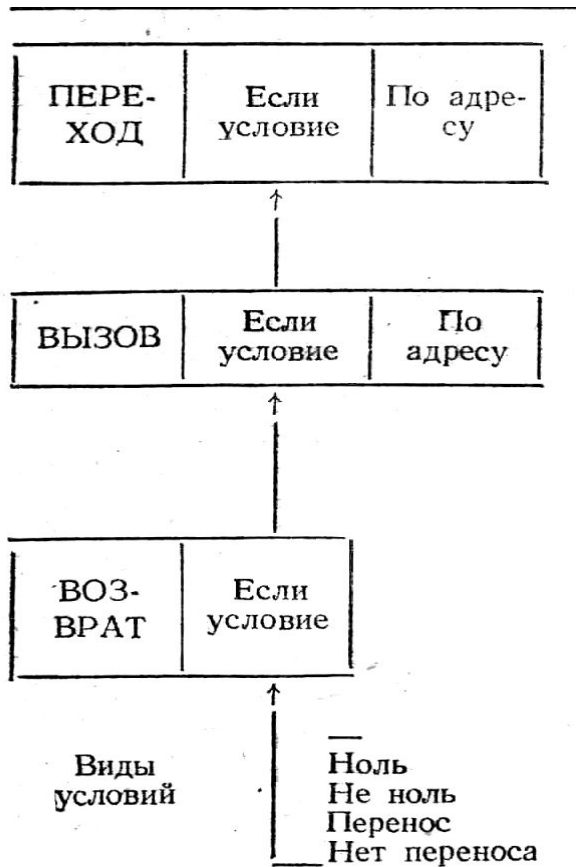
Операция загрузки. Регистр загружается словом, которое для постоянной величины указывается в команде. В регистре может находиться только двоичное слово 8 бит. Но в слове команды должны задаваться для упрощения десятичные числа, которые необходимо преобразовать в двоичные, прежде чем слово команды передано в память программы

ВВЕСТИ	Порт $N$
--------	----------

ВЫВЕС- ТИ	Порт $N$
--------------	----------

Операция ввода. Содержимое порта ввода  $N$  передается в аккумулятор

## Команды ветвления программ



Команда перехода. При выполнении условия в счетчик команд засылается адрес, и тем самым продолжается выполнение программы с имеющейся по этому адресу команды. В противном случае выполняется следующая за командой перехода команда

Команда вызова. Переход с обязательным возвратом CALL. При выполнении условия в счетчик команд загружается адрес после отправки на хранение в память состояния имевшегося в нем содержимого. В остальном действие то же, что при команде ПЕРЕХОД

Команда возврата RETURN. Последний сохраняемый в памяти состояния адрес возвращается в счетчик команд, продолжается выполнение прерванной командой вызова программы

## Типы команд микропроцессора

Если нужен переход, но следует обеспечить, чтобы после отработки подпрограммы последовал возврат в основную программу, такая задача решается парой команд: ВЫЗОВ подпрограммы (CALL, переход с обязательным возвратом) и ВОЗВРАТ в основную программу (RETURN).

Команда ВЫЗОВ по принципу действия соответствует команде перехода, но сохраняет содержимое счетчика команд, перезаписывая его в память состояния. В конце выполняемой подпрограммы должна стоять команда ВОЗВРАТ, действие которой состоит в том, что сохраненное в памяти состояние извлекается снова в счетчик команд и тем самым продолжается выполнение прерванной программы

## Команды управления

Разрешение прерывания	Прерывание разрешено
Запрет прерывания	Прерывание запрещено
HALT	Команда останов. Процессор пребывает в состоянии ожидания столь долго, пока не возникает прерывание
NOP	Команда без действия (операций)



# Структура командных слов различного типа

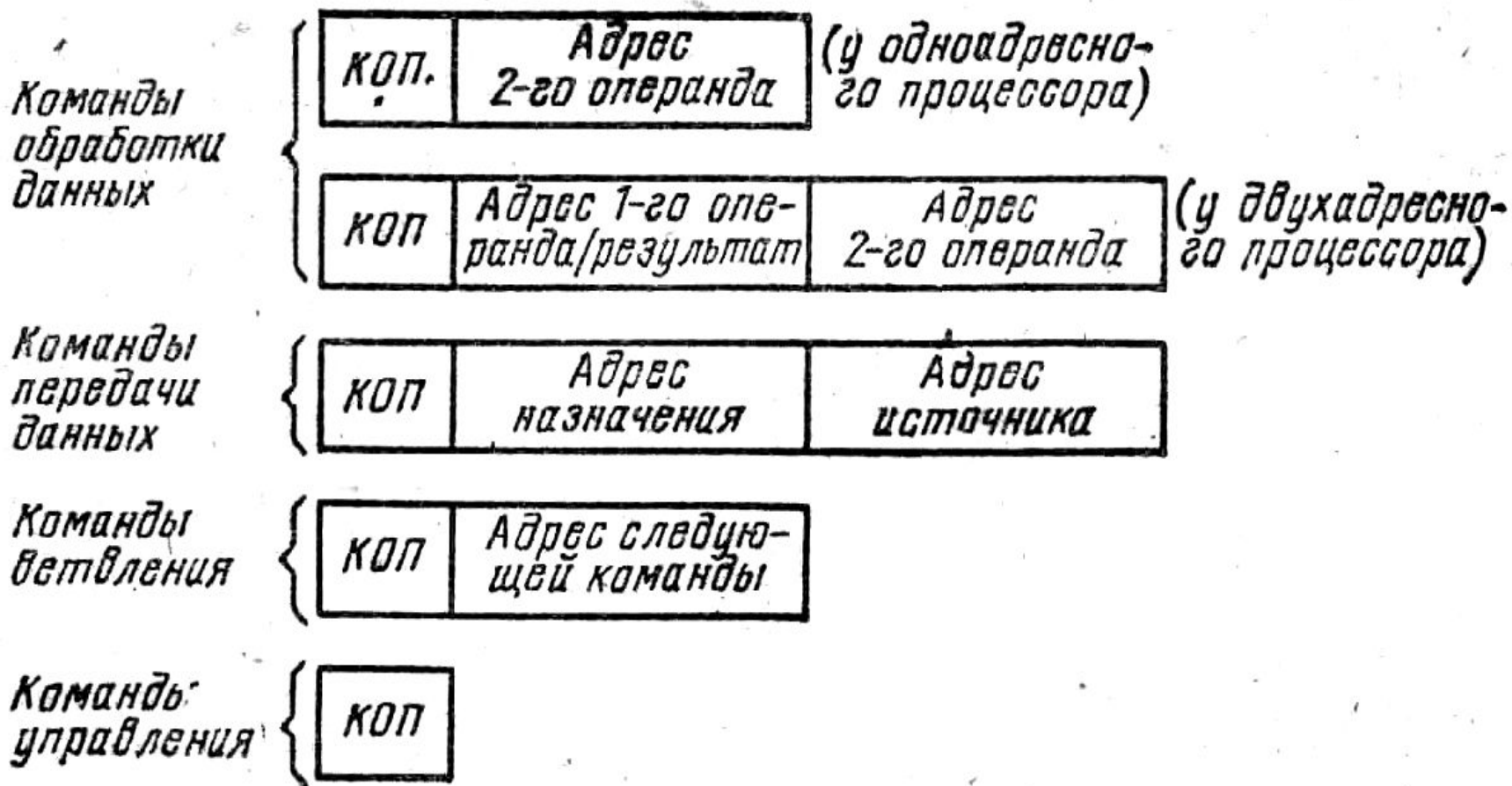


Рис. Структура слов команд

Совокупность микроопераций, выполняемых за один тактовый интервал, называют микрокомандой (МК).

Главная отличительная особенность МК (по отношению к команде) заключается в том, что МК состоит из сигналов (или непосредственно определяет их появление), непосредственно воздействующих на те или иные входы функциональных элементов (например, на затворы транзисторов).

Есть два способа организации выполнения микрокоманд (микроопераций):

- Аппаратный - рис а) на следующем слайде;
- Программный – рис б) там же.

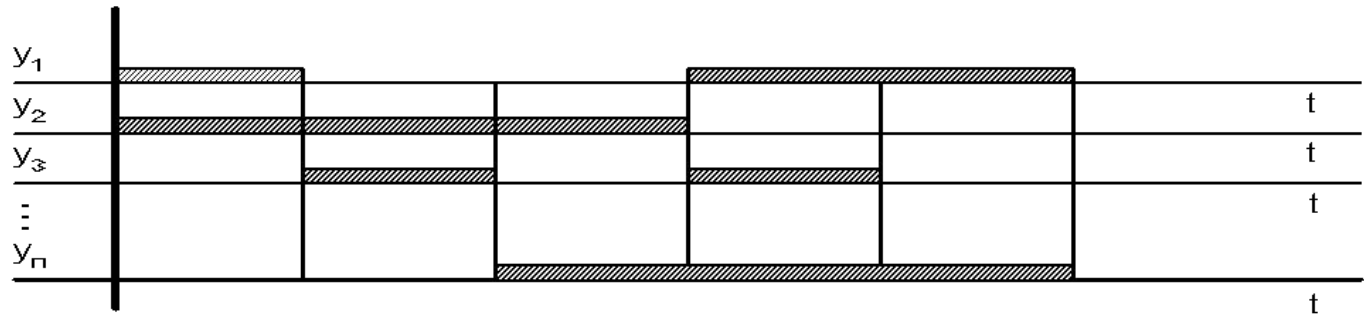
БУС – блок условий

БУП МК – блок управления микрокомандой

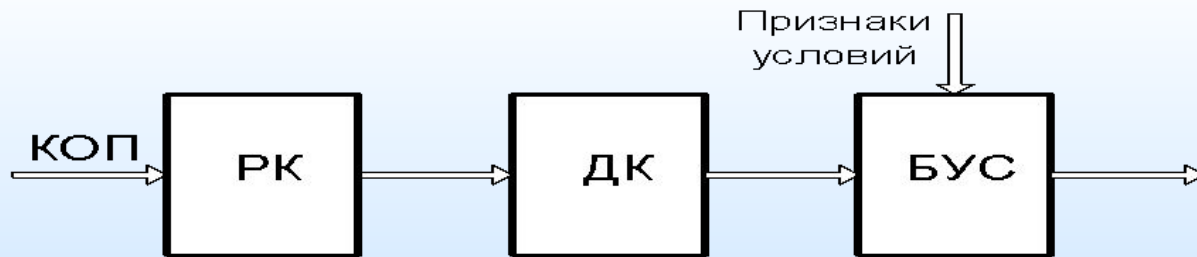
ПЗУ МК – постоянное запоминающее устройство микрокоманд

# Выполнение микроопераций (2 способа)

а)



б)



в)



Последовательность микрокоманд, обеспечивающая выполнение операции, задаваемой кодом команды, а точнее - ее операционной частью, т.е. кодом операции (КОП), называют микропрограммой (МПР).

Таким образом, процесс выполнения любой команды может быть описан некоторой микропрограммой и реализован за несколько машинных тактов, в каждом из которых выполняется одна микрокоманда, т. е. несколько (или одна) микроопераций. Управляющее устройство предназначено для формирования совокупности управляющих сигналов, обеспечивающих выполнение всех микроопераций, составляющих ту или иную команду.

### Два подхода к применению микропрограммирования

Микропрограммная реализация некоторого списка команд, в терминах которых будут затем записываться алгоритмы прикладных задач, т. е. создание программного уровня МПС. Объем работы небольшой – несколько сотен микропрограмм. Непосредственная микропрограммная реализация алгоритмов прикладных задач. Значительный объем работы, но максимальный выигрыш в производительности микропроцессорной системы.



Двум подходам к применению микропрограммирования соответствуют два способа микропрограммирования

- непосредственно в двоичных кодах микрокоманд
- с использованием языка символьной записи микрокоманд - языка микроассемблера, либо на языке высокого уровня (ЯВУ)

### Микропрограммирование на языке микроассемблера

Различные поля микрокоманд задаются их символическими обозначениями, которые в совокупности составляют символическую запись микрокоманды.

Микропрограмма в символической записи легче составляется, лучше читаема, проще для понимания и отладки. Это дает большую производительность микропрограммирования и улучшение качества микропрограмм.

### Микропрограммирование на языке микроассемблера

Разработчик при этом должен располагать микроассемблером — программой, производящей перевод символических обозначений микрокоманд в их двоичные коды. Такой микроассемблер представляет собой достаточно сложную программу.

### Микропрограммирование на языке микроассемблера

Для некоторых МК разработаны кроссовые системы микропрограммирования систем, строящихся на базе конкретного МК. Кроме задания символических обозначений полей микрокоманд подобные системы предоставляют обычно возможность задавать символические имена адресуемым в микрокоманде элементам памяти — регистрам, ячейкам основной и управляющей памяти.

### Микропрограммирование на языке микроассемблера

Однако, в ряде систем микропрограммирования привязка символических наименований к физическим адресам управляющей памяти, хранящей микропрограммы, возлагается на программиста.



**Спасибо за  
внимание!**