

# Микропроцессорные системы

---

КУРС ЛЕКЦИЙ

ЧУ ПО «СОЦИАЛЬНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»

ПРЕПОДАВАТЕЛЬ: БОРИСОВ АЛЕКСЕЙ АЛЬБЕРТОВИЧ

# Содержание

- Лекция 1. Классификация микропроцессоров
- Лекция 2. Организация памяти МП-систем
- Лекция 3. Виды статических ОЗУ
- Лекция 4. Классификация динамических ОЗУ
- Лекция 5. Структура ассоциативной памяти
- Лекция 6. Структура системы с основной и кэш-памятью
- Лекция 7. Алгоритмы замещения информации в кэш-памяти
- Лекция 8. Системы ввода/вывода
- Лекция 9. Структура внешнего устройства и модуля ввода-вывода
- Лекция 10. Архитектура микроконтроллеров
- Лекция 11. Типы памяти микроконтроллеров
- Лекция 12. Базовая архитектура микроконтроллера 8051
- Лекция 13. Питание микроконтроллеров
- Лекция 14. Система команд микроконтроллера семейства 8051
- Лекция 15. Связь микроконтроллера с внешней средой и временем
- Лекция 16. Система прерываний микроконтроллера 8051

## Содержание

- Лекция 17. Основные типы интерфейсов микроконтроллеров
- Лекция 18. Подключение клавиатуры и дисплея
- Лекция 19. Конвейеризация в микропроцессорных системах
- Лекция 20. Мультипроцессоры
- Лекция 21. Классификация компьютеров параллельного действия
- Лекция 22. Транспьютеры. Архитектура
- Лекция 23. Программирование транспьютеров
- Лекция 24. Обработка параллельных процессов в транспьютерах
- Лекция 25. Структура связи транспьютеров по внешнему каналу
- Лекция 26. Учебный лабораторный комплекс SDK-1.1
- Лекция 27. Карта портов ввода-вывода
- Лекция 28. Прерывания ADuC812
- Лекция 29. Основные этапы программирования стенда
- Лекция 30. Компилятор языка Си фирмы Keil Software
- Лекция 31. Инструментальная система для Win 9x/NT
- Лекция 32. Резидентный загрузчик HEX202



## **Список необходимой литературы для изучения курса**

- 1. Бурькова Е.В. Освоение микропроцессорной техники в формировании информационной компетентности студентов университета. – Челябинск, 2005.**
- 2. Бурькова Е.В. Моделирование микропроцессорных систем: методические указания к лабораторному практикуму – Оренбург, ГОУ ОГУ, 2007.**
- 3. Бурькова Е.В. Проектирование микропроцессорных систем: методические указания к курсовому проектированию – Оренбург, ГОУ ОГУ, 2008.**
- 4. Новиков Ю.В., Скоробогатов П.К. Основы микропроцессорной техники. – Москва. 2003.**
- 5. Предко М. Руководство по микроконтроллерам. - Москва, 2001**
- 6. Таненбаум Э. Архитектура компьютера.– Санкт-Петербург: ПИТЕР, 2003.**
- 7. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем.– Санкт-Петербург: ПИТЕР, 2006.**



# Лекция 1. Классификация микропроцессоров

## По технологии изготовления:

1. PMOS - P – Channel Metal Oxide Semiconductor – i4004, 8008;
2. NMOS - N – Channel Metal Oxide Semiconductor – i8080, i8085;
3. HMOS (N – Channel high-performance Metal Oxide Semiconductor) - 8086, 80186, 80286, 80386, 80486, Pentium;
4. CMOS (КМОП) – сигнальные и медийные МП - семейство TMS320C2X, cyrix6x86, Athlon , Pentium и т.д.

## По типу архитектуры:

1. однокристалльные МП;
2. однокристалльные микроконтроллеры;
3. разрядно – модульные МП;
4. CISC и RISC процессоры;
5. Транспьютеры.

## Классификация микропроцессоров (продолжение)

### По разрядности данных:

1. 2- разрядные – K589;
2. 4-разрядные – K145ИК18, КМ1804, I4004;
3. 8- разрядные – i8008, i8080, K580, K1816, i8051;
4. 16- разрядные – i8086, i8088, 80186, 80286;
5. 32- разрядные – i80386, i80486, Pentium;
6. 64- разрядные – AMD Opteron, PowerPC 970, Intel Xeon MP.

### По назначению:

1. универсальные;
2. специализированные МП.

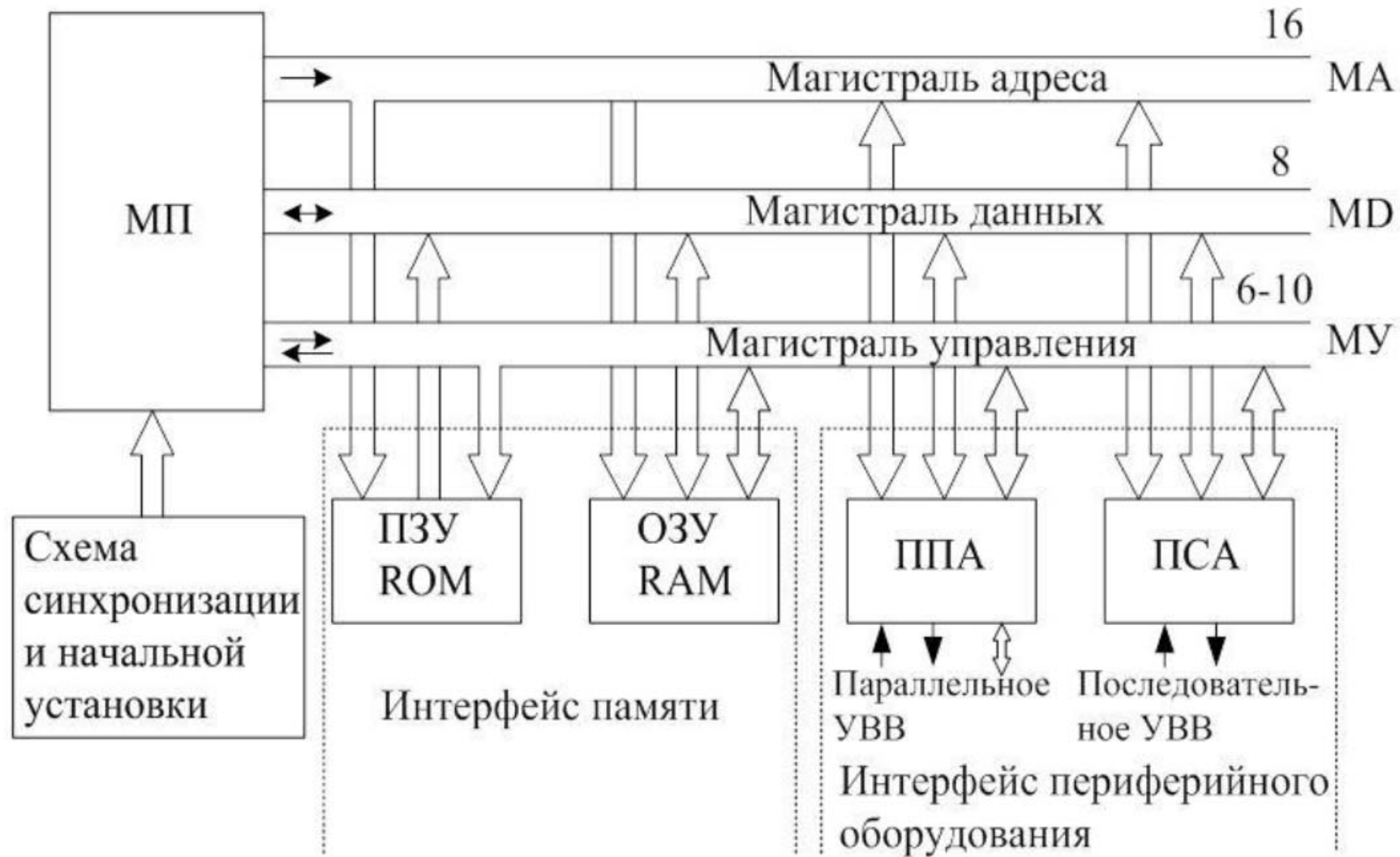
### По виду обрабатываемой информации:

1. цифровые;
2. аналоговые МП.

### По виду временной организации работы:

1. синхронные;
2. асинхронные.

# Структура микропроцессорной системы





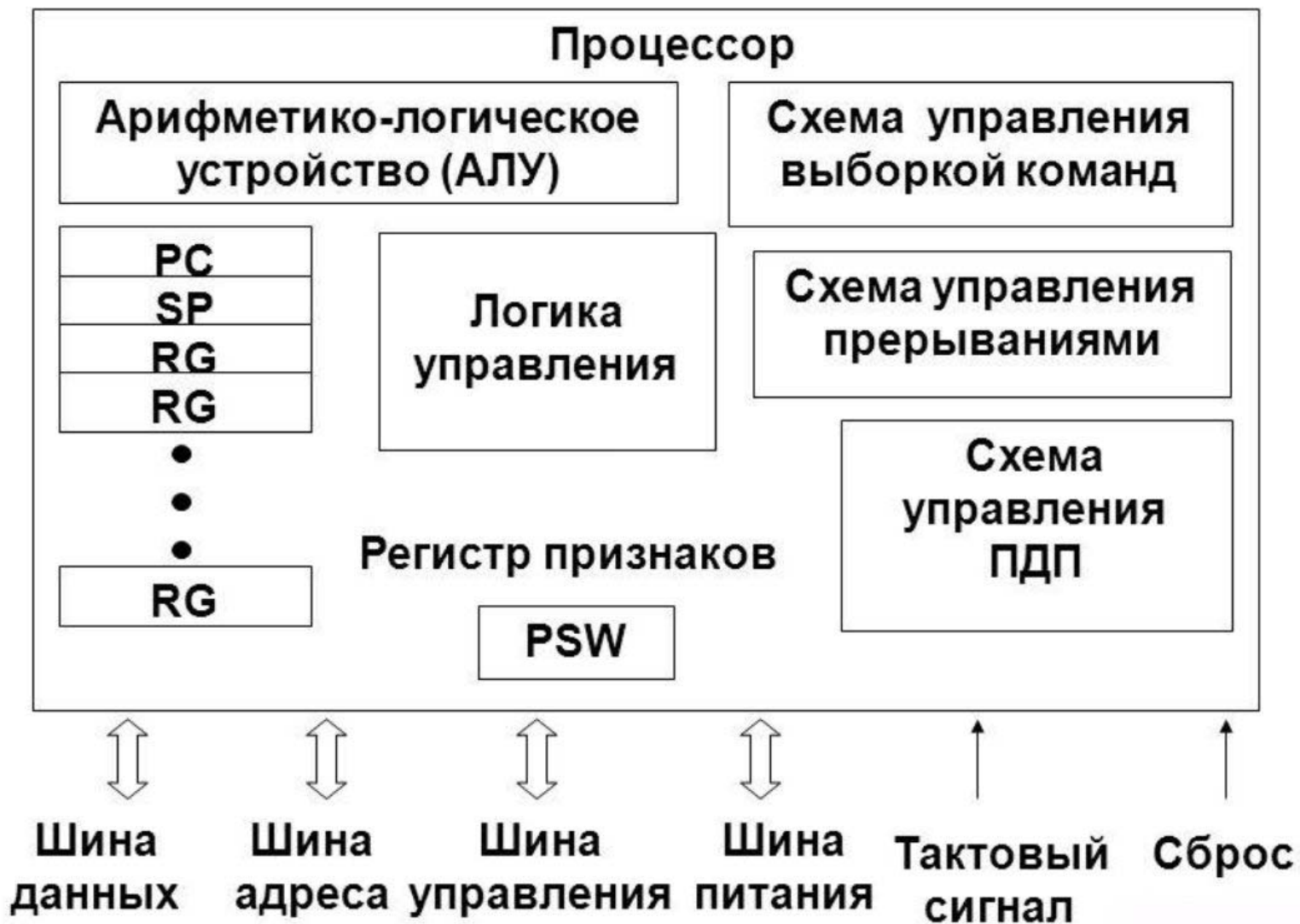
## Схема включения процессора



## Основные функции процессора:

1. выборка (чтение) выполняемых команд;
2. ввод (чтение) данных из памяти или УВВ;
3. вывод (запись) данных в память или УВВ;
4. обработка данных (операндов), в том числе арифметические операции над ними;
5. адресация памяти, т. е. задание адреса памяти, с которым будет производиться обмен;
6. обработка прерываний и режима прямого доступа к памяти (ПДП).

# Внутренняя структура микропроцессора





## Лекция 2. Организация памяти МП-систем

Память является необходимым компонентом микропроцессорной системы. Память используется как для хранения команд, так и данных. Функции памяти обеспечиваются запоминающими устройствами (ЗУ), предназначенными для фиксации, хранения и выдачи информации в процессе работы микропроцессорной системы.

При рассмотрении запоминающих устройств учитывают следующие характеристики:

1. место расположения;
2. емкость;
3. единица пересылки;
4. метод доступа;
5. быстродействие;
6. физический тип;
7. стоимость.

## Основные методы доступа к памяти:

1. последовательный,
2. прямой,
3. произвольный,
4. ассоциативный.

## Быстродействие ЗУ определяется параметрами:

1. время доступа;
2. длительность цикла памяти или период обращения;
3. скорость передачи.

$$T_N = T_A + \frac{N}{R}$$

где  $T_N$  - среднее время считывания или записи  $N$  битов;

$T_A$  - среднее время доступа;

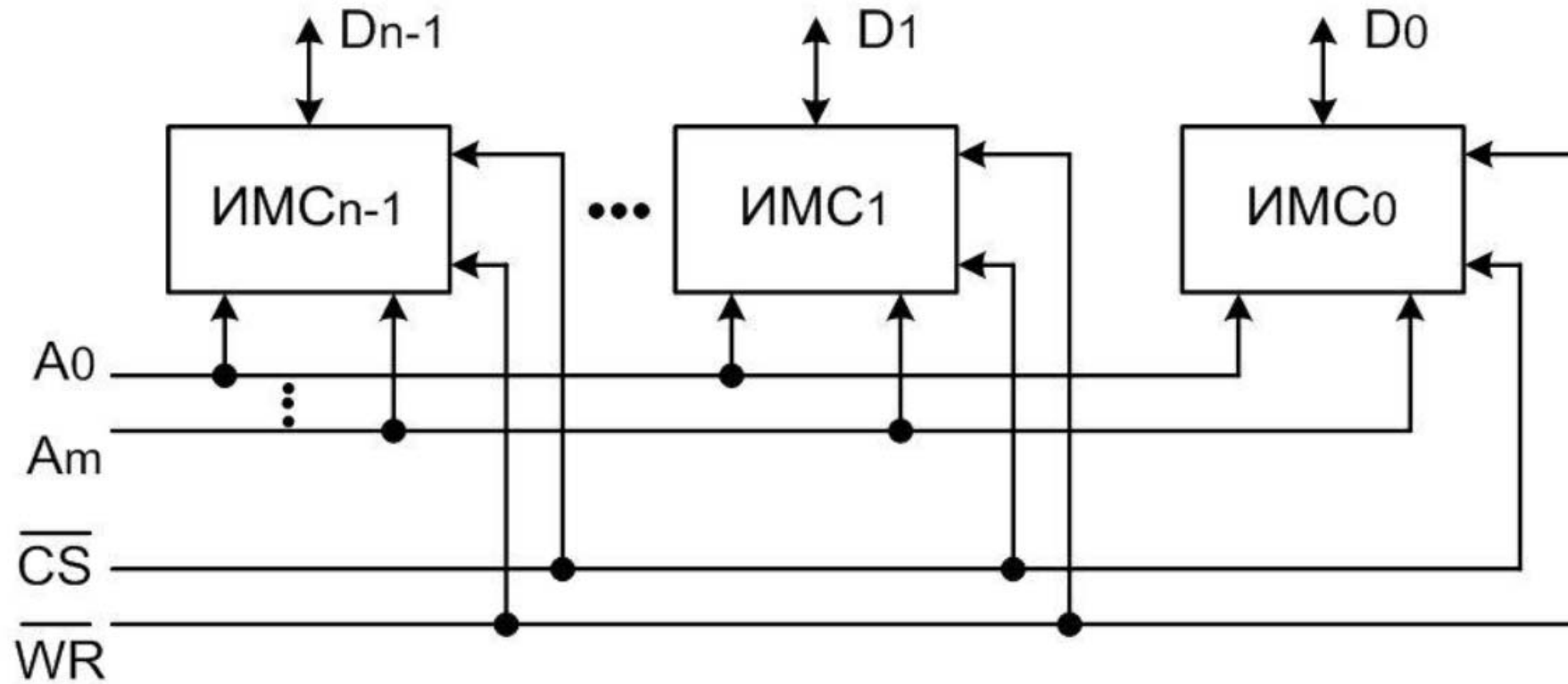
$R$  – скорость пересылки в битах в секунду.

# Иерархическая архитектура памяти

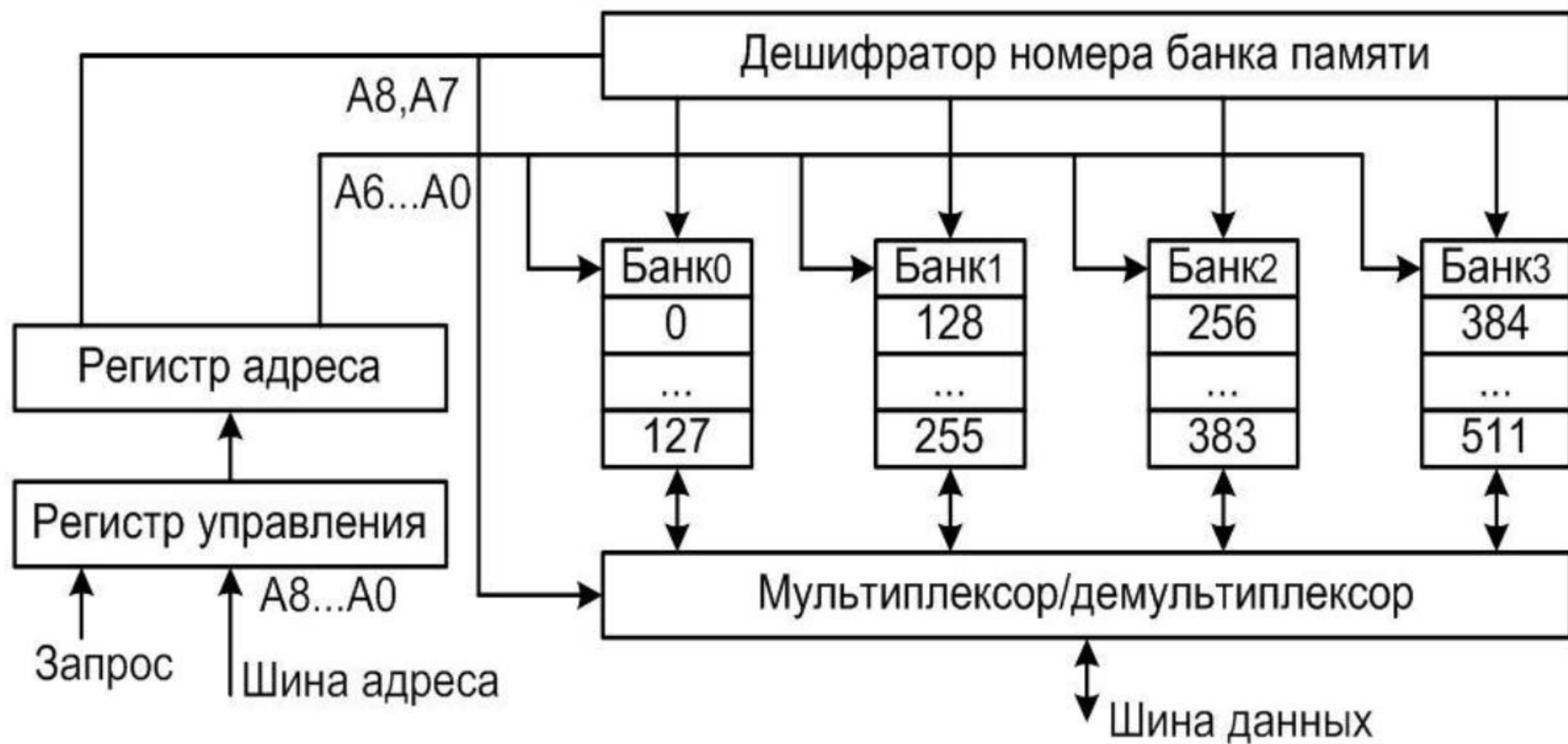




## Увеличение разрядности памяти

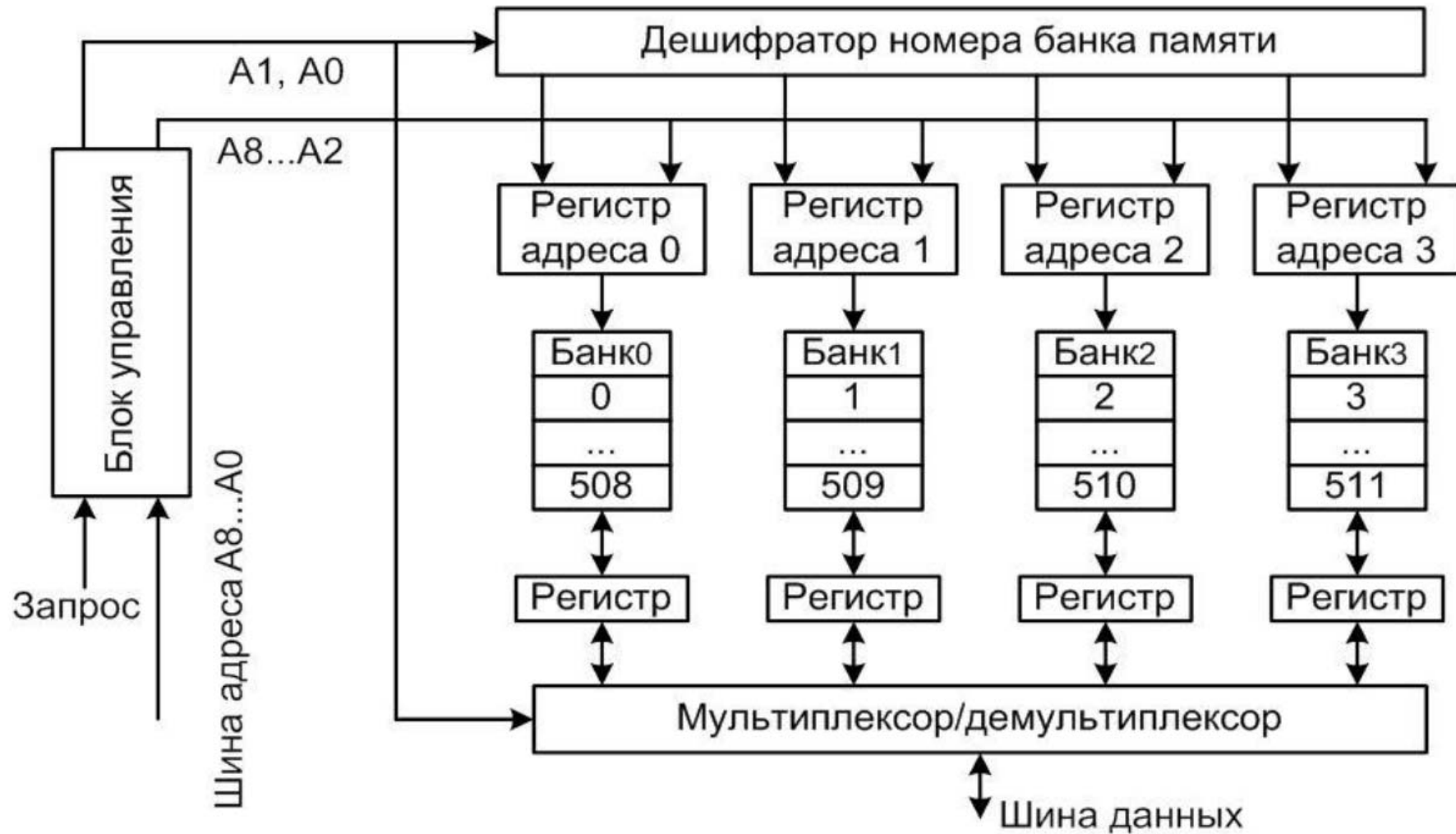


## Структура памяти на основе блочной схемы



Рассмотрим блочную структуру памяти на примере памяти емкостью 512 слов, построенной из четырех банков по 128 слов в каждом.

## Блочная память с чередованием адресов по циклической схеме

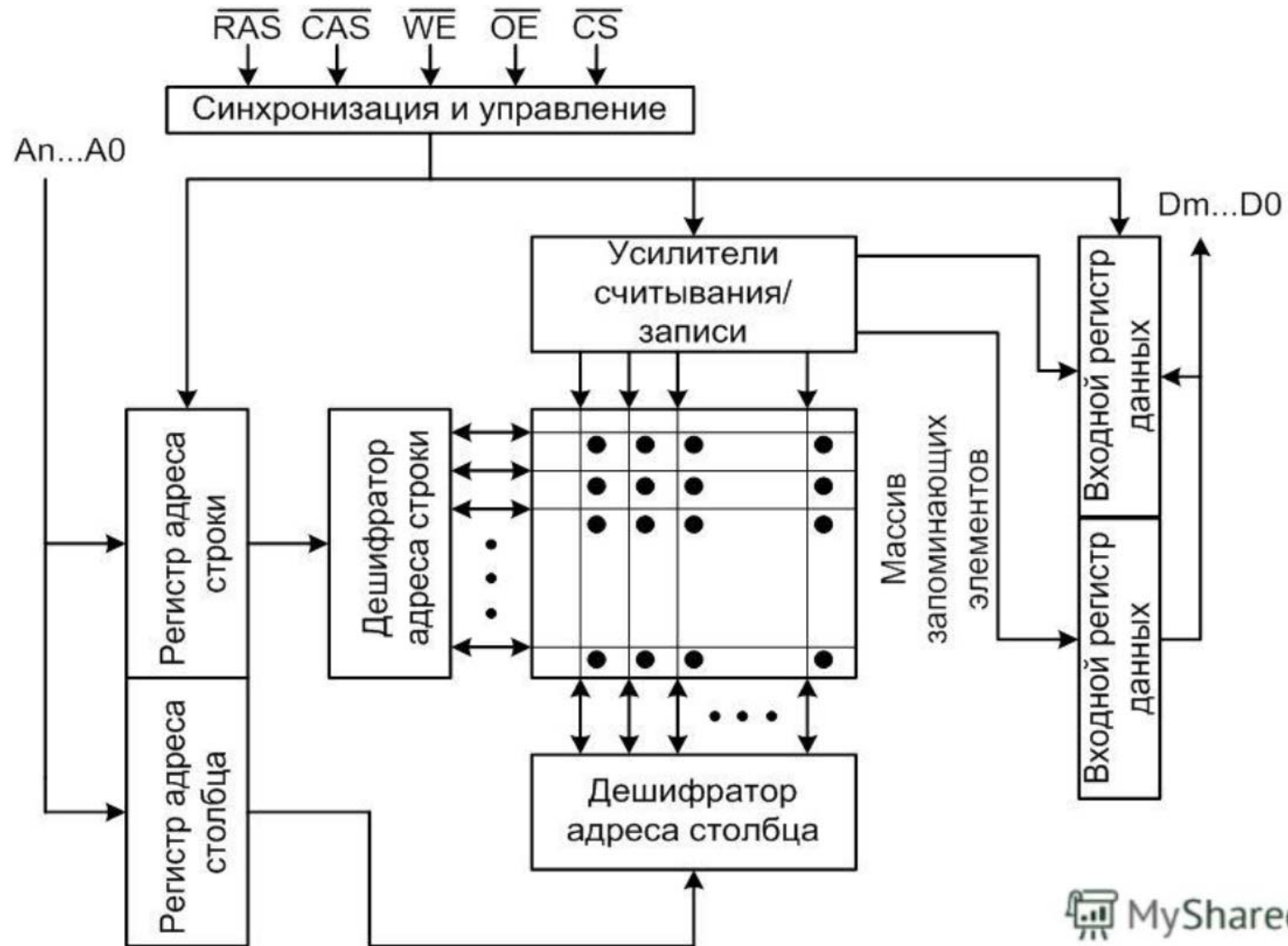


## Блочно-циклическая схема расслоения памяти

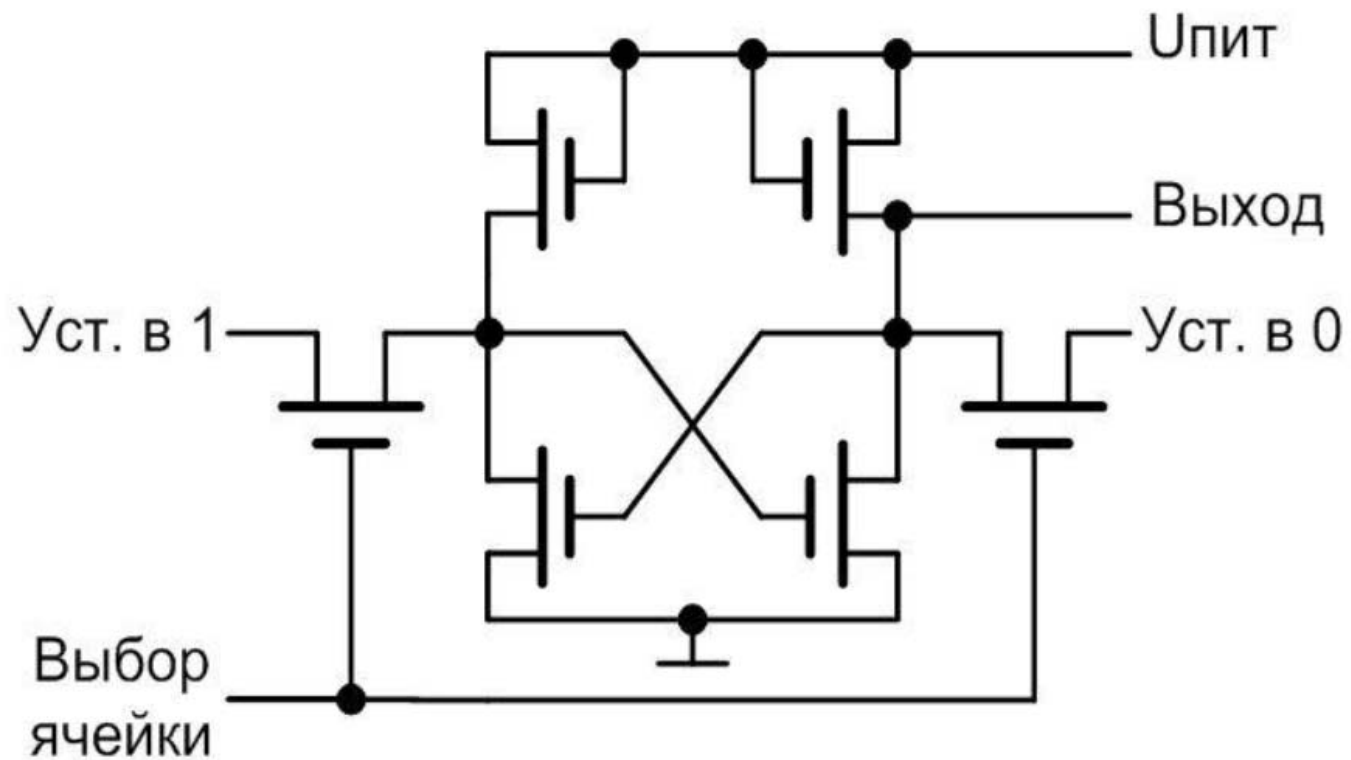




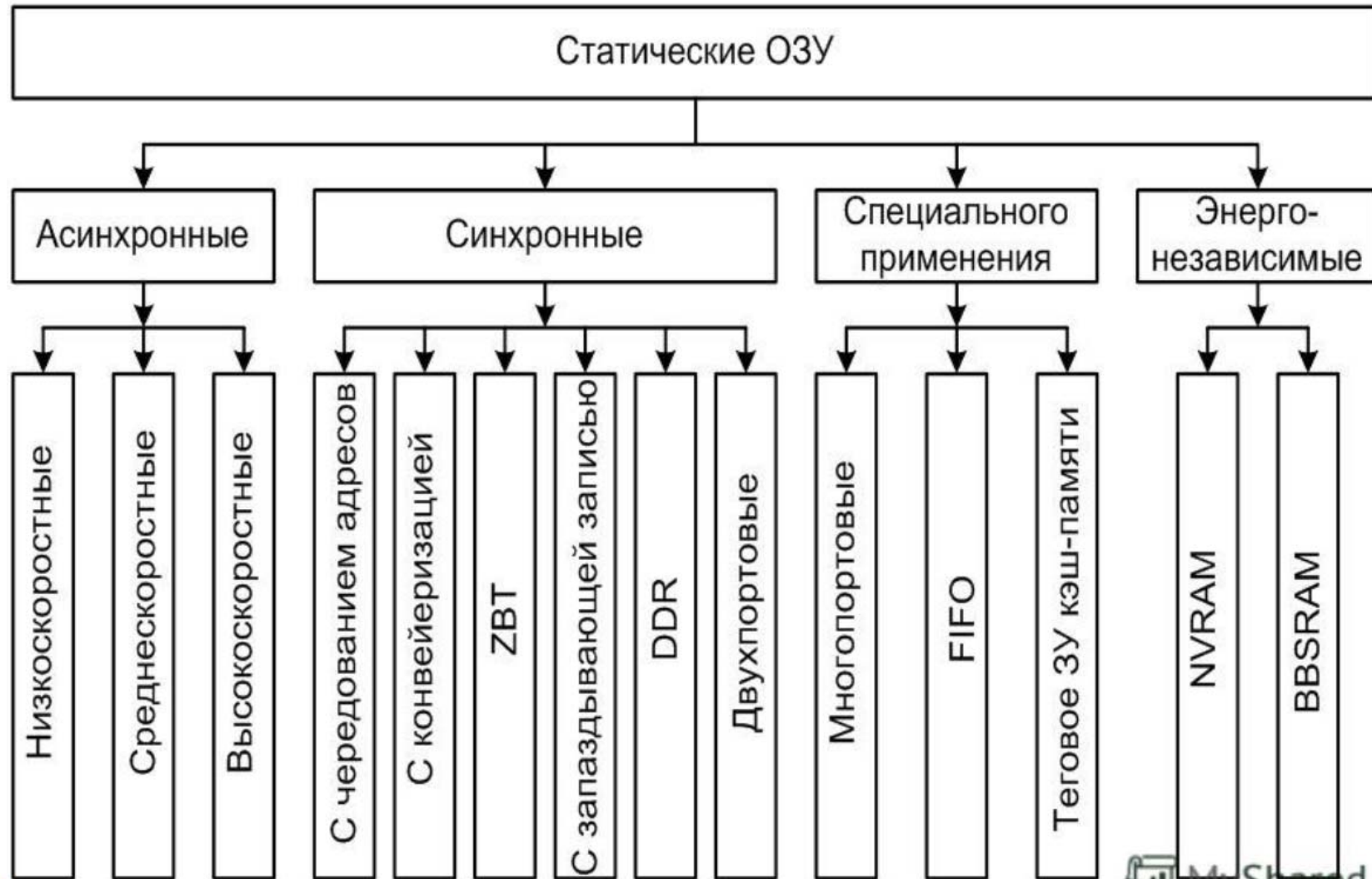
# Структура микросхемы памяти



## Запоминающий элемент статического ОЗУ



# Лекция 3. Виды статических ОЗУ



## **Асинхронные статические ОЗУ:**

время доступа составляло 15-20 нс (в лучшем случае — 12 нс)

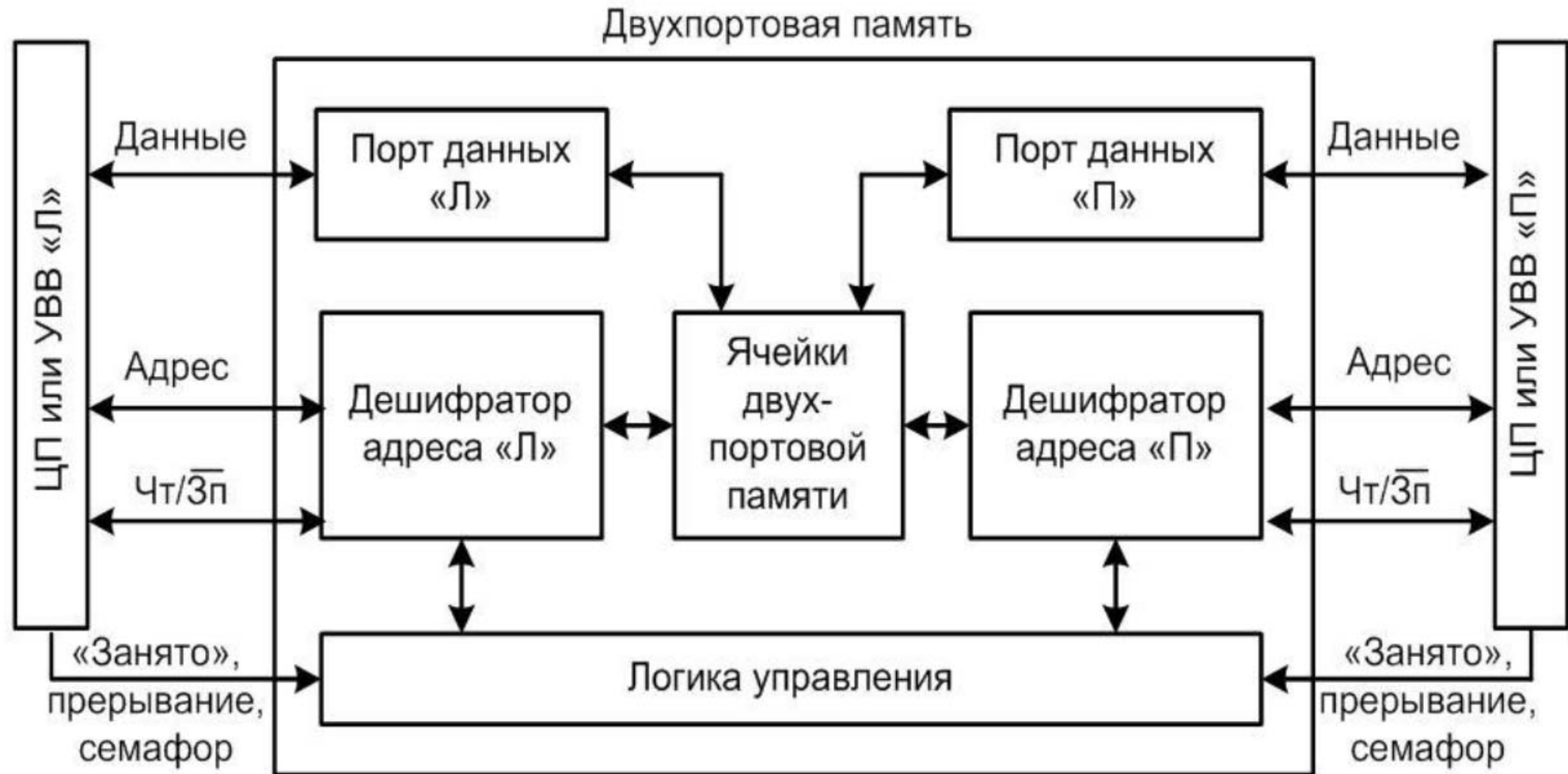
**Синхронные статические ОЗУ:SSRAM** обеспечивает работу в пакетном режиме с формулой 3-1 -1 -1, но лишь до определенных значений тактовой частоты шины. При более высоких частотах формула изменяется на 3-2-2-2.

**PB SRAM.** Начиная с Pentium II, взамен SSRAM оснащаются статической оперативной памятью с пакетным конвейерным доступом (PB SRAM — Pipelined Burst SRAM). Время доступа к PB SRAM составляет от 4,5 до 8 нс, при этом формула 3-1-1-1.

Компания IDT (Integrated Device Technology) предложила новую технологию, получившую название **ZBT SRAM** (Zero Bus Turnaround) — нулевое время переключения шины.



# Структура двухпортового ОЗУ



## Энергонезависимые оперативные запоминающие устройства

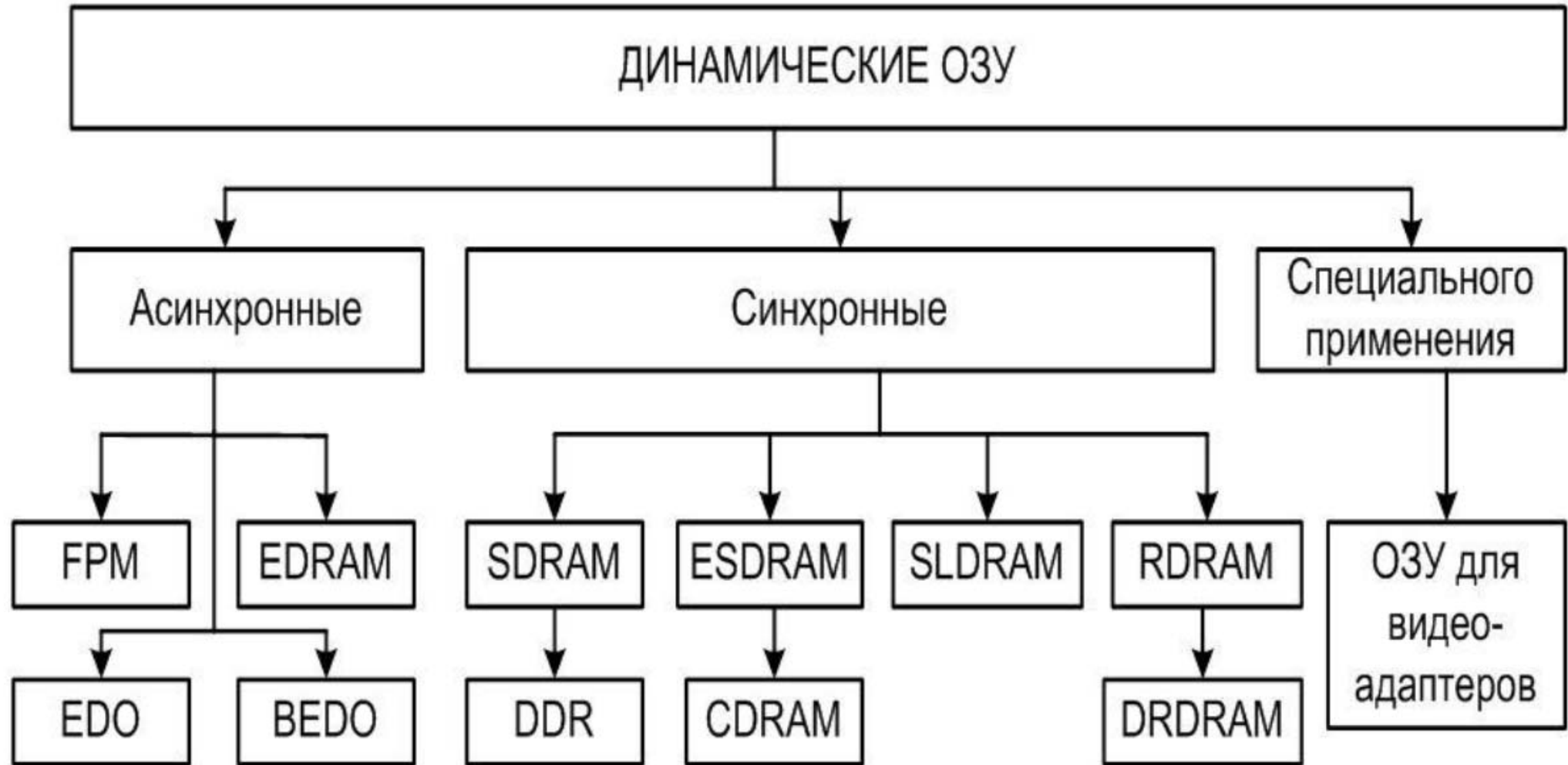
1. **NVRAM** — Non-Volatile RAM, разработана компанией Simtec;
2. **BBSRAM** - Battery-Back SRAM;
3. **FRAM** - Ferroelectric RAM — ферроэлектрическая память, разработана компанией Ramtron. Количество циклов перезаписи для FRAM обычно составляет 10 млрд.;
4. **Магнитная нанопамять;**
5. **Память на основе фазовых состояний PCM;**
6. **Магниторезистивная память MRAM.**

# Сравнительные характеристики основных типов оперативной памяти

*Сравнительные характеристики основных типов памяти*

	MRAM	SRAM	DRAM	FLASH	FRAM
Скорость чтения	Высокая	Самая высокая	Средняя	Высокая	Высокая
Скорость записи	Высокая	Самая высокая	Средняя	Низкая	Средняя
<u>Масштабируемость</u>	Хорошая	Хорошая	<u>Огранич.</u>	<u>Огранич.</u>	<u>Огранич.</u>
Плотность ячеек	Средняя/ высокая	Низкая	Высокая	Средняя	Средняя
<u>Энергонезависимость</u>	Да	Нет	Нет	Да	Да
Число циклов стирания/записи	<u>Неогранич.</u>	<u>Неогранич.</u>	<u>Неогранич.</u>	<u>Огранич.</u>	<u>Огранич.</u>
Ток утечки ячеек	Низкий	Низкий/ Высокий	Высокий	Низкий	Низкий
Возможность работы при низких напряжениях	Да	Да	<u>Огранич.</u>	<u>Огранич.</u>	<u>Огранич.</u>
Сложность производства	Средняя	Низкая	Средняя	Средняя	Средняя

# Лекция 4. Классификация динамических ОЗУ





## Асинхронная динамическая память

**Микросхемы DRAM** - формула 5-5-5-5. Микросхемы данного типа могли работать на частотах до 40 МГц и из-за низкого быстродействия (время доступа составляло около 120 нс) просуществовали недолго.

**Микросхемы FPM DRAM.** Микросхемы динамического ОЗУ, реализующие режим FPM, также относятся к ранним типам DRAM. Схема чтения для FPM DRAM описывается формулой 5-3-3-3 (всего 14 тактов). Время доступа до 60 нс.

**Микросхемы EDO DRAM.** с гиперстраничным режимом доступа (HPM, Hyper Page Mode), более известные как EDO (Extended Data Output — расширенное время удержания данных на выходе) формула 5-2-2-2, что на 20% быстрее, чем у FPM. Время доступа составляет порядка 30-40 нс. Максимальная частота системной шины для микросхем EDO DRAM не превышает 66 МГц.

**Микросхемы BEDO DRAM.** Технология EDO была усовершенствована компанией VIA Technologies. Новая модификация EDO известна как BEDO (Burst EDO — пакетная EDO). Формула 5-1-1-1.

**Микросхемы EDRAM.** Более быстрая версия DRAM была разработана подразделением фирмы Ramtron — компанией Enhanced Memory Systems. Технология реализована в вариантах FPM, EDO, Shared

## Синхронные динамические ОЗУ

**Микросхемы SDRAM.** Аббревиатура SDRAM (Synchronous DRAM — синхронная DRAM) используется для обозначения микросхем «обычных» синхронных динамических ОЗУ. Кардинальные отличия SDRAM от рассмотренных выше асинхронных динамических ОЗУ можно свести к четырем положениям:

- синхронный метод передачи данных на шину;
- конвейерный механизм пересылки пакета;
- применение нескольких внутренних банков памяти;
- передача части функций контроллера памяти логике

микросхемы.

Несмотря на то, что формула для этого типа динамической памяти такая же, что и у BEDO, способность работать на более высоких частотах приводит к тому, что SDRAM с двумя банками при тактовой частоте шины 100 МГц по производительности может почти вдвое превосходить память типа BEDO.



## Синхронные динамические ОЗУ

**Микросхемы DDR SDRAM.** Важным этапом в дальнейшем развитии технологии SDRAM стала DDR SDRAM (Double Data Rate SDRAM - SDRAM с удвоенной скоростью передачи данных).

Существует несколько спецификаций DDR SDRAM, в зависимости от тактовой частоты системной шины: DDR266, DDR333, DDR400, DDR533. Пиковая пропускная способность микросхемы памяти спецификации DDR333 составляет 2,7 Гбайт/с, а для DDR400 — 3,2 Гбайт/с.

**Микросхемы RDRAM, DRDRAM.** (Rambus Direct RAM). Существует несколько разновидностей этой технологии: Base, Concurrent и Direct. Во всех тактирование ведется по обоим фронтам синхросигналов (как в DDR), благодаря чему результирующая частота составляет соответственно 500-600, 600-700 и 800 МГц.

**Микросхемы SDRAM** (Sync Link DRAM).

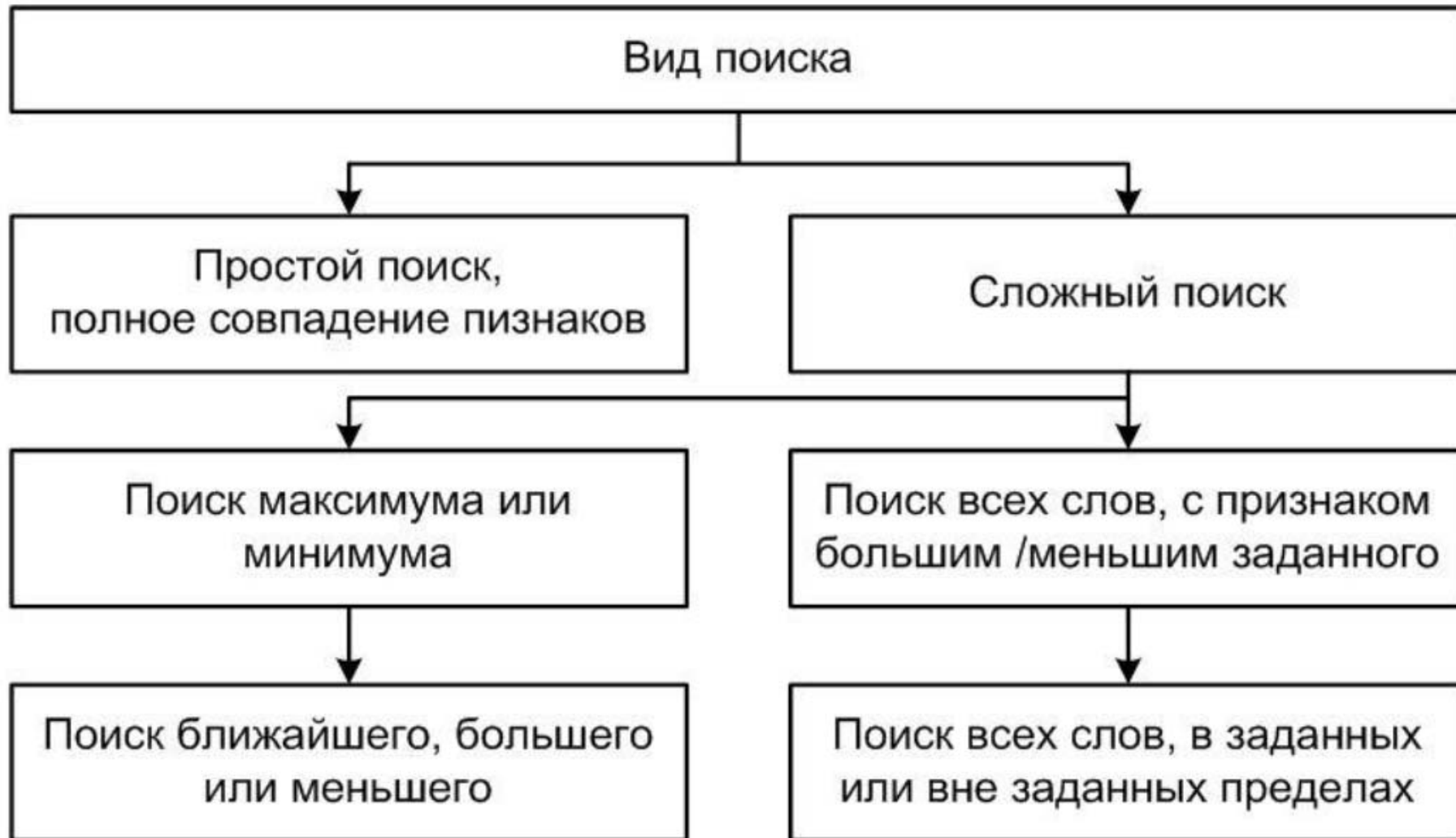
**Микросхемы ESDRAM.**

**Микросхемы CDRAM.**

# Лекция 5. Структура ассоциативной памяти

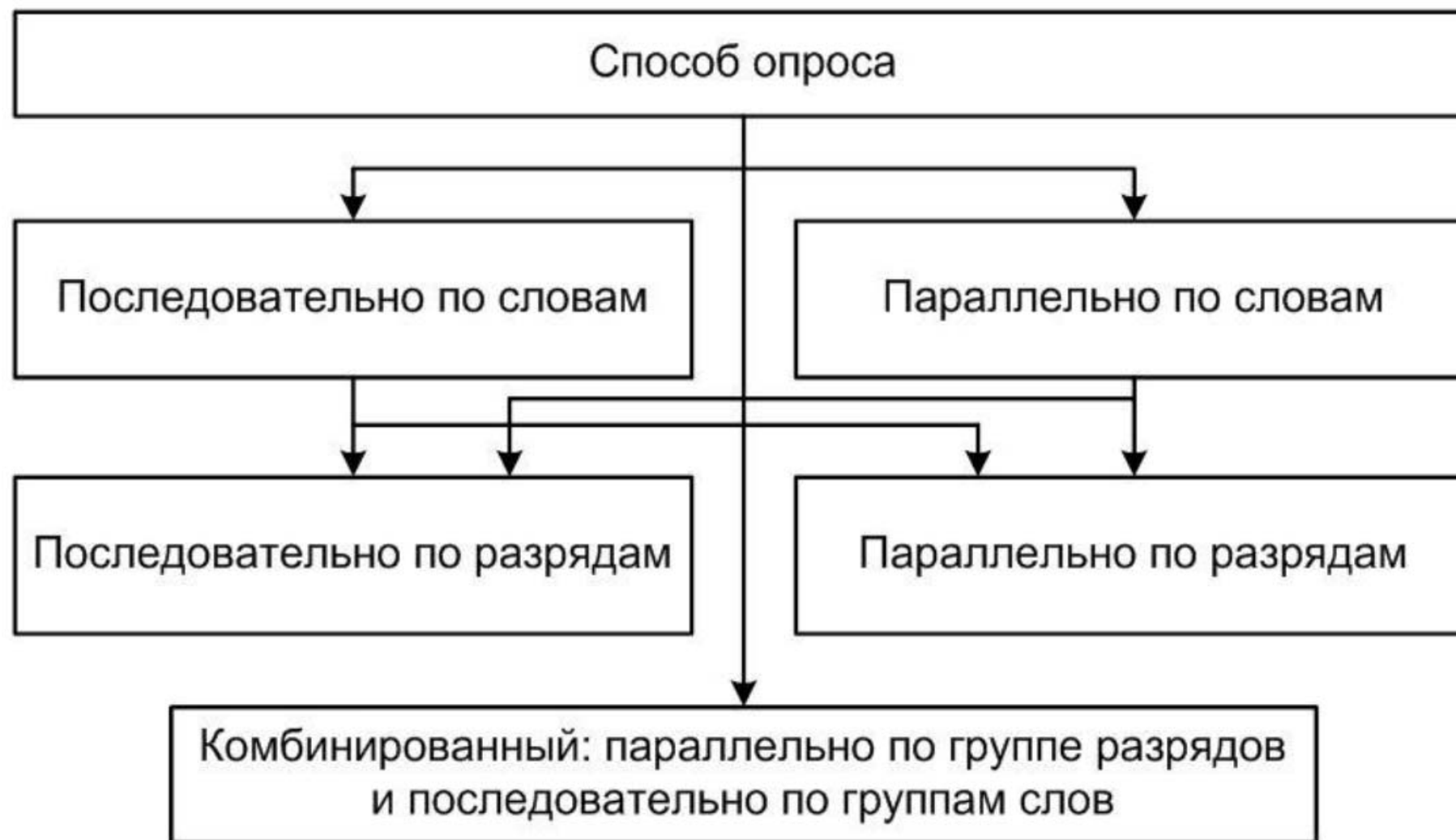


# Классификация ассоциативных ЗУ по виду поиска






# Классификация ассоциативных ЗУ по способу опроса



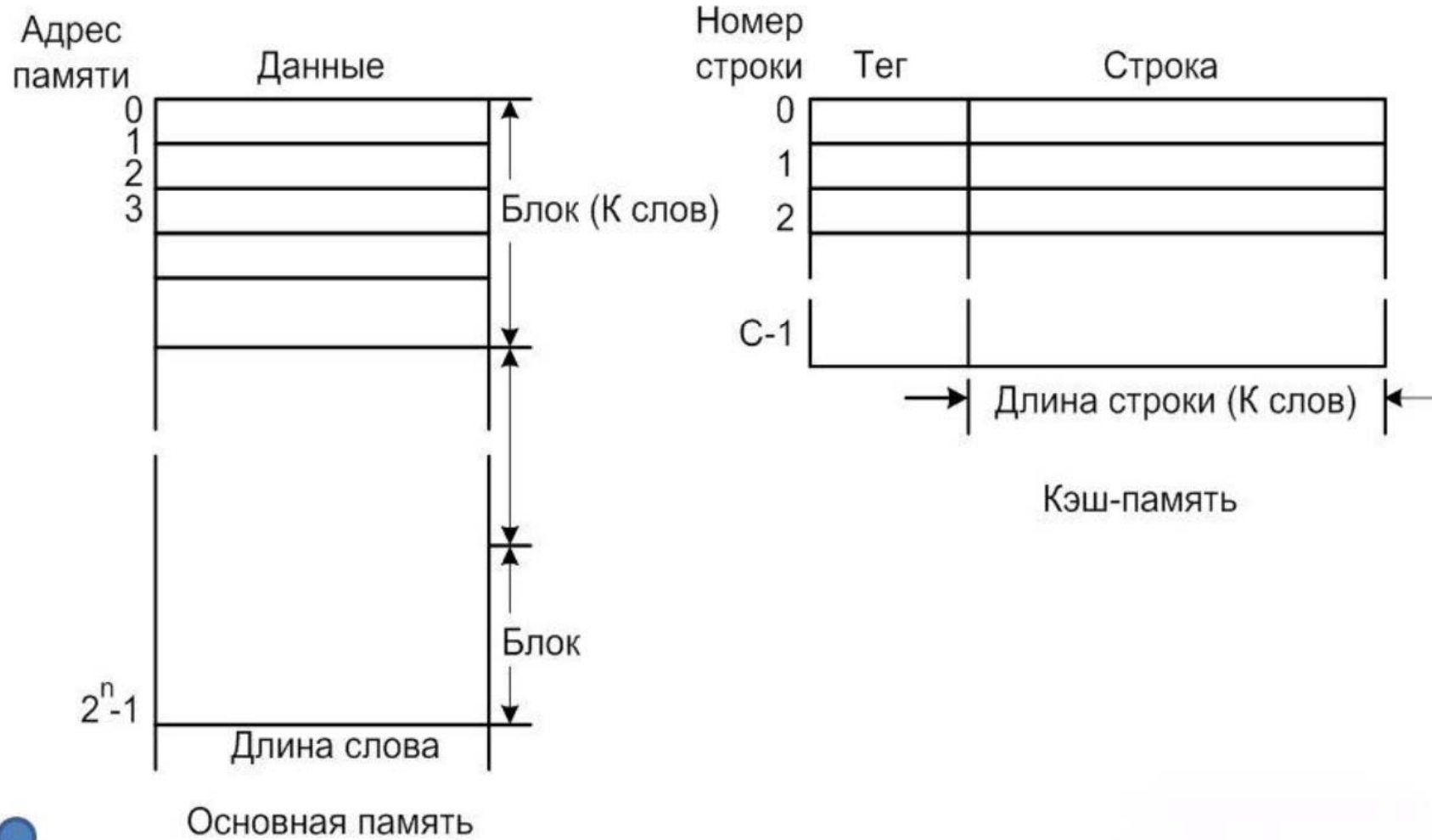
# Классификация ассоциативных ЗУ по способу выборки



Цепь очередности реализуется с помощью достаточно сложного устройства, где фиксируются слова, образующие многозначный ответ. Цепь очередности позволяет производить считывание слов в порядке возрастания номера ячейки АЗУ независимо от величины ассоциативных признаков.

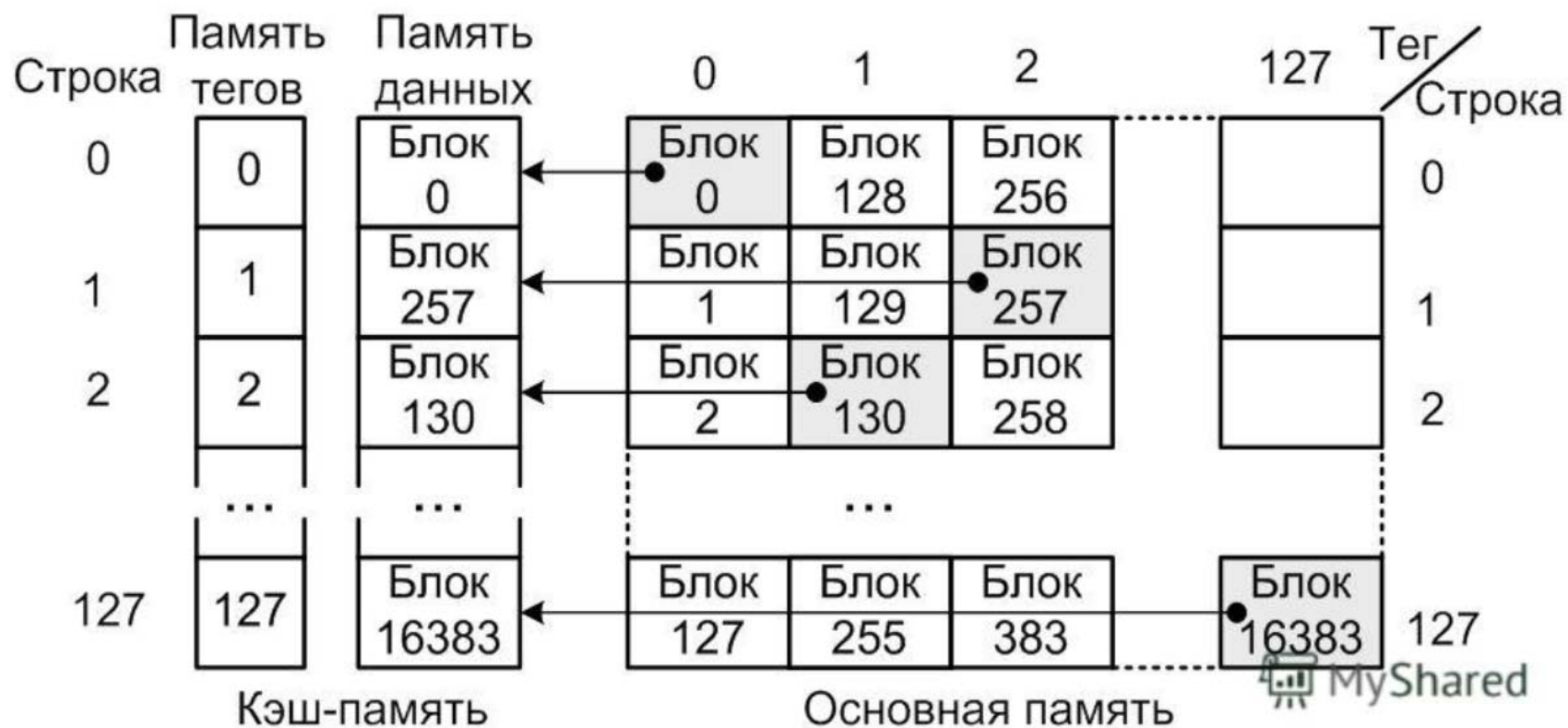
При алгоритмическом способе извлечения многозначного ответа выборка производится в результате серии опросов. Серия опросов может формироваться путем упорядочивания тех разрядов, которые были замаскированы и не участвовали в признаке поиска.  MyShared

# Лекция 6. Структура системы с основной и кэш-памятью



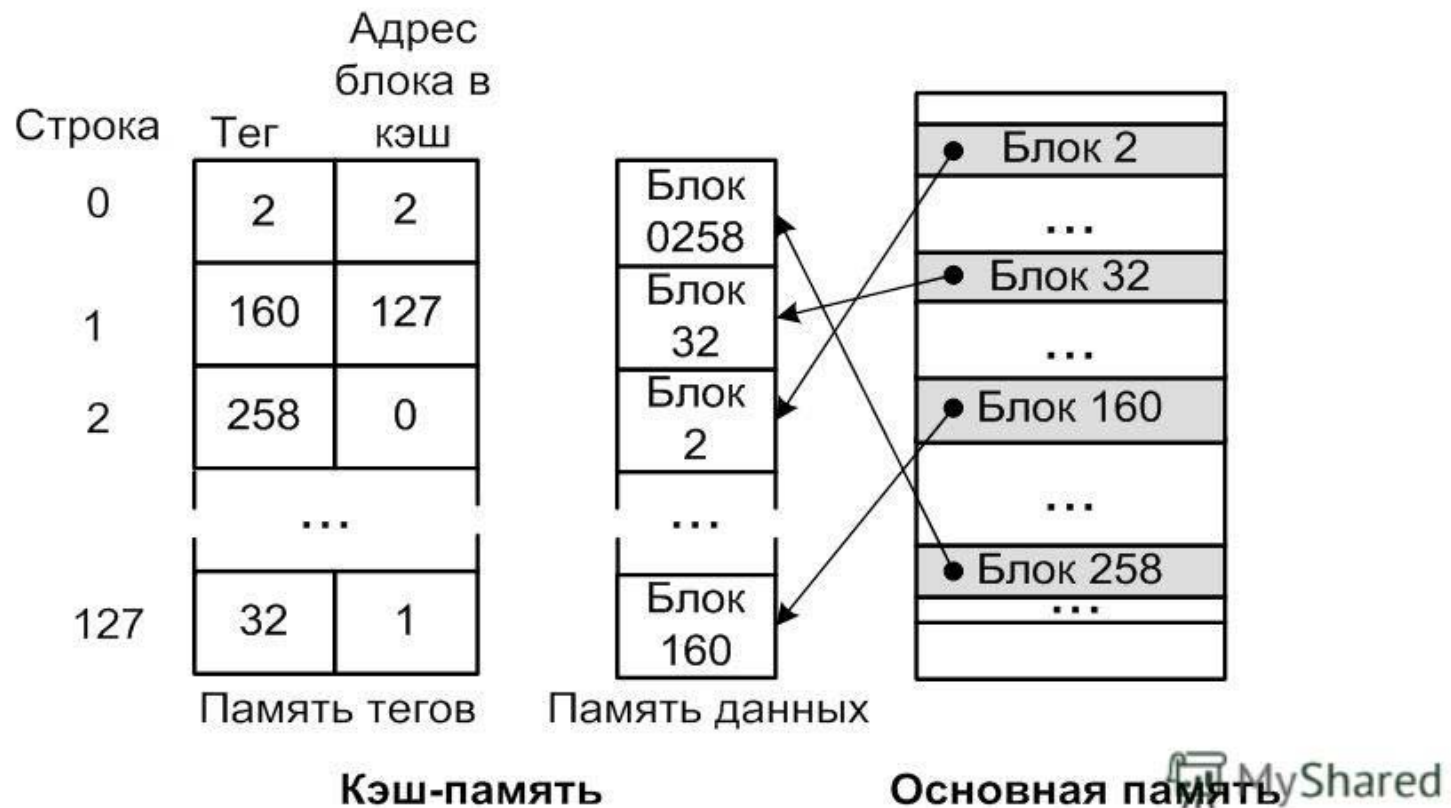
## Прямое отображение основной памяти

При прямом отображении адрес строки  $i$  кэш-памяти, на которую может быть отображен блок из ОП, определяется выражением:  $i = j \bmod m$ , где  $m$  — общее число строк в кэш-памяти. В нашем примере  $i = j \bmod 128$ , где  $i$  может принимать значения от 0 до 127, а адрес блока  $j$  — от 0 до 16 383. Иными словами, на строку кэша с номером  $i$  отображается каждый 128-й блок ОП, если отсчет начинать с блока, номер которого  $i$ .



При реализации такого отображения 14-разрядный адрес блока основной памяти условно разбивается на два поля. Логика кэш-памяти интерпретирует эти 14 бит как 7-разрядный тег и 7-разрядное поле строки. Поле строки указывает на одну из  $128 = 2^7$  строк кэш-памяти, а именно на ту, куда может быть отображен блок с заданным адресом.

### Полностью ассоциативное отображение





## Множественно-ассоциативное отображение

Кэш-память разбивается на  $v$  подмножеств (в дальнейшем будем называть их модулями), каждое из которых содержит  $k$  строк. Зависимость между модулем и блоками основной памяти такая же, как и при прямом отображении: на строки, входящие в модуль  $i$ , могут быть отображены только вполне определенные блоки основной памяти, в соответствии с соотношением  $i = j \bmod v$ , где  $j$  — адрес блока основной памяти.

### Пример четырехходовой кэш-памяти



## Множественно-ассоциативное отображение

Блоки 0, 32, 64, 96 и т. д. основной памяти отображаются на модуль с номером 0; блоки 1, 33, 65, 97 и т. д. отображаются на модуль 1 и т. д. Любой из блоков в последовательности может быть загружен в любую из четырех строк соответствующего модуля.

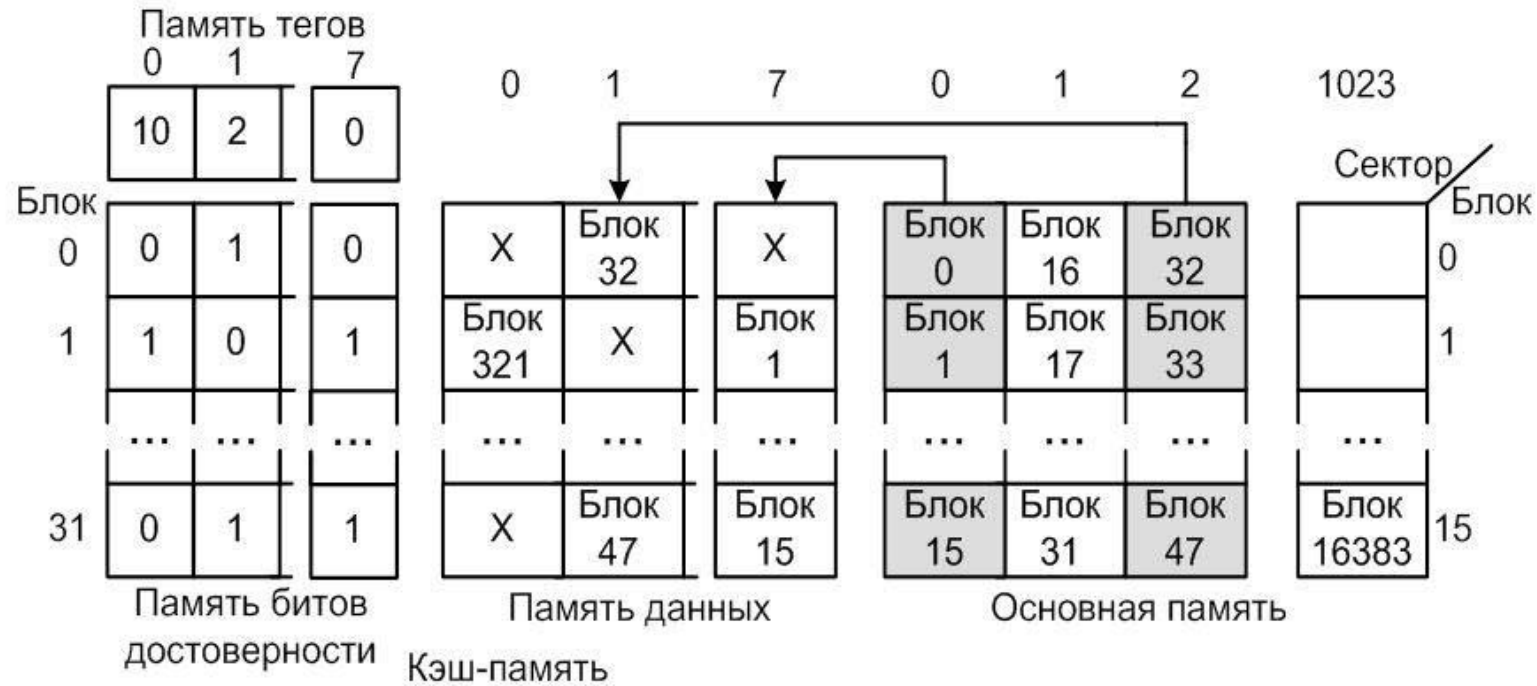
При такой постановке роль тега выполняют 9 старших разрядов адреса блока основной памяти, в которых содержится порядковый номер блока в последовательности блоков, отображаемых на один и тот же модуль кэш-памяти.

В предельных случаях, когда  $v = m$ ,  $k = 1$ , множественно-ассоциативное отображение сводится к прямому, а при  $v = 1$ ,  $k = m$  — к ассоциативному. Наиболее общий вид организации множественно-ассоциативного отображения — использование двух строк на модуль ( $v = m/2$ ,  $k = 2$ ).

Четырехвходовая множественно-ассоциативная кэш-память ( $v = m/4$ ,  $k = 4$ ) дает дополнительное улучшение за сравнительно небольшую дополнительную цену. Дальнейшее увеличение числа строк в модуле не дает существенного эффекта.



## Отображение секторов



Здесь сектор состоит из  $16 = 2^4$  блоков по 16 слов, и основная память содержит  $1024 = 2^{10}$  сектора. В 14-разрядном адресе блока основной памяти старшие 10 разрядов показывают номер сектора, а младшие 4 — номер блока внутри сектора. В свою очередь, кэш-память состоит из  $8 = 2^3$  секторов, и 7-разрядный адрес строки в кэш включает в себя адрес сектора кэш (3 старших разряда) и номер блока внутри сектора (4 младших разряда).

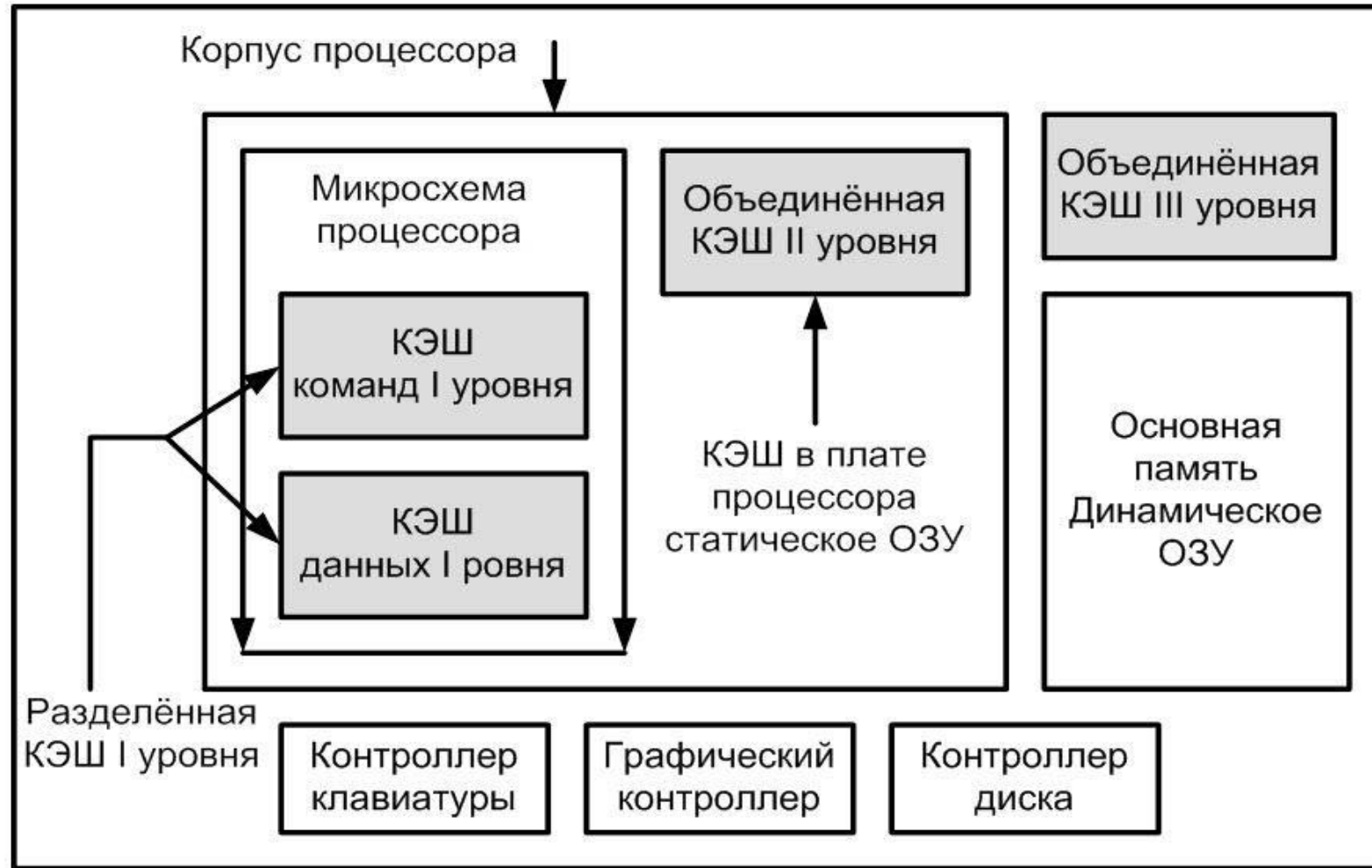


## Лекция 7. Алгоритмы замещения информации в кэш-памяти

1. Алгоритм замещения на основе *наиболее давнего использования* (LRU — Least Recently Used). Замещается та строка кэш-памяти, к которой дольше всего не было обращения. Проведившиеся исследования показали, что алгоритм LRU работает достаточно хорошо в сравнении с оптимальным алгоритмом.
  2. Алгоритм, работающий по принципу «*первый вошел, первый вышел*» (FIFO — First In First Out). Заменяется строка, дольше всего находившаяся в кэш-памяти.
  3. Замена наименее часто использовавшейся строки (LFU — Least Frequently Used). Заменяется строка в кэш-памяти, к которой было меньше всего обращений. Связывают каждую строку со счетчиком обращений, к содержимому которого после каждого обращения добавляется единица. Главным претендентом на замещение является строка, счетчик которой содержит наименьшее число.
  4. Произвольный выбор строки для замены. Замещаемая строка выбирается случайно. Реализовано с помощью счетчика, содержимое которого увеличивается на единицу с каждым тактовым импульсом, независимо от того, имело место попадание или промах.
- Значение в счетчике определяет заменяемую строку.



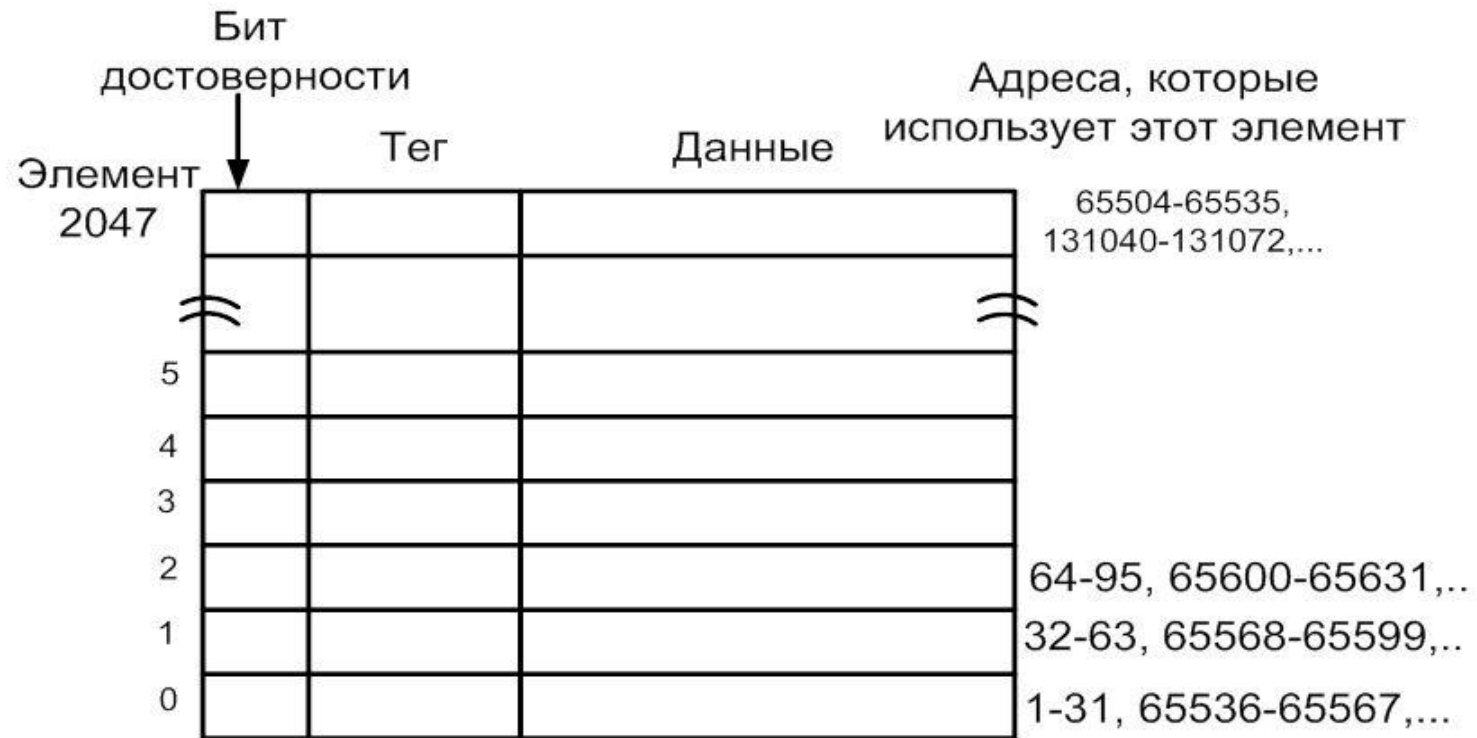
## Уровни КЭШ-памяти



Система с тремя уровнями КЭШ-памяти  MyShared



## КЭШ-память прямого отображения



Представление адреса

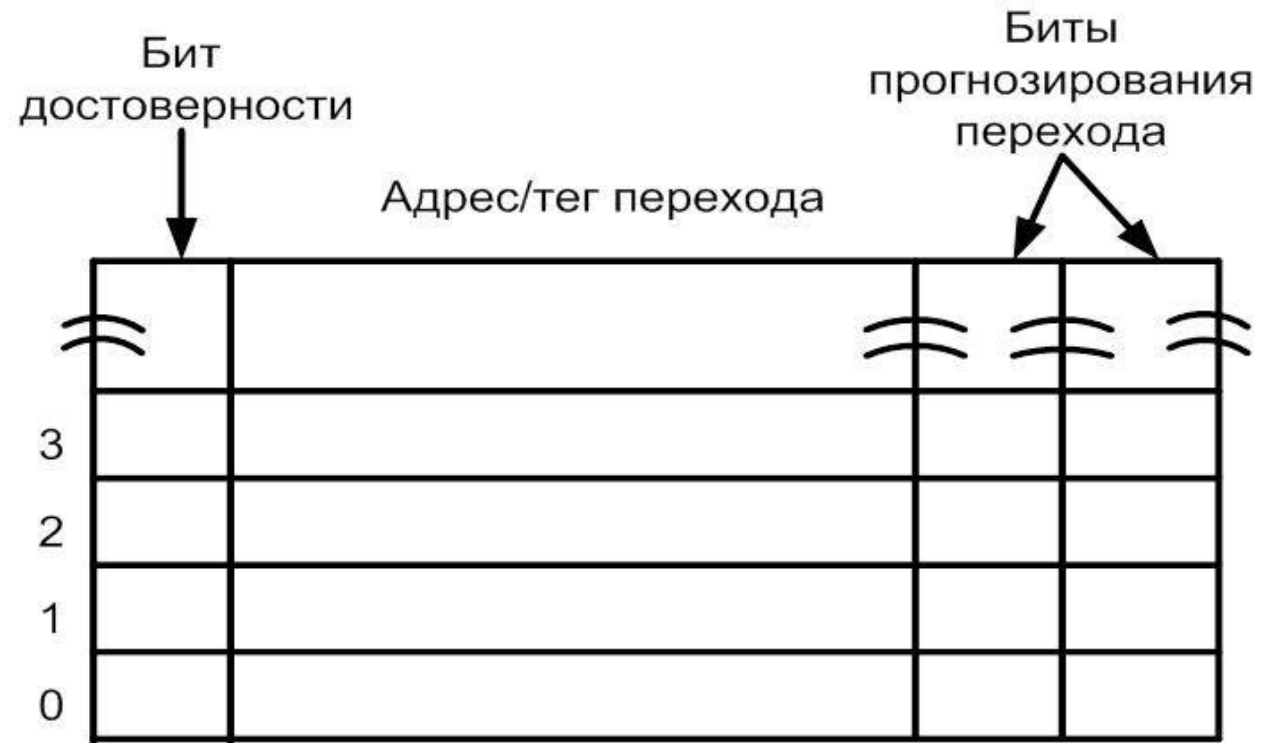
ТЕГ	№ строки	Смещение	Байт
-----	----------	----------	------

## Динамическое прогнозирование ветвлений

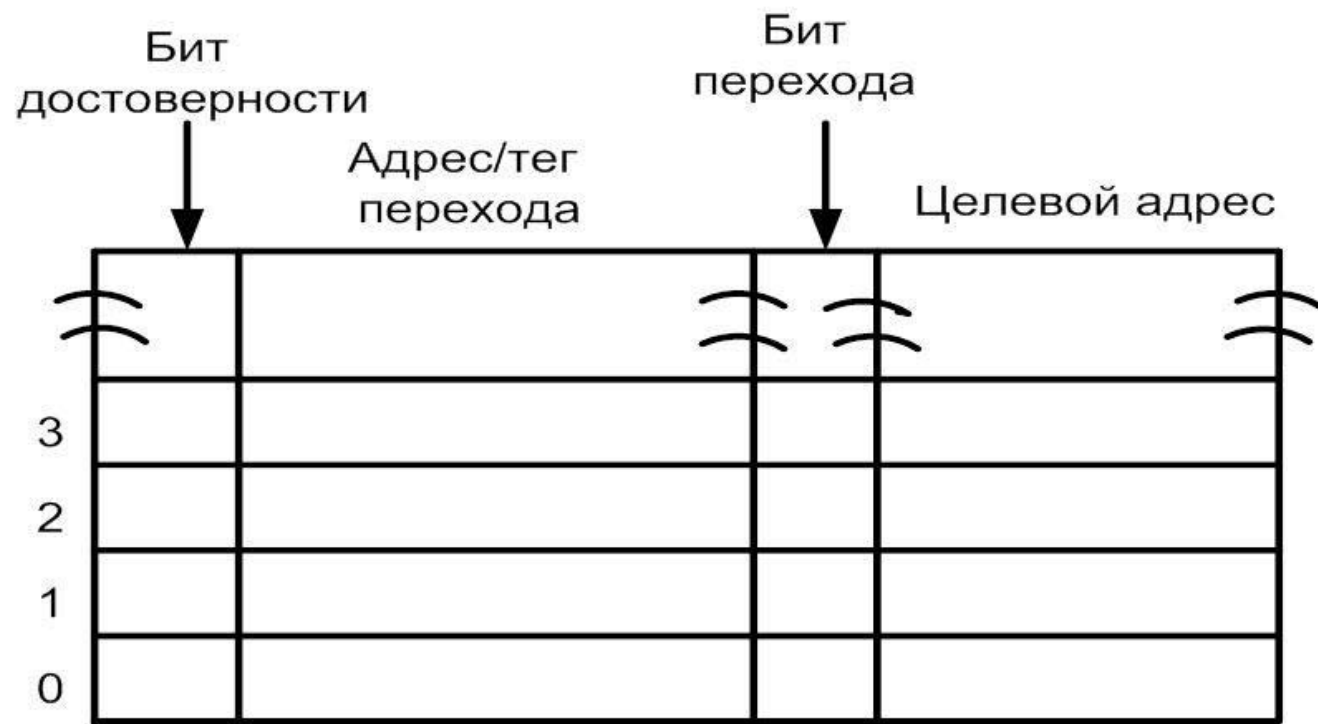


Таблица условных переходов

## Таблица с битами прогнозирования



## Таблица соответствия между адресом перехода и целевым адресом





## Лекция 8. Системы ввода/вывода

Технически система ввода/вывода в рамках микропроцессорной системы реализуется комплексом *модулей ввода/вывода* (МВВ).

Модуль ввода/вывода выполняет сопряжение ВУ с ядром микропроцессорной системы и различные коммуникационные операции между ними.

### Две основные функции МВВ:

- обеспечение интерфейса с ЦП и памятью (*«большой» интерфейс*);
- обеспечение интерфейса с одним или несколькими периферийными устройствами (*«малый» интерфейс*).



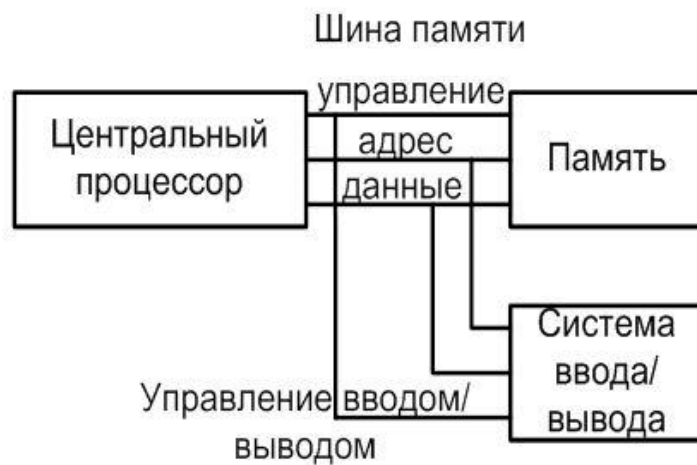
## Способы подключения СВВ к процессору



а)



б)



в)

## Распределение совмещенного адресного пространства



Совмещенное адресное пространство используется в вычислительных машинах MIPS и SPARC.

### **Достоинства совмещенного адресного пространства:**

- расширение набора команд для обращения к внешним устройствам позволяет сократить длину программы и повысить быстродействие;
- значительное увеличение количества подключаемых внешних устройств;
- возможность внепроцессорного обмена данными между внешними устройствами, если в системе команд есть команды пересылки между ячейками памяти;
- возможность обмена информацией не только с аккумулятором, но и с любым регистром центрального процессора.

### **Недостатки совмещенного адресного пространства:**

- сокращение области адресного пространства памяти;
- усложнение декодирующих схем адресов в СВВ;
- трудности распознавания операций передачи информации при вводе/выводе среди других операций. Сложности в чтении и отладке программы;
- трудности при построении СВВ на простых модулях ввода/вывода: сигналы управления не смогут координировать сложную процедуру ввода/вывода. Поэтому МВВ часто должны генерировать дополнительные сигналы под управлением программы.





### **Достоинства выделенного адресного пространства:**

- адрес внешнего устройства в команде ввода/вывода может быть коротким. В большинстве СВВ количество внешних устройств намного меньше количества ячеек памяти. Короткий адрес ВУ подразумевает такие же короткие команды ввода/вывода и простые дешифраторы;
- программы становятся более наглядными, так как операции ввода/вывода выполняются с помощью специальных команд;
- разработка СВВ может проводиться отдельно от разработки памяти.

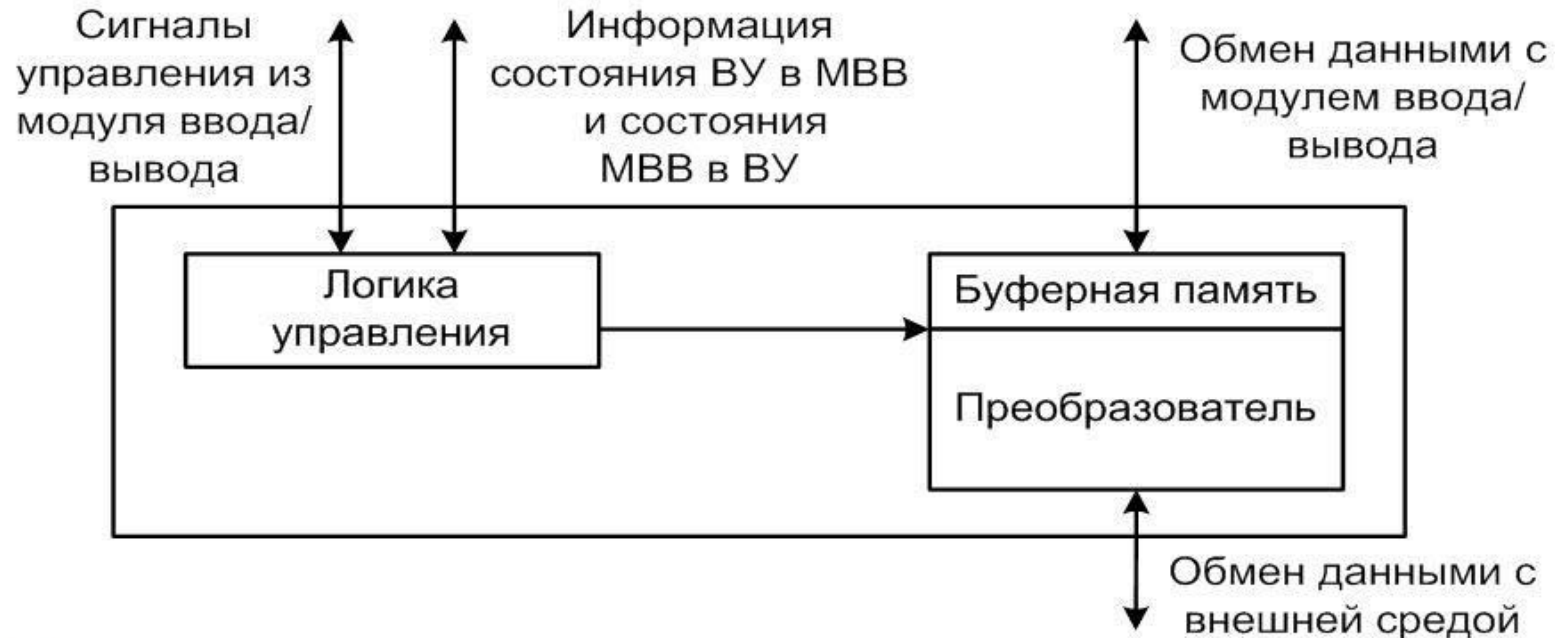
### **Недостатки выделенного адресного пространства:**

- ввод/вывод производится только через аккумулятор центрального процессора. Для передачи информации от ВУ в РОН, если аккумулятор занят, требуется выполнение четырех команд (сохранение содержимого аккумулятора, ввод из ВУ, пересылка из аккумулятора в РОН, восстановление содержимого аккумулятора);
- перед обработкой содержимого ВУ это содержимое нужно переслать в ЦП.

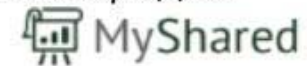


## Лекция 9. Структура внешнего устройства и модуля ввода-вывода

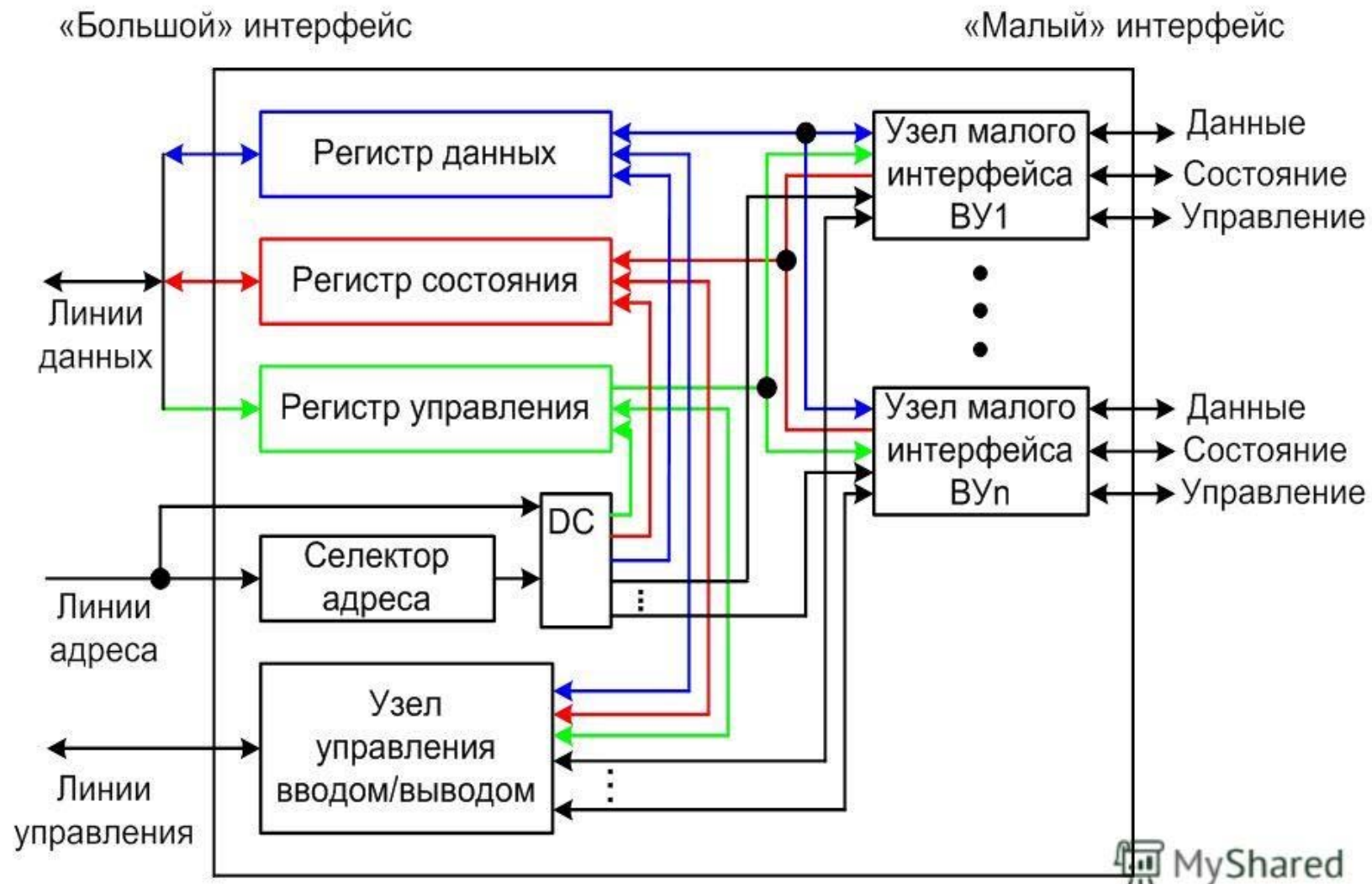
- Все множество ПУ можно свести к трем категориям:
- для общения с пользователем;
  - для общения с ВМ;
  - для связи с удаленными устройствами.



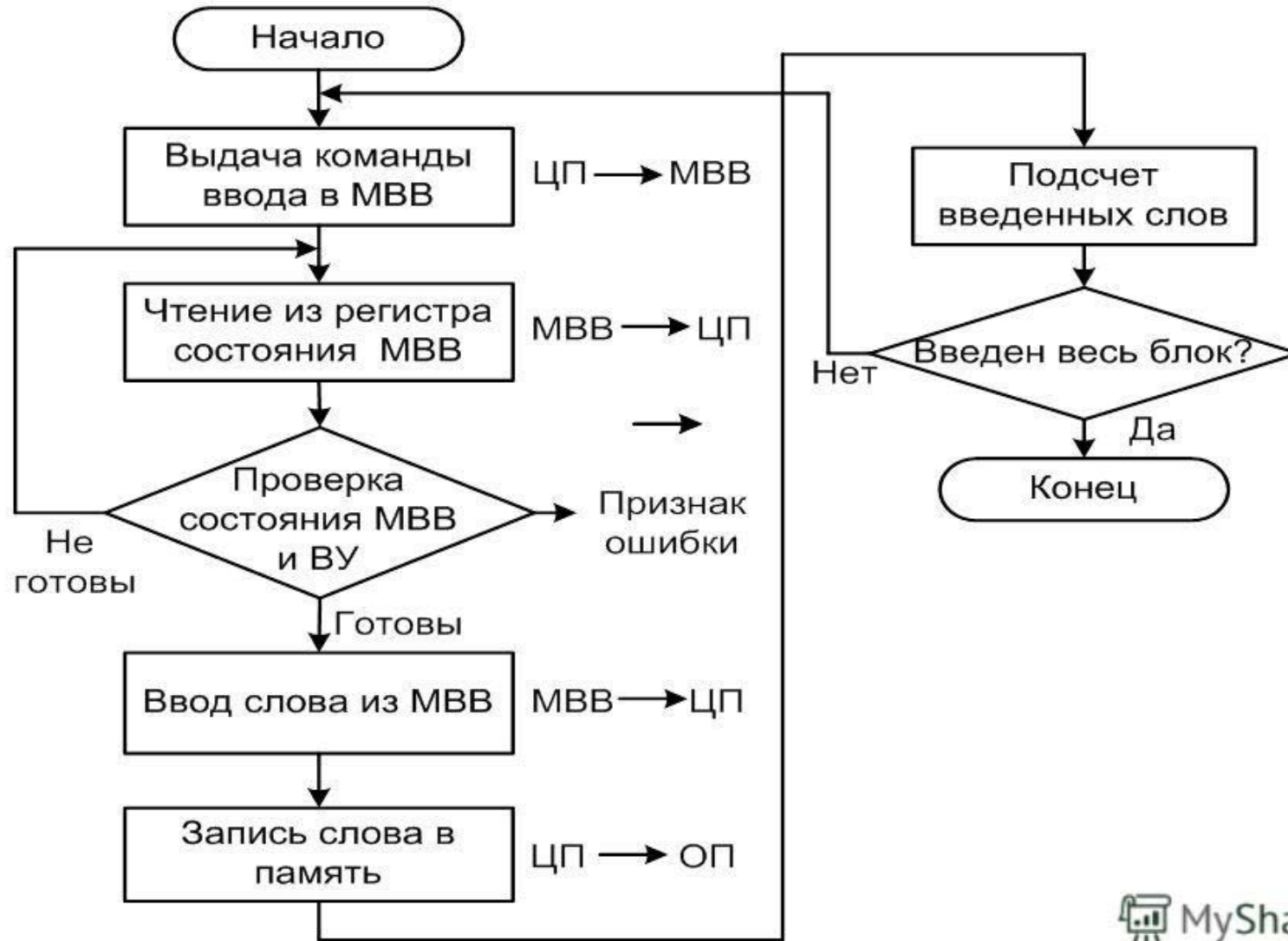
Структура внешнего устройства



# Структура модуля ввода/вывода

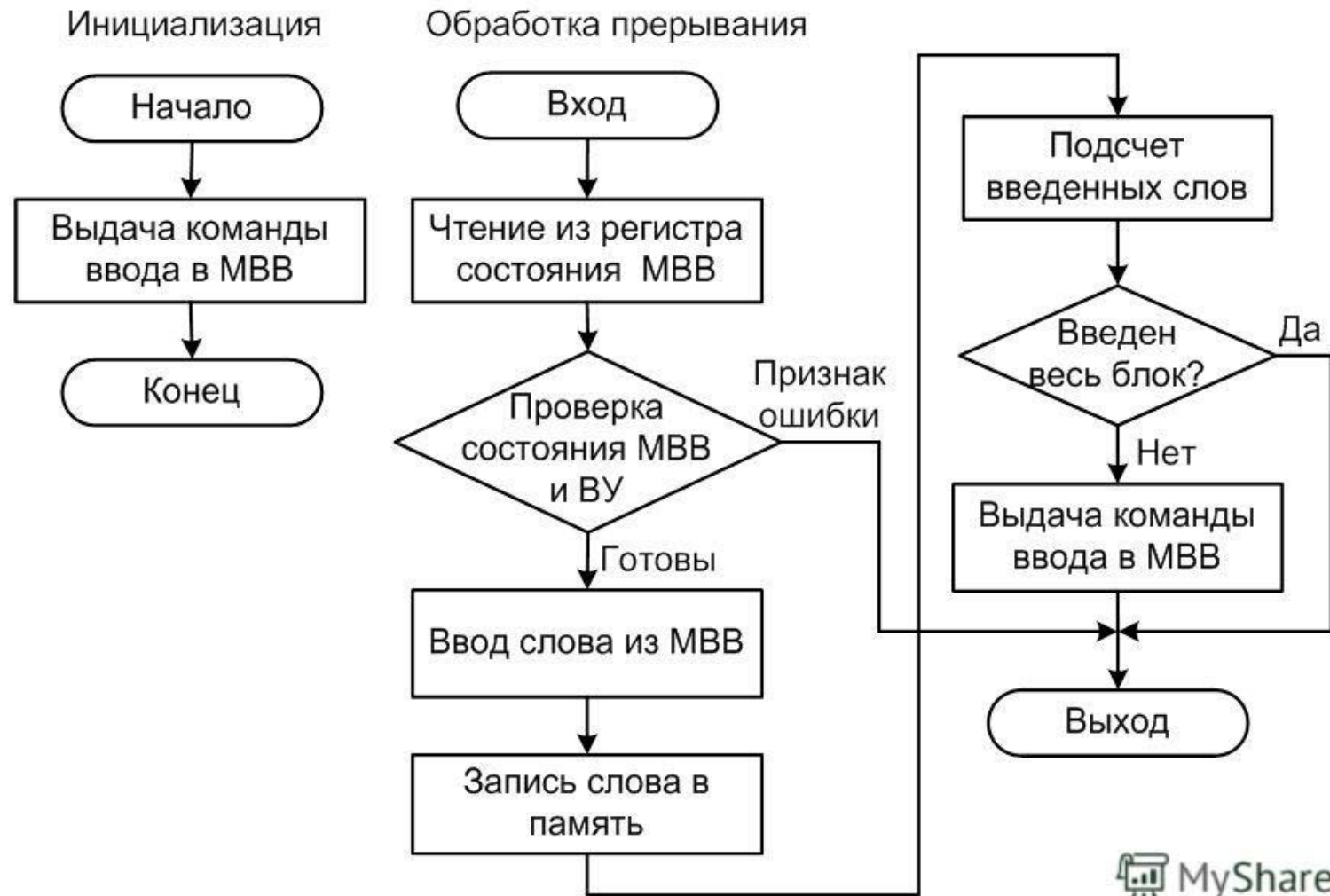


# Программно-управляемый ввод данных

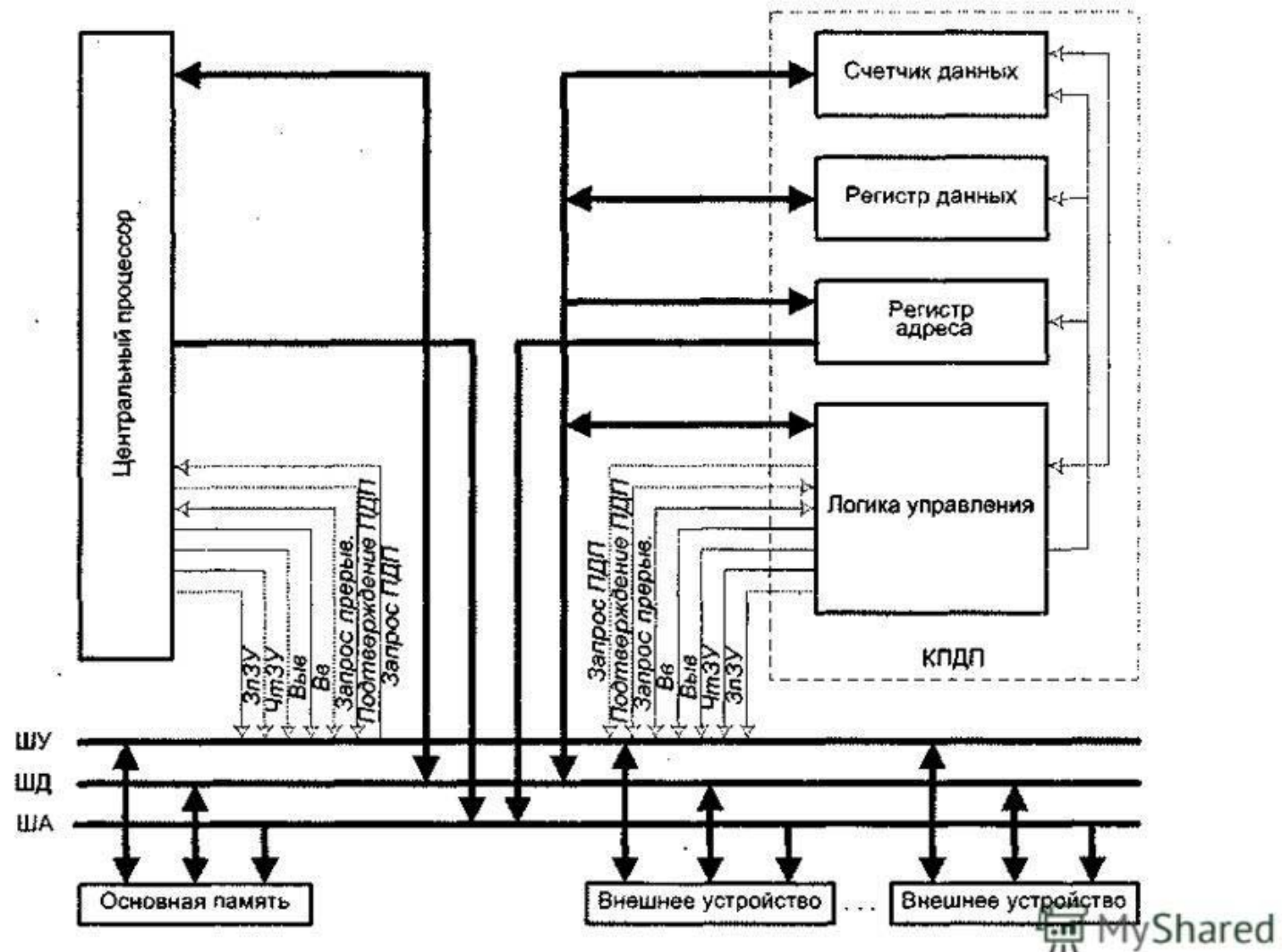




## Ввод данных по прерыванию



# Организация прямого доступа к памяти



## Организация прямого доступа к памяти

Точки возможного вмешательства в цикл команды при прямом доступе к памяти и при обычном прерывании



Возможные конфигурации систем прямого доступа к памяти



## Лекция 10. Архитектура микроконтроллеров

Микроконтроллер – это самостоятельная компьютерная система, которая содержит процессор, вспомогательные схемы и устройства ввода/ вывода данных, размещенные в общем корпусе.

### Основные типы:

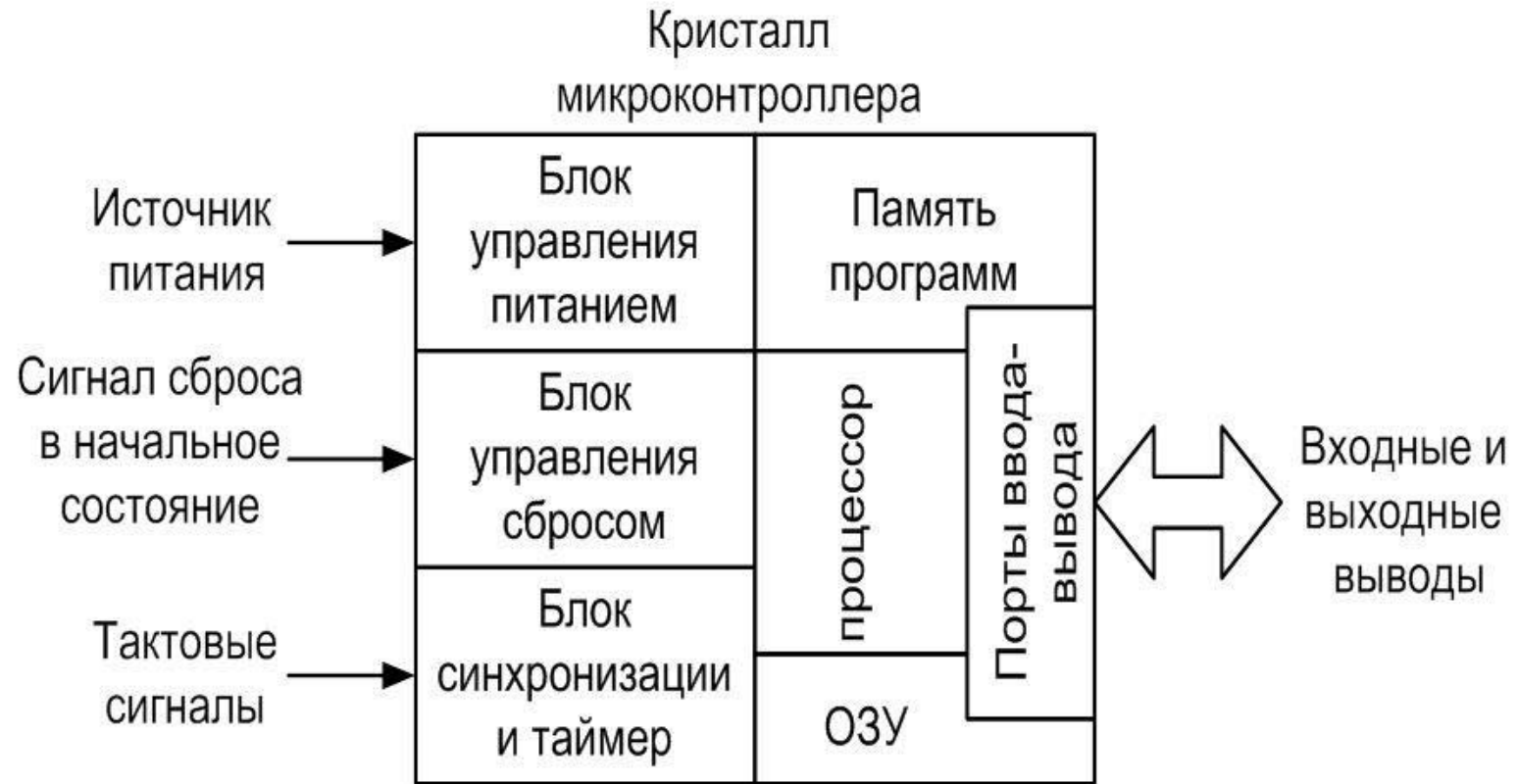
1. Встраиваемые 8-разрядные МК;
2. 16- и 32-разрядные МК с внешней памятью;
3. Цифровые сигнальные процессоры (DSP).

Типичные значения  $\max$  частоты тактовых сигналов составляют для различных микроконтроллеров 10-30 МГц. Главным фактором, ограничивающим их скорость, является время доступа к памяти, применяемой в МК.

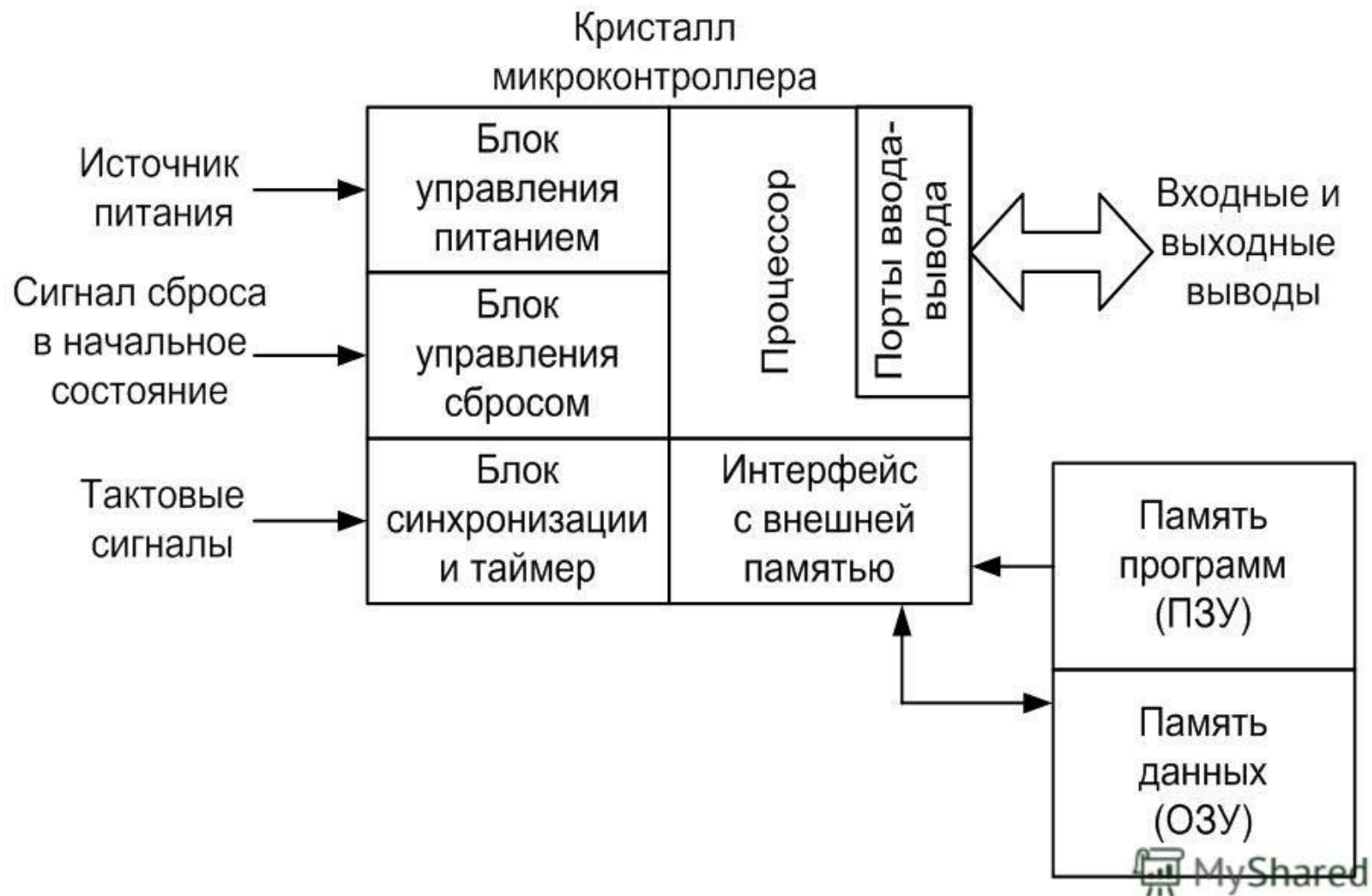




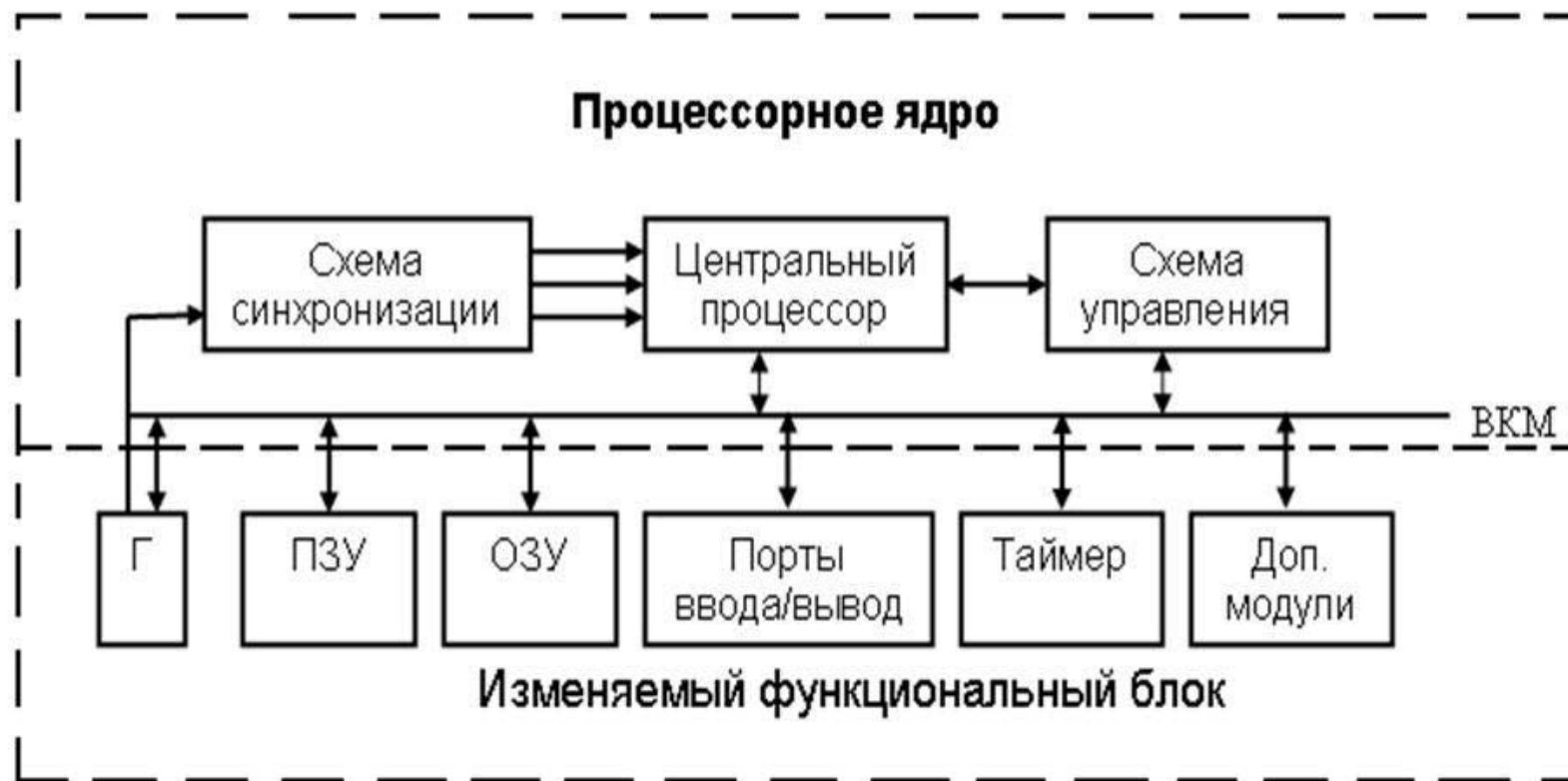
## Структура встраиваемых микроконтроллеров



## Микроконтроллеры с внешней памятью

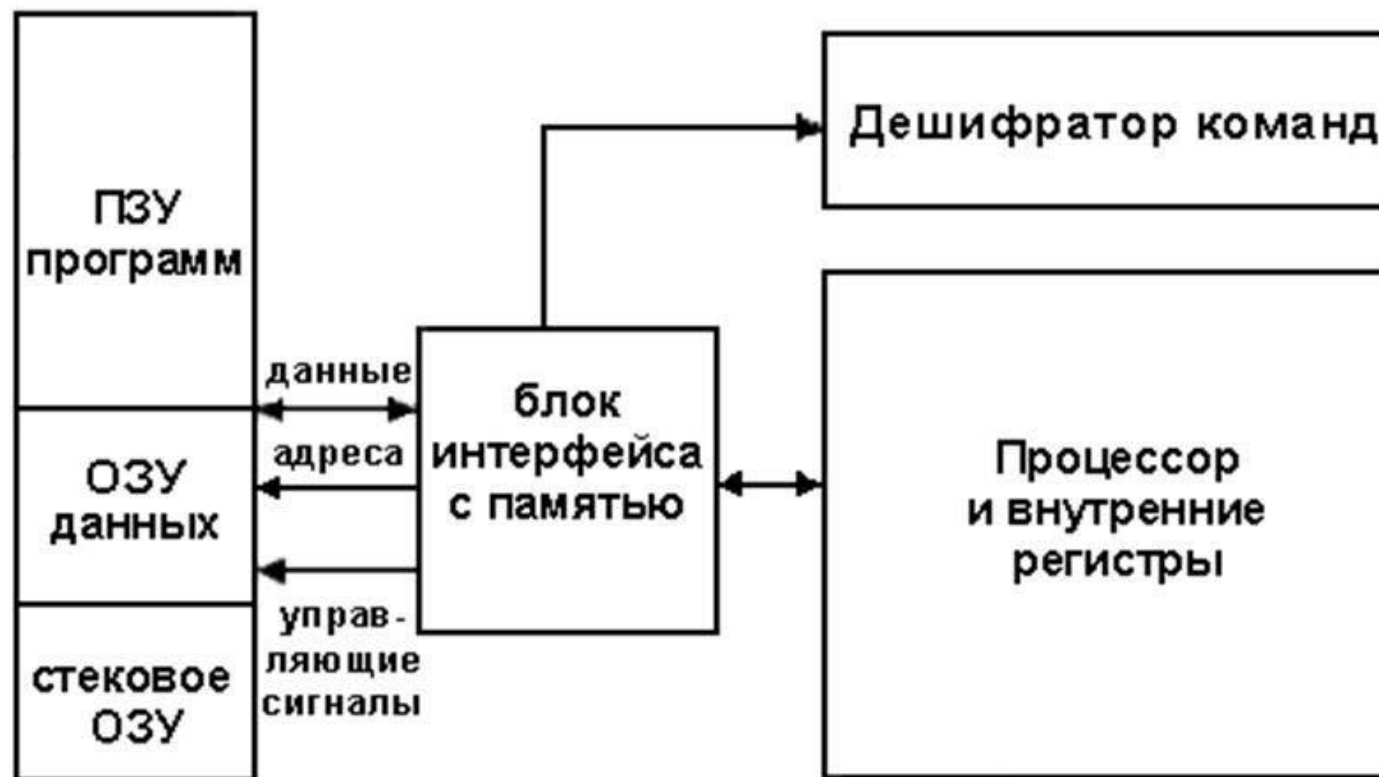


## Модульная организация микроконтроллеров



# Принстонская архитектура микроконтроллера

Блок памяти





## Гарвардская архитектура микроконтроллера



# Лекция 11. Типы памяти микроконтроллеров

Подсистема памяти микроконтроллеров содержит:

1. Память программ (ПЗУ);
2. Память данных (ОЗУ);
3. Регистры МК.

## Память программ

1. ROM - (масочно-программируемое ПЗУ);
2. PROM (однократно-программируемое ПЗУ);
3. EPROM (электрически программируемое ПЗУ с ультрафиолетовым стиранием);
4. EEPROM (ПЗУ с электрической записью и стиранием);
5. Flash-разновидность EEPROM.



## Память данных

Память данных МК выполняется на основе статического ОЗУ. Содержимое ячеек ОЗУ сохраняется при снижении тактовой частоты МК до сколь угодно малых значений (с целью снижения энергопотребления).

**U**STANDBY-«напряжение хранения информации». При снижении напряжения питания ниже минимально допустимого уровня **UDDMIN**, но выше уровня **U**STANDBY работа программы МК выполняться не будет, но информация в ОЗУ сохраняется.

Уровень напряжения хранения составляет обычно около 1В, что позволяет в случае необходимости перевести МК на питание от автономного источника (батареи) и сохранить в этом режиме данные ОЗУ.

## Регистры микроконтроллеров

1. Регистры процессора: аккумулятор, регистры состояния, индексные регистры,
2. Регистры управления: регистры управления прерываниями, таймером,
3. Регистры, обеспечивающие ввод/вывод данных: регистры данных портов, регистры управления параллельным, последовательным или аналоговым вводом/ выводом.

Одним из важных вопросов является размещение регистров в адресном пространстве МК. В некоторых МК все регистры и память данных располагаются в одном адресном пространстве. Это означает, что память данных совмещена с регистрами. Такой подход называется «отображением ресурсов МК на память».

В других МК адресное пространство устройств ввода/вывода отделено от общего пространства памяти. Отдельное пространство ввода/вывода дает некоторое преимущество процессорам с гарвардской архитектурой, обеспечивая возможность считывать команду во время обращения к регистру ввода/вывода.





## Стек микроконтроллеров

В микроконтроллерах ОЗУ данных используется также для организации вызова подпрограмм и обработки прерываний. При этих операциях содержимое программного счетчика и основных регистров (аккумулятор, регистр состояния и другие) сохраняется и затем восстанавливается при возврате к основной программе.

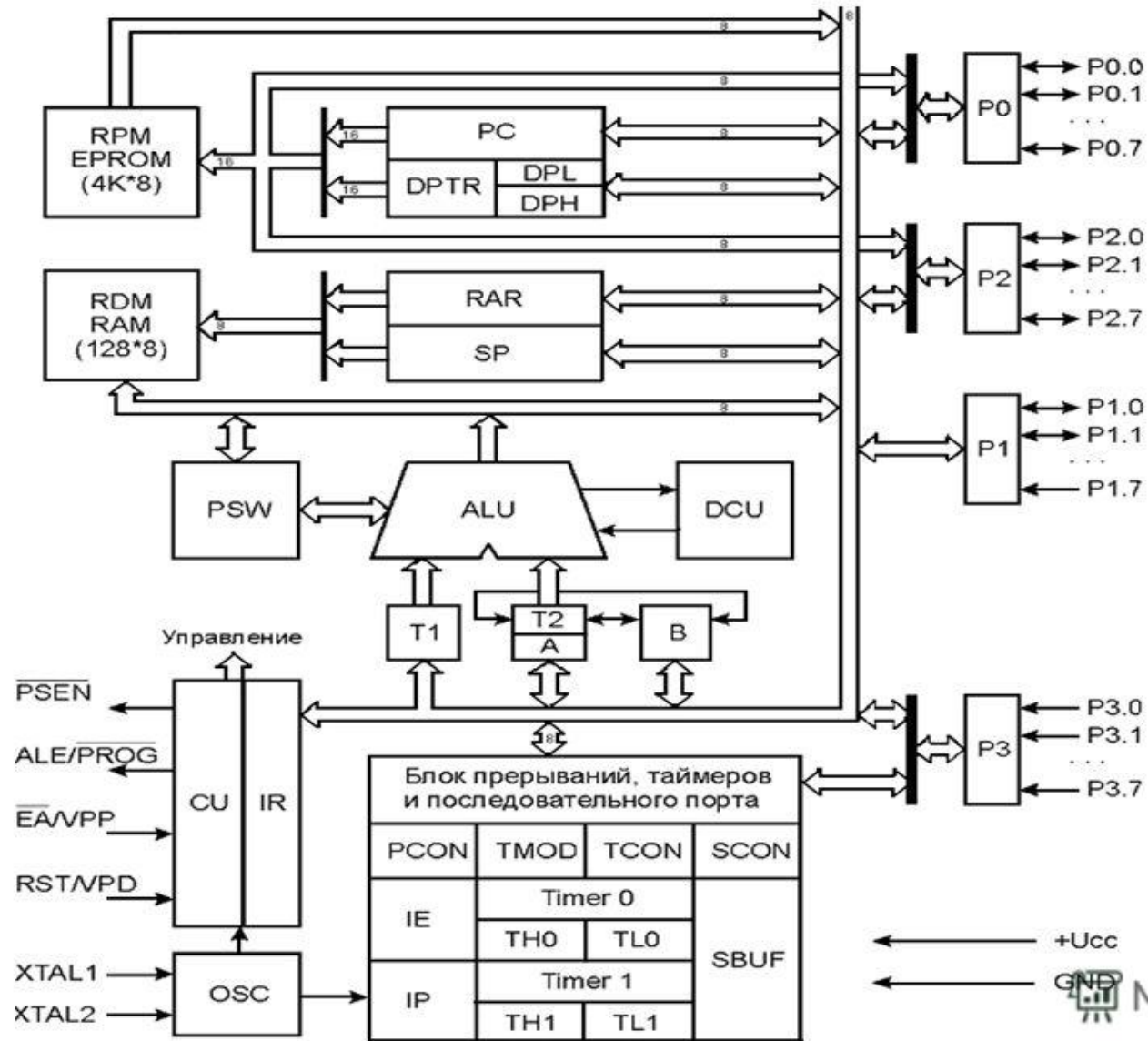
## Внешняя память

Если МК содержит специальные аппаратные средства для подключения внешней памяти, то эта операция производится штатным способом (как для МП).

Второй, более универсальный, способ заключается в том, чтобы использовать порты ввода/вывода для подключения внешней памяти и реализовать обращение к памяти программными средствами.



# Лекция 12. Базовая архитектура микроконтроллера 8051



MyShared

## Характеристики микроконтроллеров семейства 8051

- тактовая частота 24 МГц;
- командный цикл 12 тактов;
- объем памяти программ 4 Кб;
- объем памяти данных 128 байт;
- число линий ввода/вывода – 32;
- два 8/16-разрядных таймера;
- множество внутренних и внешних источников прерываний;
- программируемый последовательный порт;
- интерфейс с внешней памятью объемом до 128 Кб.

МК 8051 имеет Гарвардскую архитектуру, то есть память программ и память данных являются самостоятельными и независимыми друг от друга устройствами. К адресному пространству памяти данных примыкает адресное пространство регистров специальных функций SFR (Special Function Register).

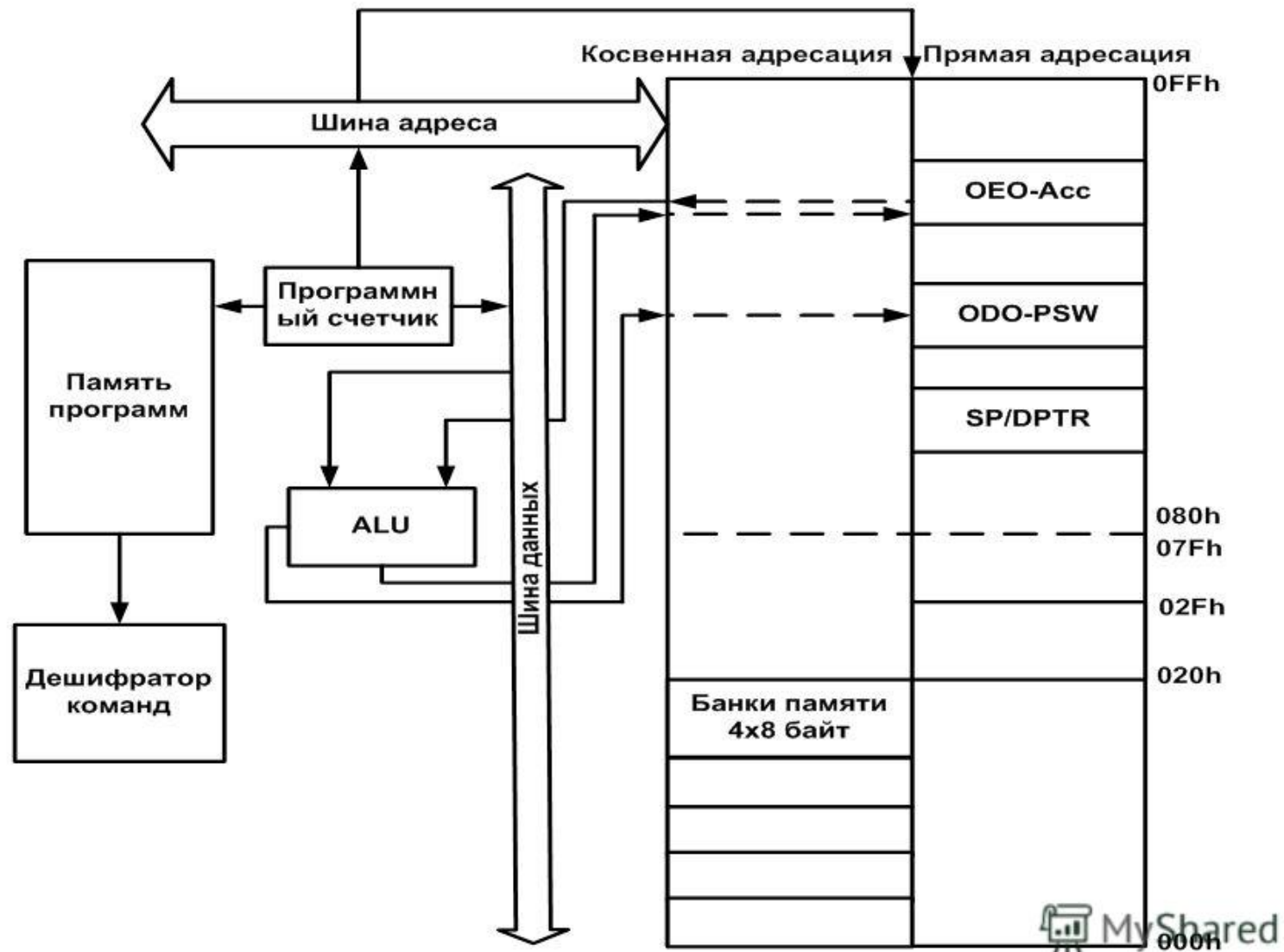


## Адресное пространство регистров специальных функций

Адрес	Символ	Наименование
0E0H	*ACC	Аккумулятор (Accumulator)
0F0H	*B	Регистр расширитель аккумулятора (Multiplication Register)
0D0H	*PSW	Слово состояния программы (Program Status Word)
080H	*P0	Порт 0 (SFR P0)
090H	*P1	Порт 1 (SFR P1)
0A0H	*P2	Порт 2 (SFR P2)
0B0H	*P3	Порт 3 (SFR P3)
081H	SP	Регистр указатель стека (Stack Pointer)
083H	DPH	Старший байт регистра указателя данных DPTR (Data Pointer High)
082H	DPL	Младший байт регистра указателя данных DPTR (Data Pointer Low)
08CH	TH0	Старший байт таймера 0 ()
08AH	TL0	Младший байт таймера 0 ()



# Базовая архитектура микроконтроллера 8051





## Регистр флагов (PSW)

Символ	Позиция	Имя и назначение																				
P	PSW.0	Флаг приоритета. Устанавливается и сбрасывается аппаратно в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе																				
-	PSW.1	Не используется																				
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций																				
RS0 - RS1	PSW.3 - PSW.4	Биты выбора используемого банка регистров. Могут быть изменены программным путем																				
		<table border="1" style="margin: auto;"> <thead> <tr> <th>RS0</th> <th>RS1</th> <th>Банк</th> <th>Границы адресов ОЗУ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>00H - 07H</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>08H - 0FH</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>10H - 17H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18H - 1FH</td> </tr> </tbody> </table>	RS0	RS1	Банк	Границы адресов ОЗУ	0	0	0	00H - 07H	1	0	1	08H - 0FH	0	1	2	10H - 17H	1	1	3	18H - 1FH
		RS0	RS1	Банк	Границы адресов ОЗУ																	
		0	0	0	00H - 07H																	
		1	0	1	08H - 0FH																	
0	1	2	10H - 17H																			
1	1	3	18H - 1FH																			
FO	PSW.5	Флаг пользователя. Может быть установлен, сброшен или проверен программой пользователя																				
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заеме в бите 3 аккумулятора																				
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается как аппаратно, так и программным путем																				

## Лекция 13. Питание микроконтроллеров

Во многих приложениях используется батарейное питание микроконтроллеров, а в некоторых случаях применяются конденсаторы большой емкости, которые обеспечивают сохранение работоспособности при кратковременных отключениях питания. Поэтому проблема снижения энергопотребления весьма актуальна для микроконтроллеров.

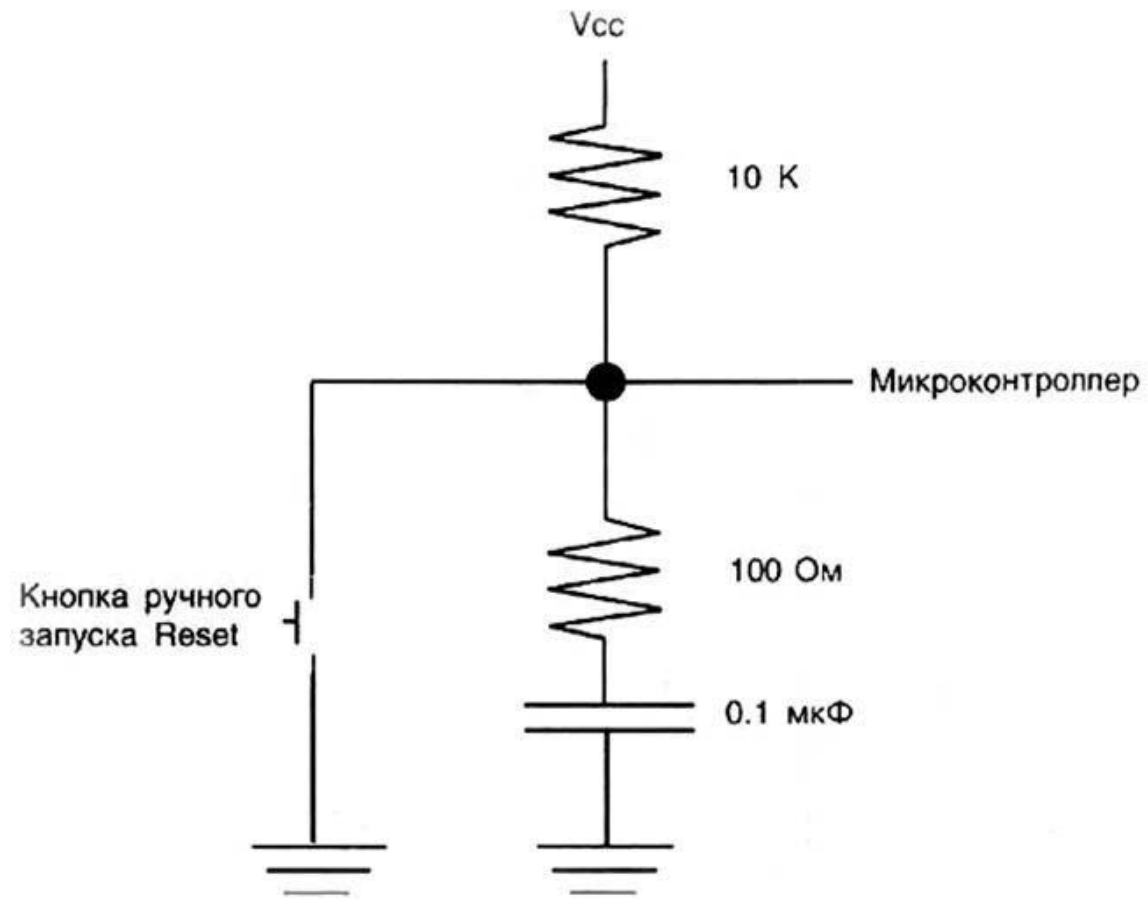
### Потребляемая мощность

Имеются три значения мощности, потребляемой микроконтроллером в различных рабочих условиях:

1. Собственная мощность, необходимая для нормальной работы микроконтроллера;
2. Мощность, потребляемая устройствами ввода-вывода, которую требуется учитывать, когда микроконтроллер производит обмен данными с внешними устройствами;
3. Мощность, потребляемая в «спящем» режиме.



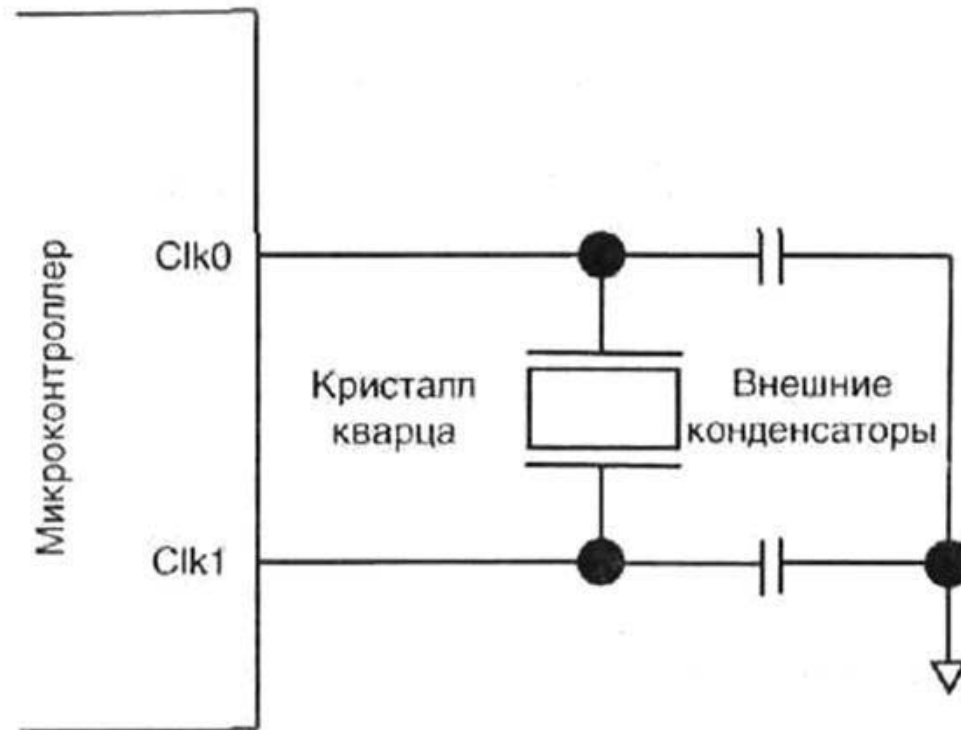
## Схема запуска микроконтроллера (сброс в начальное состояние)



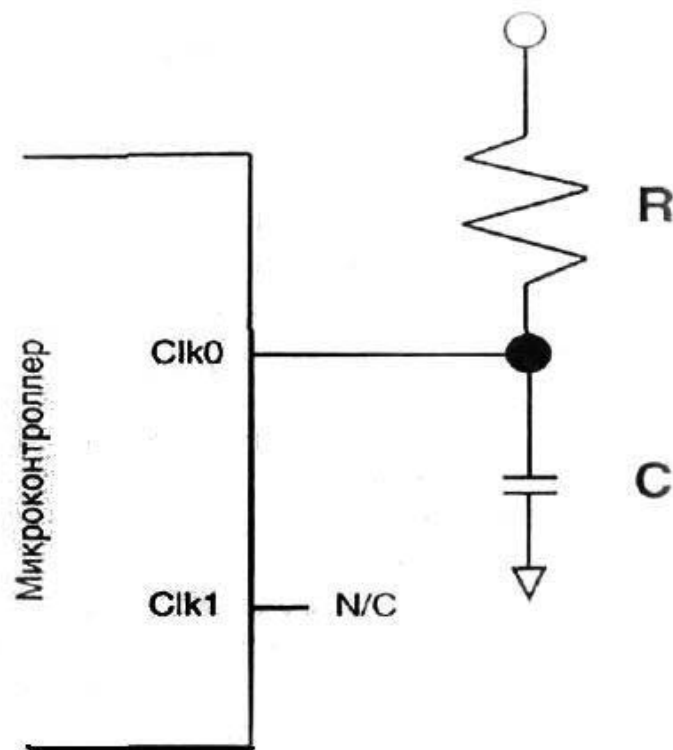
## Тактирование системы

Существует три способа задания тактовой частоты микроконтроллера

### 1. Использование кварцевого резонатора



## 2. Использование RC-генератора



## 3. Использование внешней микросхемы тактового генератора



## Лекция 14. Система команд микроконтроллера семейства 8051

Микро-ЭВМ рассматриваемого семейства являются типичными микропроцессорными устройствами с архитектурой SISIC - со стандартным набором команд. Поэтому их система команд довольно обширна и включает в себя 111 основных команд. Их длина – один, два или три байта, причем большинство из них (94%) – одно- или двухбайтные.

Все команды выполняются за один или два машинных цикла (соответственно 1 или 2 мкс при тактовой частоте 12 МГц), исключение – команды умножения и деления, которые выполняются за четыре машинных цикла (4 мкс). Микро-ЭВМ семейства 8051 используют прямую, непосредственную, косвенную и неявную, адресацию данных.



## Типы команд

Тип команды	Первый байт D7...D0	Второй байт D7...D0	Третий байт D7...D0
тип 1	коп		
тип 2	коп	#d	
тип 3	коп	ad	
тип 4	коп	bit	
тип 5	коп	rel	
тип 6	коп	a7...a0	
тип 7	коп	ad	#d
тип 8	коп	ad	rel
тип 9	коп	ads	add
тип 10	коп	#d	rel
тип 11	коп	bit	rel
тип 12	коп	ad16h	ad16h
тип 13	коп	#d16h	#d16h

MyShared

## Группы команд

1. пересылки данных;
2. арифметических операций;
3. логических операций;
4. операций над битами;
5. передачи управления.

## Способы адресации

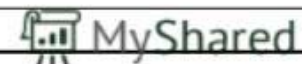
1. Непосредственная адресация     **ADD A, # 77.**

Значение непосредственного операнда обозначается символом # .

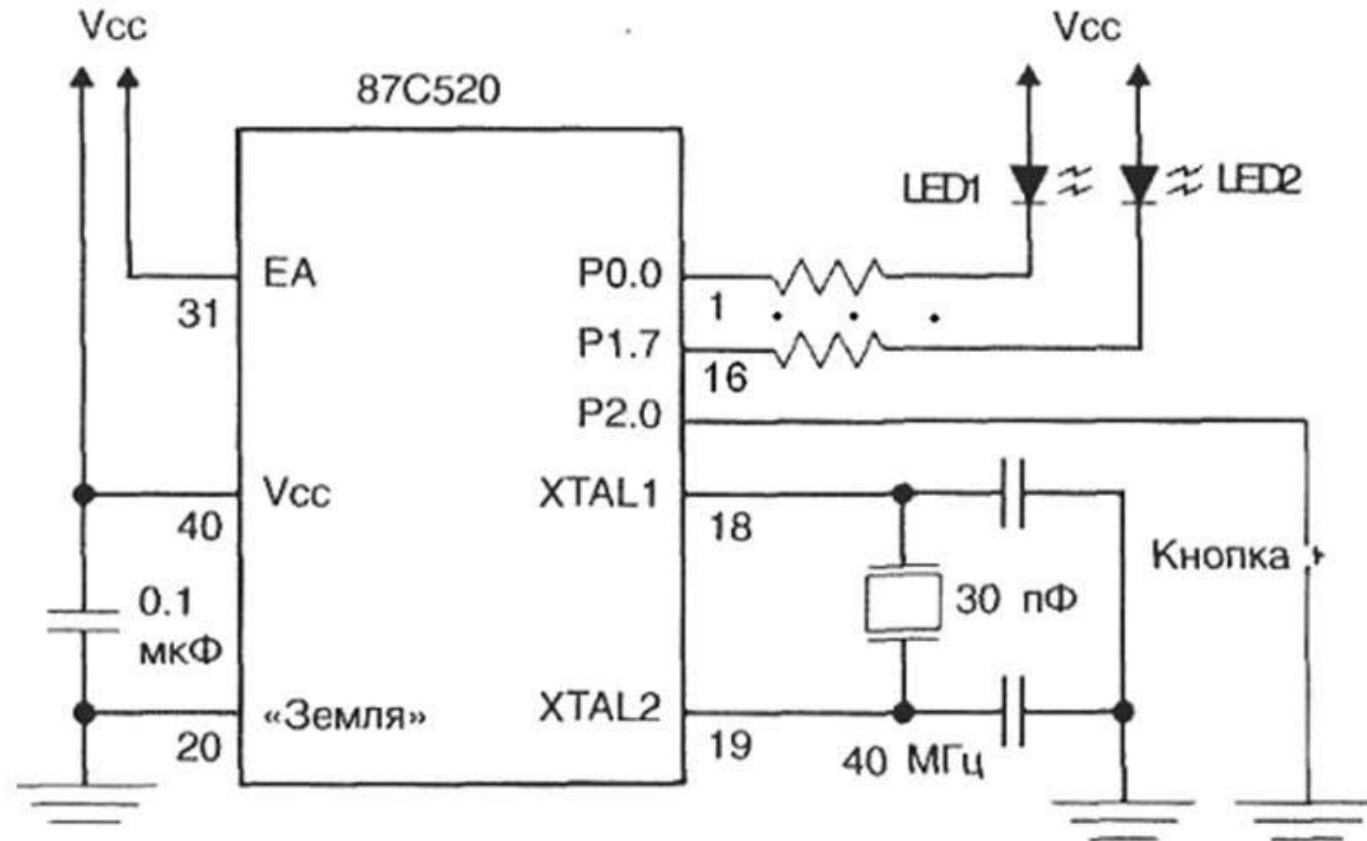
2. Регистровая адресация     **MOV A, Rn.**
3. Прямая адресация     **MOV A, 0D0h.**
4. Косвенно-регистровая адресация     **MOV A, @Ri.**
5. Индексная адресация     **MOVC A, @A+DPTR.**

## Команды пересылки данных микроконтроллера 8051

Название команды	Мнемокод	Пример
Пересылка в аккумулятор из регистра (n=0÷7)	MOV A, Rn	MOV A, R1 MOV A, P2 MOV A, PSW
Пересылка в аккумулятор прямоадресуемого байта	MOV A, ad	MOV A, 21A3
Пересылка в аккумулятор байта из РПД (i=0,1)	MOV A, @Ri	MOV A, @R2
Загрузка в аккумулятор константы	MOV A, #d	MOV A, #45 MOV A, #013h
Пересылка в регистр из аккумулятора	MOV Rn, A	MOV R3, A MOV P0, A
Пересылка в регистр прямоадресуемого байта	MOV Rn, ad	MOV R2, 0121
Загрузка в регистр константы	MOV Rn, #d	MOV R7, #0ABh MOV TMOD, #00000001b MOV P3, #10010001b
Пересылка по прямому адресу аккумулятора	MOV ad, A	MOV 0021, A
Пересылка по прямому адресу регистра	MOV ad, Rn	MOV 0100, R5
Загрузка в стек	PUSH ad	
Извлечение из стека	POP ad	
Обмен аккумулятора с регистром	XCH A, Rn	
Обмен аккумулятора с прямоадресуемым байтом	XCH A, ad	
Обмен аккумулятора с байтом из РПД	XCH A, @Ri	
Обмен младших тетрад аккумулятора и байта РПД	XCHD A, @Ri	



## Функциональная схема подключения ресурсов

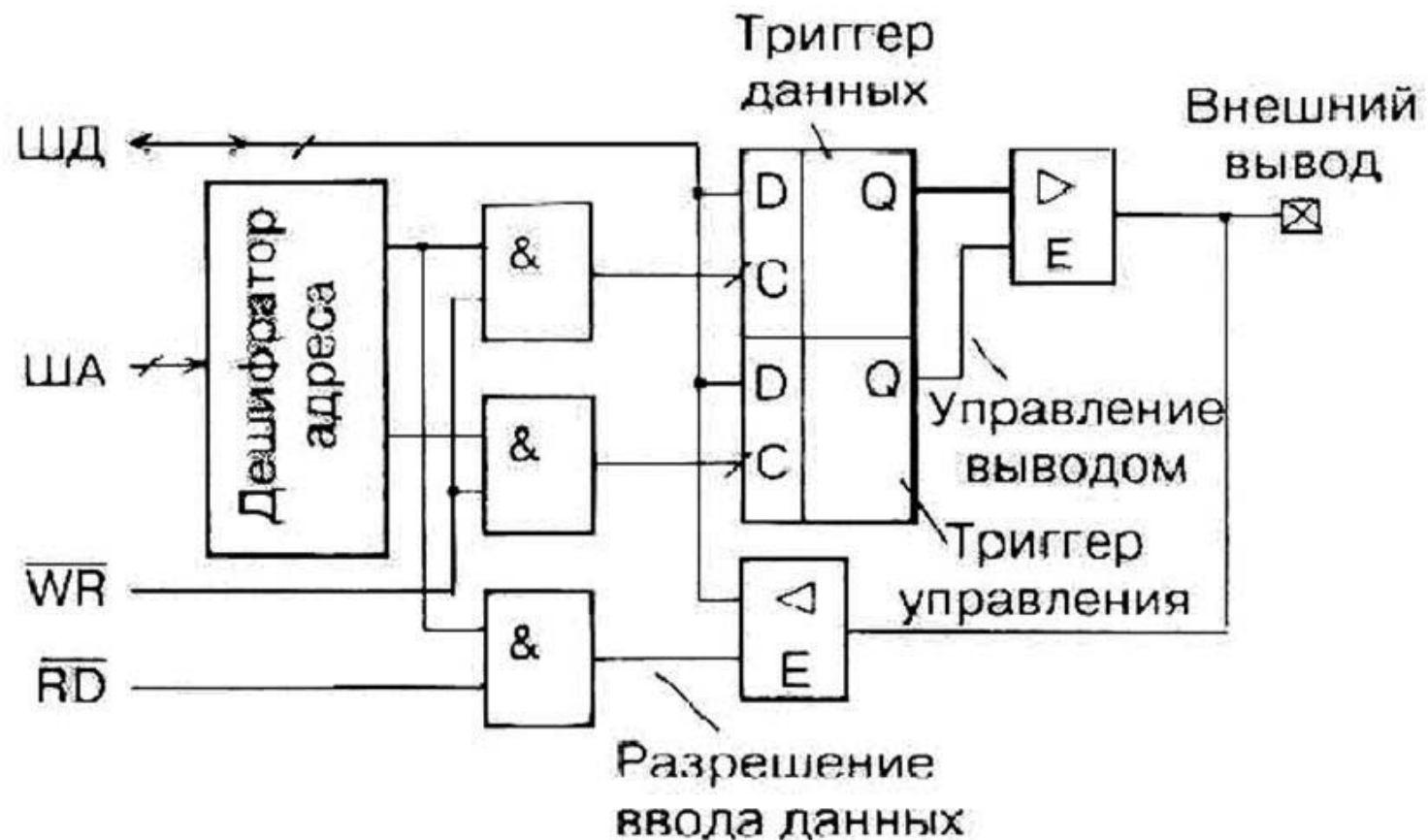




## Таблица распределения подключаемых ресурсов к разрядам портов

Ресурс	Разряд порта
Семисегментный индикатор 1	
A1	P0.1
B1	P0.2
C1	P0.3
D1	P0.4
E1	P0.5
F1	P0.6
G1	P0.7
Семисегментный индикатор 2	
A2	P1.1
B2	P1.2
C2	P1.3
D2	P1.4
E2	P1.5
F2	P1.6
G2	P1.7

## Лекция 15. Связь микроконтроллера с внешней средой и временем



Типовая схема двунаправленного порта ввода/вывода МК

## Организация портов ввода вывода

<b>Вывод порта</b>	<b>Альтернативная функция</b>
<b>P3.0</b>	<b>RXD - вход последовательного порта</b>
<b>P3.1</b>	<b>TXD - выход последовательного порта</b>
<b>P3.2</b>	<b>INT0 - внешнее прерывание 0</b>
<b>P3.3</b>	<b>INT1 - внешнее прерывание 1</b>
<b>P3.4</b>	<b>T0 - вход таймера-счетчика 0</b>
<b>P3.5</b>	<b>T1 - вход таймера-счетчика 1</b>
<b>P3.6</b>	<b>WR - строб записи во внешнюю память данных</b>
<b>P3.7</b>	<b>RD - строб чтения из внешней памяти данных</b>

## Структура модуля таймера-счетчика

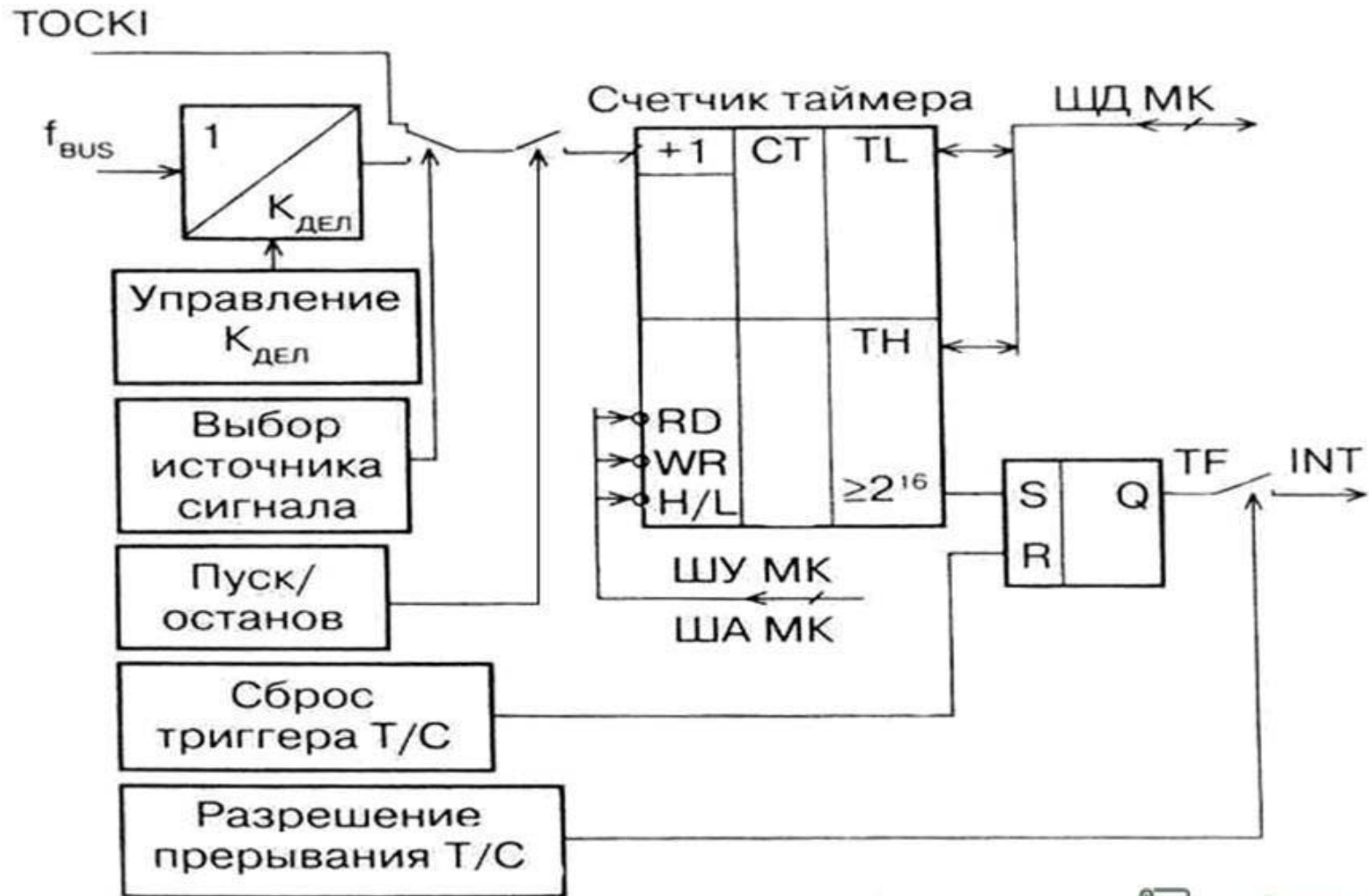


Таблица 23.2 - Регистр режима работы таймера/счетчика TMOD

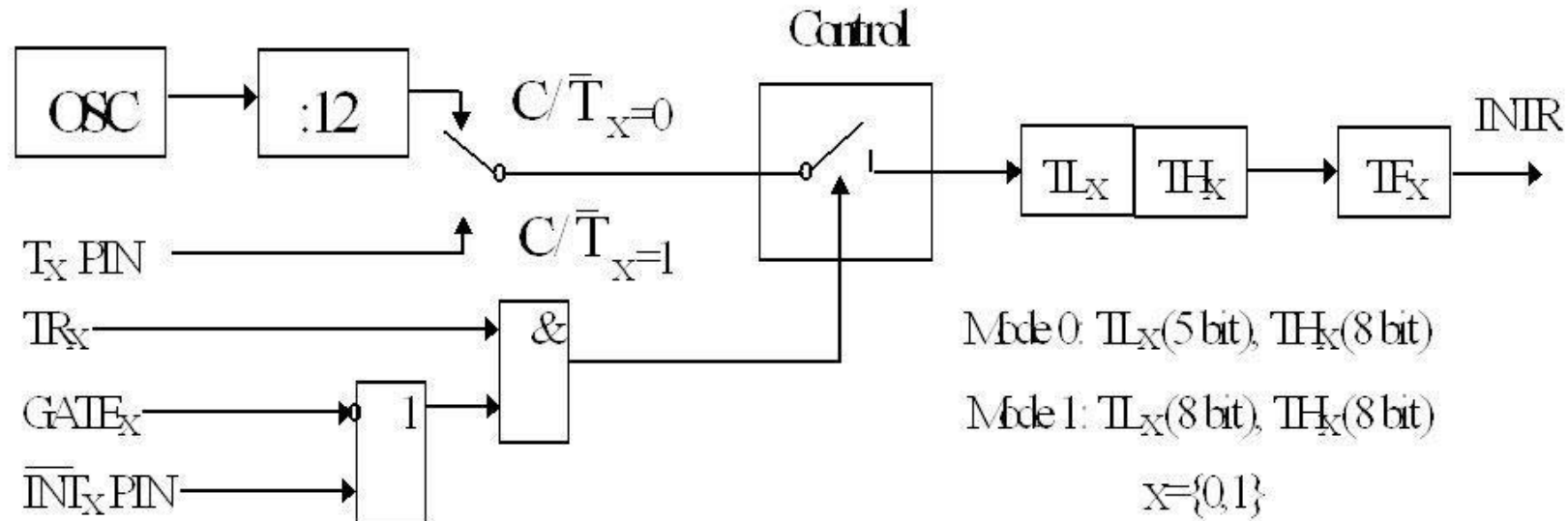
Символ	Позиция	Имя и назначение		
GATE	TMOD.7 для T/C1 и TMOD.3 для T/CO	<b>Управление блокировкой.</b> Если бит установлен, то таймер/счетчик "x" разрешен до тех пор, пока на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то T/C разрешается, как только бит управления "TRx" устанавливается		
	TMOD.6 для T/C1 и TMOD.2 для T/CO	<b>Бит выбора режима таймера или счетчика событий.</b> Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если; установлен, то работает счетчик от внешних сигналов на входе "Tx"		
M1	TMOD.5 для T/C1 и TMOD.1 для T/CO	<b>Режим работы</b>		
		<b>M1M0</b>		
		0	0	Таймер BE48. "TLx" работает как 5-битный предделитель
		0	1	16 битный таймер/счетчик. "THx" и "TLx" включен последовательно
M0	TMOD.4 для T/C1 и TMOD.0 для T/CO	1	0	8-битный авто перезагружаемый таймер/счетчик. "THx" хранит значение, которое должно быть перезагружено в "TLx" каждый раз по переполнению
		1	1	Таймер/счетчик 1 останавливается. Таймер/счетчик 0: TLO работает как 8-битный таймер/счетчик, и его режим определяется управляющими битами таймера 0. TH0 работает только как 8 битный таймер, и его режим определяется управляющими битами таймера 1



Таблица 23.3 - Регистр управления/статуса таймера TCON.

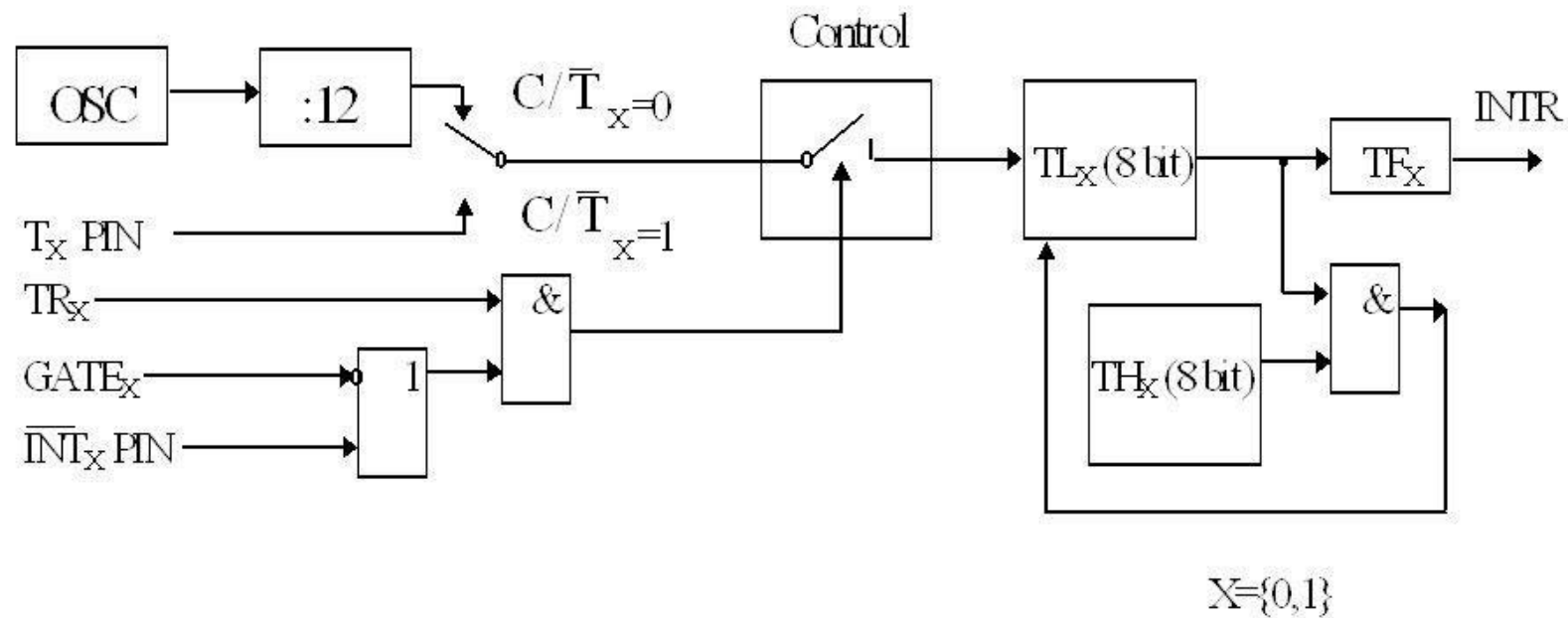
Символ	Позиция	Имя и назначение
TF1	TCON.7	<b>Флаг переполнения таймера 1.</b> Устанавливается аппаратно при переполнении таймера/счетчика. Сбрасывается при обслуживании прерывания аппаратно
TR1	TCON.6	<b>Бит управления таймером 1.</b> Устанавливается, / сбрасывается программой для пуска/останова
TF0	TCON.5	<b>Флаг переполнения таймера 0.</b> Устанавливается аппаратно. Сбрасывается при обслуживании прерывания
TR0	TCON.4	<b>Бит управления таймером 0.</b> Устанавливается / сбрасывается программой для пуска/останова таймера/счетчика
IE1	TCON.3	<b>Флаг фронта прерывания 1.</b> Устанавливается аппаратно, когда детектируется срез внешнего сигнала INT1. Сбрасывается при обслуживании прерывания
IT1	TCON.2	<b>Бит управления типом прерывания 1.</b> Устанавливается / сбрасывается программно для спецификации запроса INT1 (срез/низкий уровень)
IE0	TCON.1	<b>Флаг фронта прерывания 0.</b> Устанавливается по срезу сигнала INT0. Сбрасывается при обслуживании прерывания
IT0	TCON.0	<b>Бит управления типом прерывания 0.</b> Устанавливается / сбрасывается программно для спецификации запроса INT0 (срез/низкий уровень)

## Логика работы T/C0 и T/C1 в режимах 0, 1



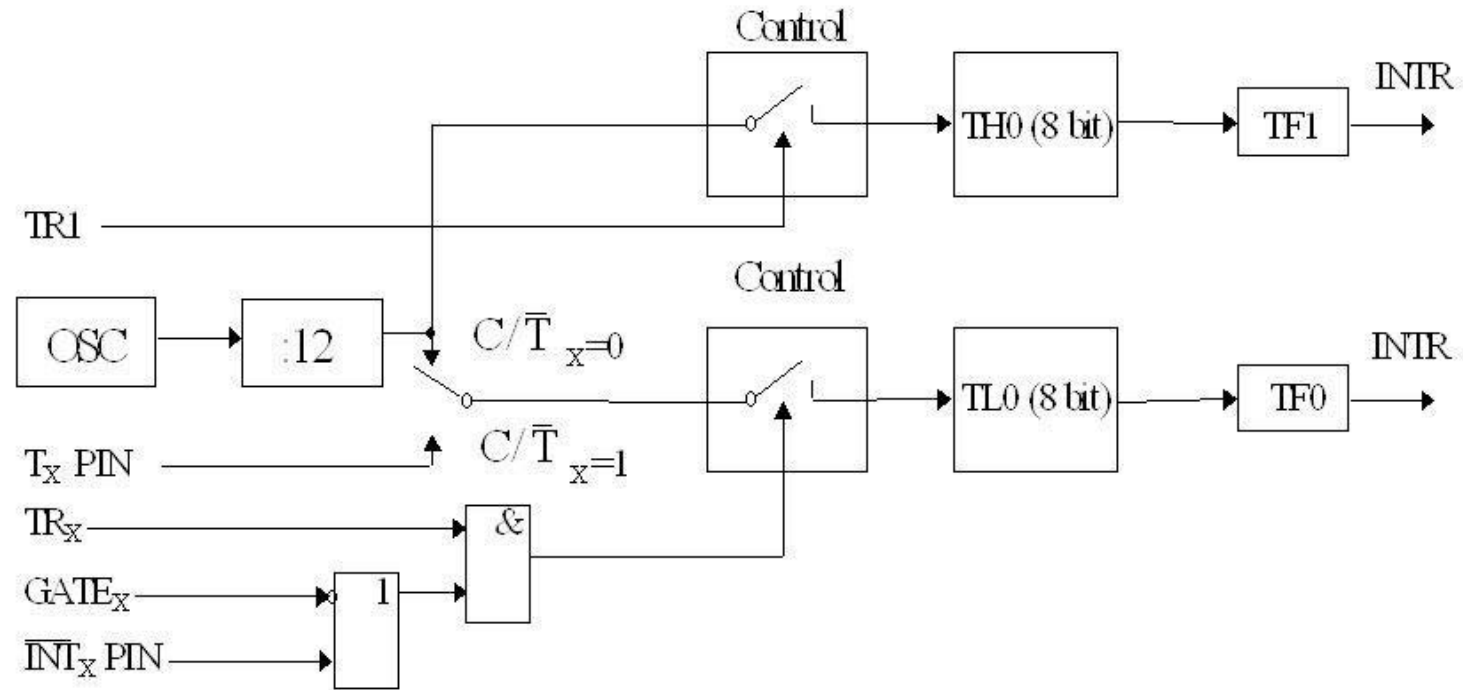
а - логика работы T/C0 и T/C1 в режимах 0 и 1

## Логика работы T/C0 и T/C1 в режиме 2



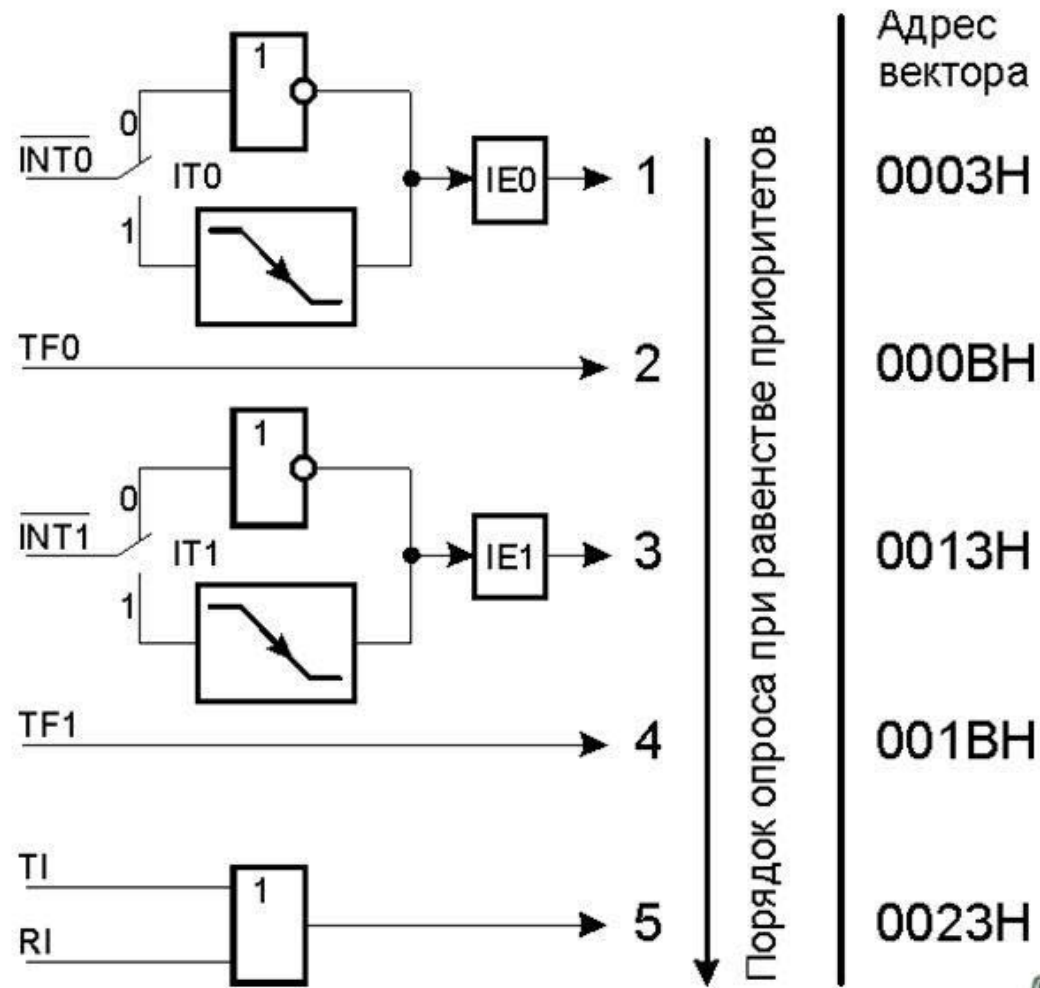
б - логика работы T/C0 и T/C1 в режиме 2

## Логика работы Т/С0 и Т/С1 в режиме 3



в - логика работы Т/С0 в режиме 3

# Лекция 16. Система прерываний микроконтроллера 8051





## Регистр масок прерывания (IE)

Симв.	Позиц.	Имя и назначение
EA	IE.7	Снятие блокировки прерывания. Сбрасывается, программно для запрета всех прерываний независимо от состояний IE.4 - IE.0
	IE.6	Не используется
	IE.5	Не используется
ES	IE.4	Бит разрешения прерывания, от приемопередатчика Установка/сброс программой для разрешения/запрета прерываний от флагов TI или RI .
ET1	IE.3	Бит разрешения прерывания от таймера. Установка/сброс программой для разрешения/запрета прерываний от таймера 1
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерывания 1
ET0	IE.1	Бит разрешения прерывания от таймера 0. Установка/сброс программой для разрешения/запрета прерываний от таймера 0
EX0	IE.0	Бит разрешения внешнего прерывания 0. Установка/сброс программой для разрешения/запрета прерывания 0

## Регистр приоритетов прерываний (IP)

Симв.	Поз.	Имя и назначение
PS	IP.4	Бит приоритета приемопередатчика. Установка/сброс программой для присваивания прерыванию от приемопередатчика высшего/низшего приоритета.
PT1	IP.3	Бит приоритета таймера 1. Установка/сброс программой для присваивания прерыванию от таймера 1 высшего/низшего приоритета.
PX1	IP.2	Бит приоритета внешнего прерывания 1. Установка/сброс программой для присваивания приоритета внешнему прерыванию INT1.
PT0	IP.1	Бит приоритета таймера 0. Установка/сброс программой для присваивания прерыванию от таймера 0 высшего/низшего приоритета.
PX0	IP.0	Бит приоритета внешнего прерывания 0. Установка/сброс программой для присваивания приоритета внешнему прерыванию INT0.



## Пример организации прерываний

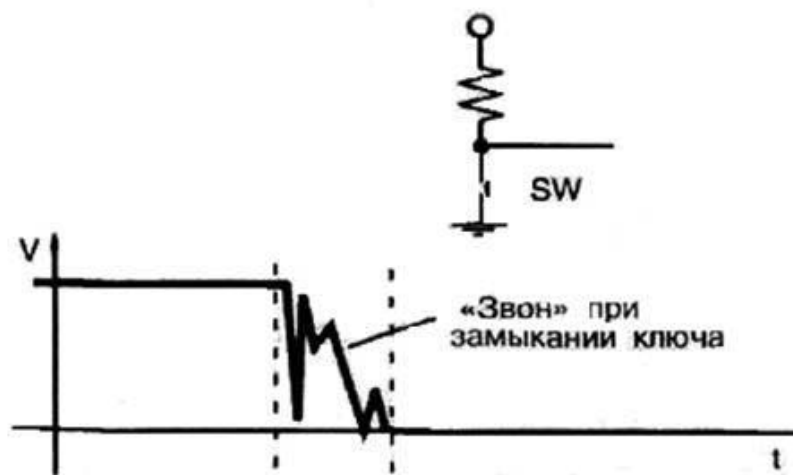
```
org 0h
jmp vkl
org 03h      ; Внешнее прерывание0
mov P3,#1
call Signal
reti
org 0Bh      ; Прерывание таймера0
cpl P3.0
cpl P3.1
reti
org 013h     ; Внешнее прерывание1
call Voditel
reti
org 01Bh     ; Прерывание таймера1
call Schet
reti
org 020h
Signal:
mov TMOD,#00000000b; Таймеры - Режим0
mov TCON,#00010000b; Включаем таймер0
mov IE,#10000010b; Разрешение прерывания только от таймера 0
ret
```



# Лекция 17. Основные типы интерфейсов микроконтроллеров

## Подавление звона контактов

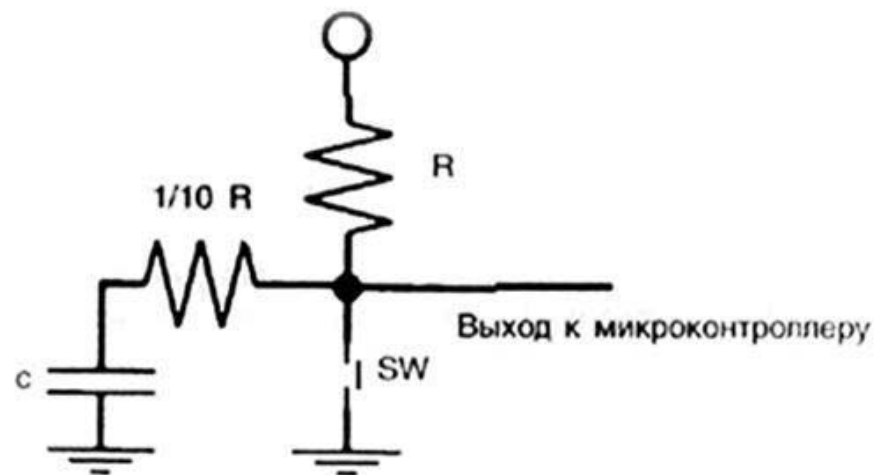
При замыкании и размыкании переключателей в цепи возникают импульсные помехи, вызванные дребезгом контактов. Это явление часто возникает и системах на базе микроконтроллеров, где для ввода данных используется клавиатура, и «звон» может восприниматься как многократное нажатие клавиши.



Дребезг контактов



Один из простых схемотехнических способов устранения звона состоит в подключении RC-цепи.

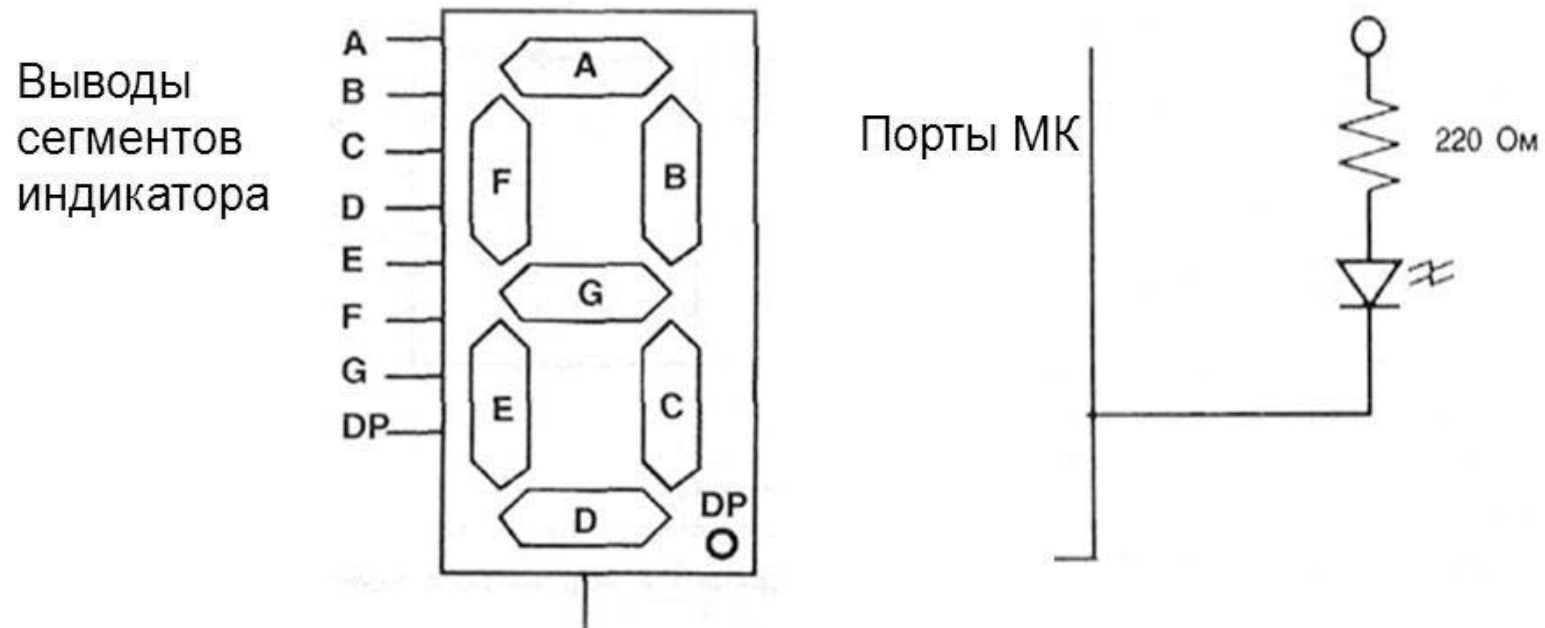


В этой схеме время, требуемое для заряда/ разряда конденсатора до порогового напряжения, маскирует «звон» контактов при переключении.

Недостатки этого метода - дополнительные затраты на компоненты, которые должны быть установлены на плате, и дополнительное время, требуемое для заряда/разряда RC-цепи.



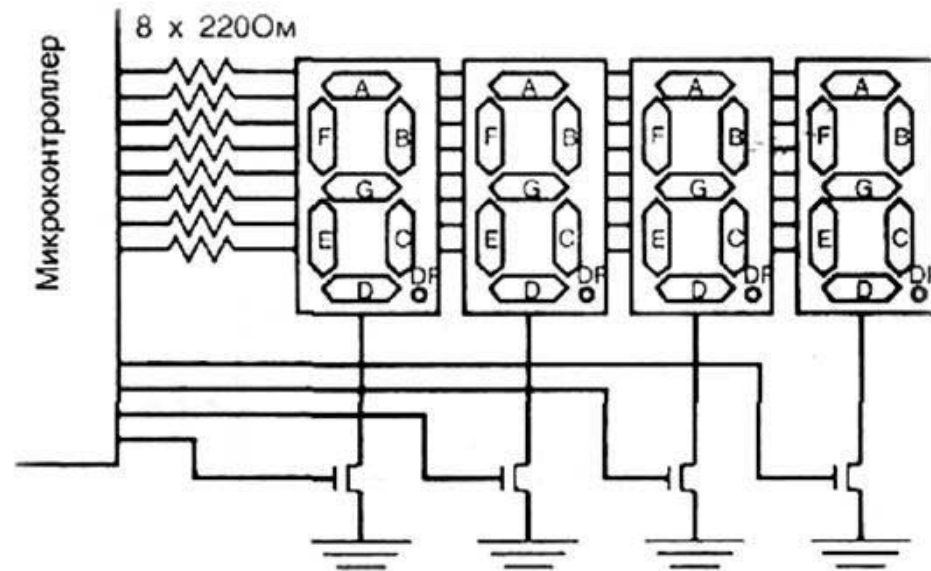
## Подключение светодиодов и 7-сегментных индикаторов



Каждый светодиод в индикаторе имеет свой буквенный идентификатор ( A, B, C, D, E, F или G), и одна из ножек светодиода подключена к соответствующему внешнему выводу. Вторые ножки всех светодиодов соединены вместе и подключены к общему выводу. Этот общий вывод определяет тип индикатора: с общим катодом или с общим анодом.

## Подключение к микроконтроллеру четырех 7-сегментных индикаторов

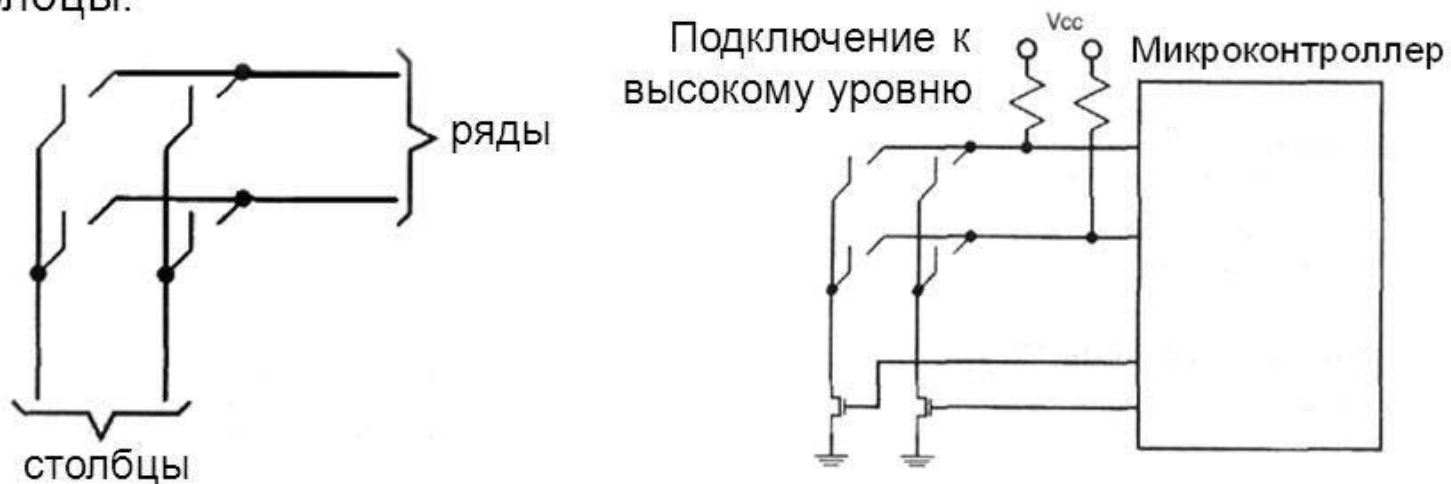
Типичный способ подключения нескольких индикаторов состоит в том, чтобы включить их параллельно и затем управлять протеканием тока через общие выводы отдельных индикаторов. Так как величина этого тока обычно превышает допустимое значение выходного тока микроконтроллера, то для управления током включаются дополнительные транзисторы, которые выбирают, какой из индикаторов будет находиться в активном состоянии.



## Лекция 18. Подключение клавиатуры и дисплея

### Ввод с матричной клавиатуры

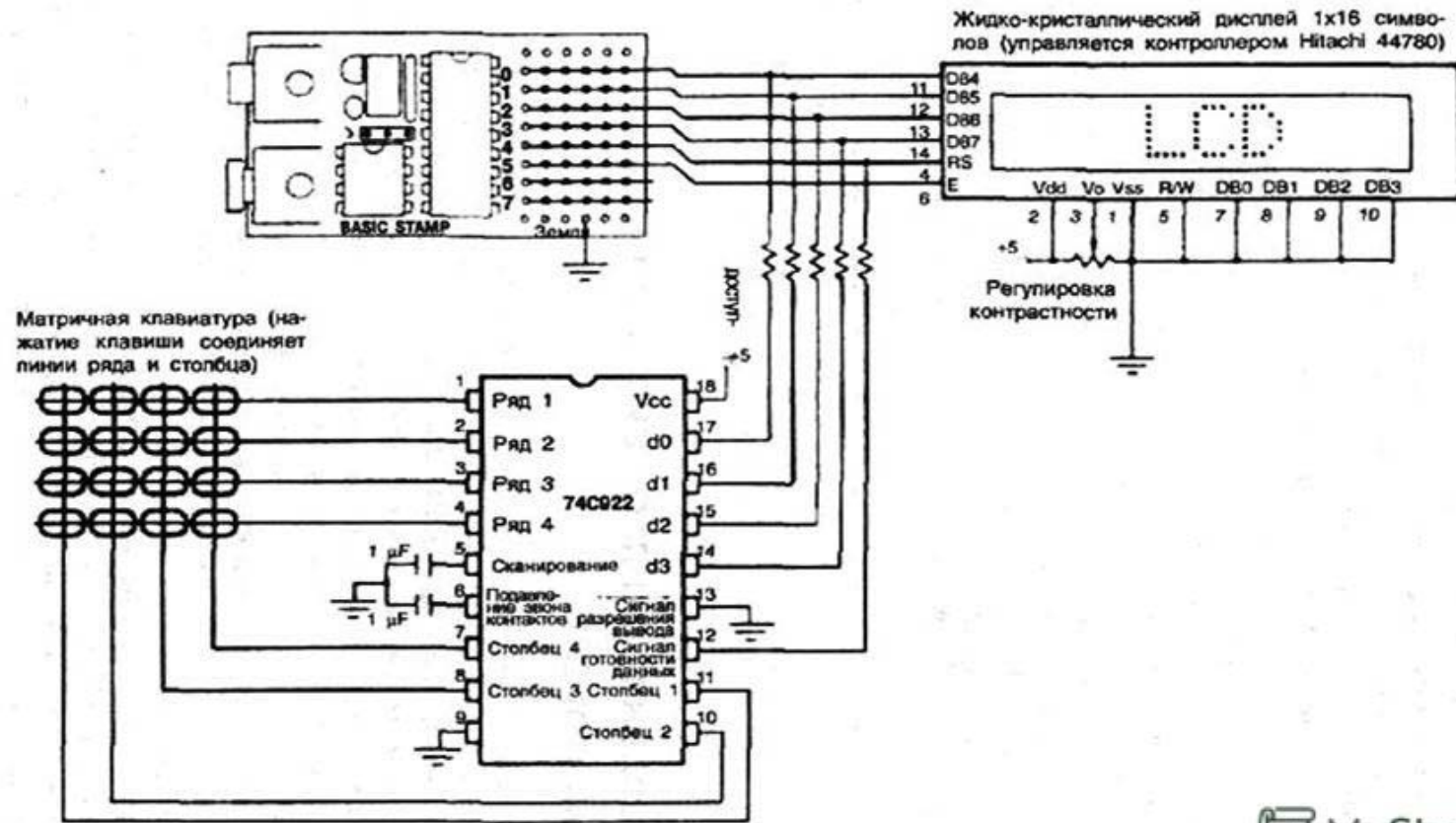
Лучшим решением является использование матричной клавиатуры, которая представляет собой набор ключей, объединенных в ряды и столбцы.



Для чтения состояния определенного ключа на столбец подается сигнал, а затем считывается состояние рядов. Обычно ряды подключают к высокому потенциалу, а опрашиваемый столбец соединяется с землей. Если при сканировании рядов считывается низкий уровень сигнала, это означает, что ключ в данной позиции ряд/столбец замкнут.

Существуют специальные микросхемы, которые выполняют опрос клавиатуры и посылают сигнал, когда данные готовы для считывания.

На схеме показано использование микросхемы типа 74С922 в качестве интерфейса между клавиатурой с матрицей 4x4 ключа и микроконтроллером BASIC Stamp.





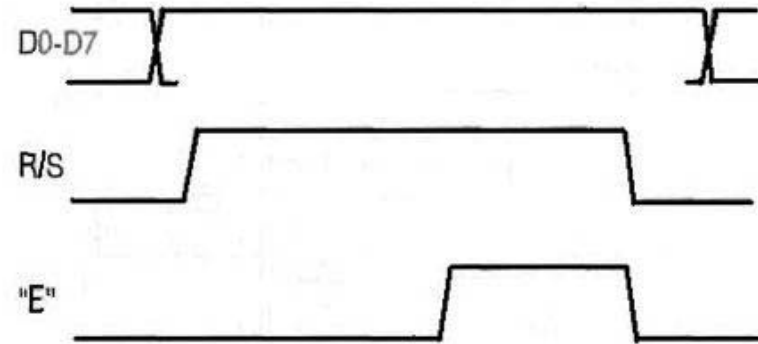
## Управление жидкокристаллическим индикатором

Большинство алфавитно-цифровых ЖКИ используют для управления контроллер Hitachi 44780 и реализуют общий интерфейс подключения. Благодаря этому ЖКИ, обеспечивающие вывод от 8 до 80 символов (организованных в виде 2 строк по 40 символов или 4 строк по 20 символов), являются полностью взаимозаменяемыми, так как их применение не требует какого-либо изменения программного обеспечения или аппаратных средств. Выводы ЖКИ имеют следующее назначение:

1. вывод 1 — «Земля»;
2. вывод 2 — напряжение питания  $V_{cc}$ ;
3. вывод 3 — вход регулировки контрастности изображения;
4. вывод 4 — сигнал выбора регистра данных или команд (R/S);
5. вывод 5 — сигнал выбора режима «чтение/запись» (R/W);
6. вывод 6 — синхросигнал E;
7. выводы 7-14 - линии передачи данных.

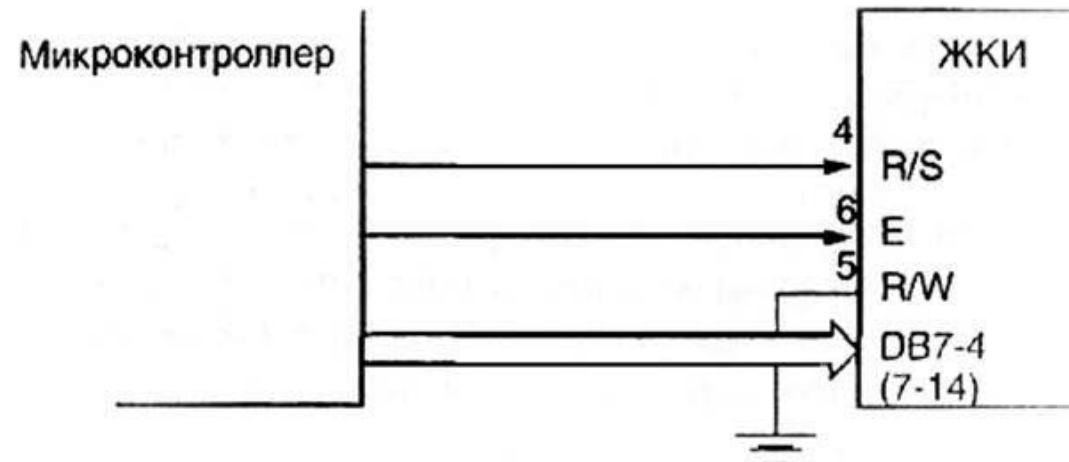


## Временные диаграммы сигналов при выводе символа на ЖКИ



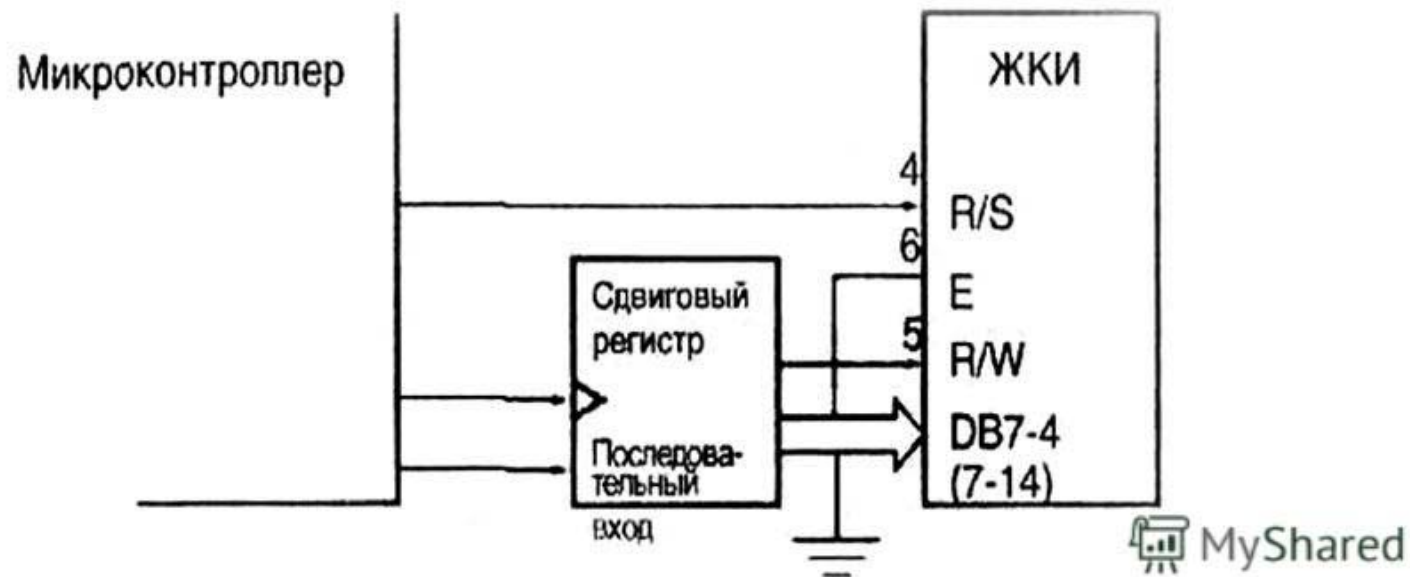
Каждая посылка сопровождается синхросигналом E, который инициирует прием данных в ЖКИ.

Чтобы подсоединить микроконтроллер к ЖКИ при четырехбитном режиме используются только 4 старших разряда линии данных DB7-4



## Управление жидкокристаллическим индикатором

Дальнейшее сокращение числа требуемых линий ввода-вывода может быть обеспечено путем использования сдвигового регистра: в этом случае потребуется всего 3 линии. В качестве сдвигового регистра обычно используется микросхема 74х174 (где «х» — или HC, или LS). Восьмибитный режим также можно реализовать с помощью сдвигового регистра, но требуется передавать девятый бит, который используется, чтобы обеспечить выдачу сигнала R/S. Бит R/S указывает, какая информация передается - команда или данные.



## Управление жидкокристаллическим индикатором

Имеется восемь программируемых символов, которые выводятся с помощью кодов с 0x000 по 0x007. Эти символы программируются с помощью команд, устанавливающих курсор ЖКИ на область памяти генератора символов («СGRAM»), и задают восемь значений адреса для строчной записи изображения символа. Следующие восемь байтов, записанные в память, представляют собой изображение каждой строки программируемого символа, начиная сверху. Набор команд, реализуемых ЖКИ:

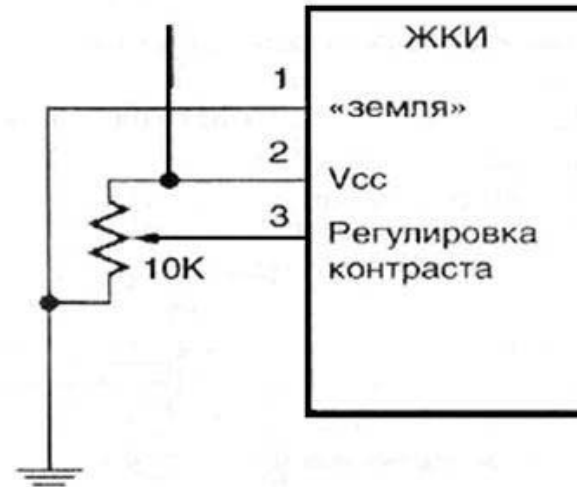
- очистить индикатор;
- вернуть курсор в начальную позицию;
- установить направление движения курсора;
- разрешить экран/курсor;
- переместить курсор/сдвинуть экран;
- установить размерность интерфейса;
- переместить курсор на область cgram;
- переместить курсор на экран;
- прочесть флаг «занято»;
- вывести ascii-символ на экран;
- прочесть ascii-символ с экрана.



## Управление жидкокристаллическим индикатором

В большинстве применений линию «R/W» подсоединяют к земле, так как чтение состояния ЖКИ не требуется. Это значительно упрощает приложение, поскольку для считывания данных необходимо менять режим работы выводов — с записи на чтение. В некоторых случаях возможность чтения состояния ЖКИ бывает полезна, например, при прокручивании данных на экране. Подключение линии «R/W» к земле также освобождает один вывод микроконтроллера.

ЖКИ с размером символов 5x10 точек практически не выпускаются, поэтому бит «F» в команде «Установка размерности интерфейса» должен всегда быть равен 0.



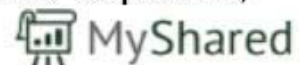


## Управление жидкокристаллическим индикатором (продолжение)

Перед тем, как вводить в ЖКИ команды или данные, его надо инициализировать.

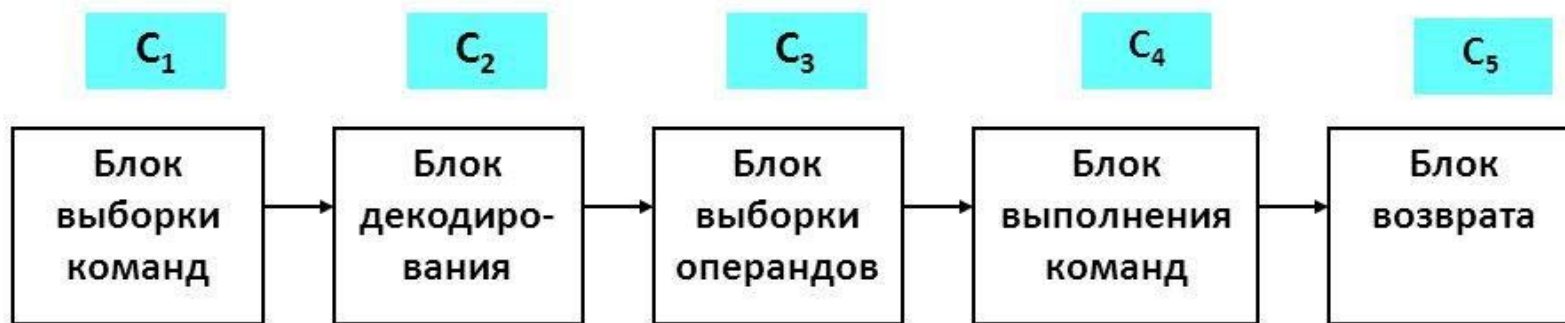
Для 8-битного режима:

1. Подождать более 15 мс после подачи питания.
2. Записать 0x30 в ЖКИ и ждать 5мс до завершения выполнения команды.
3. Записать 0x30 в ЖКИ и ждать 160мкс до завершения выполнения команды.
4. Снова записать 0x30 в ЖКИ и ждать 160мкс до завершения выполнения команды или опрашивать флаг «Занято».
5. Установить рабочие характеристики ЖКИ:
  - Ввести «Установка размерности интерфейса»
  - Ввести 0x10, чтобы выключить экран.
  - Ввести 0x01, чтобы очистить экран.
  - Ввести «Установка направления движения курсора», чтобы установить поведение курсора.
  - Ввести «Включение экрана/курсора», чтобы включить экран и, если требуется, курсор.

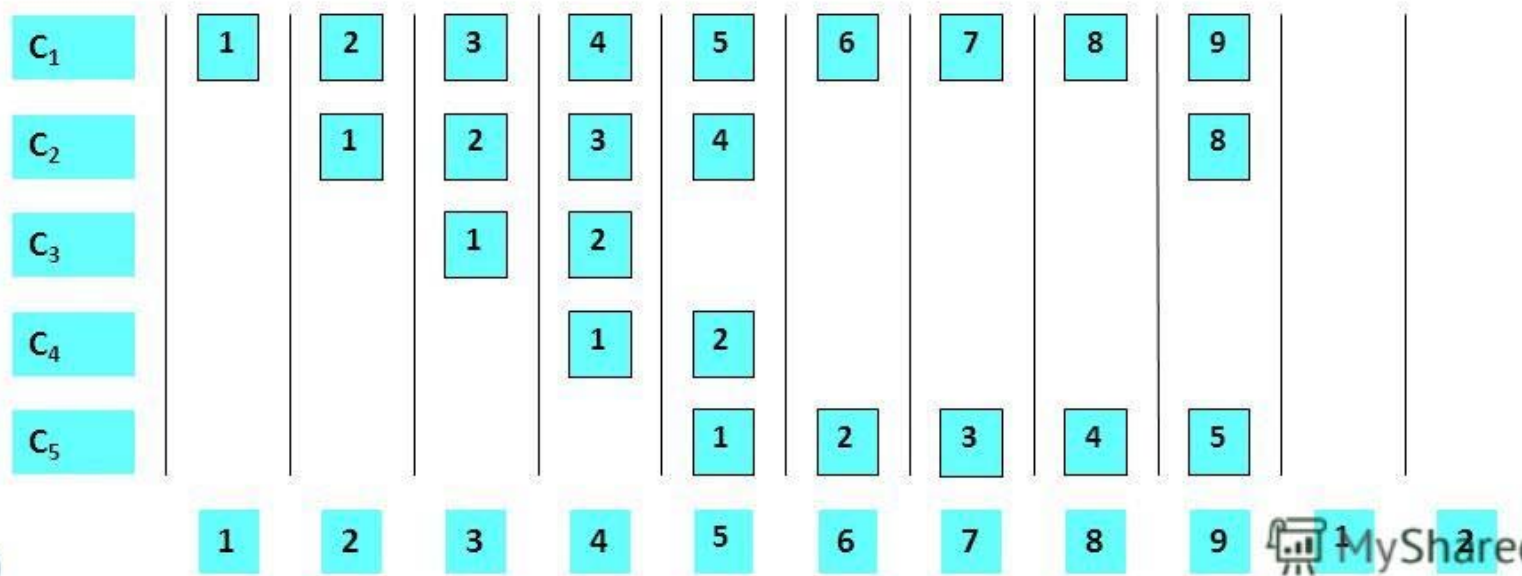




# Лекция 19. Конвейеризация в микропроцессорных системах



Конвейер с пятью стадиями



# Блок-схема архитектуры Pentium

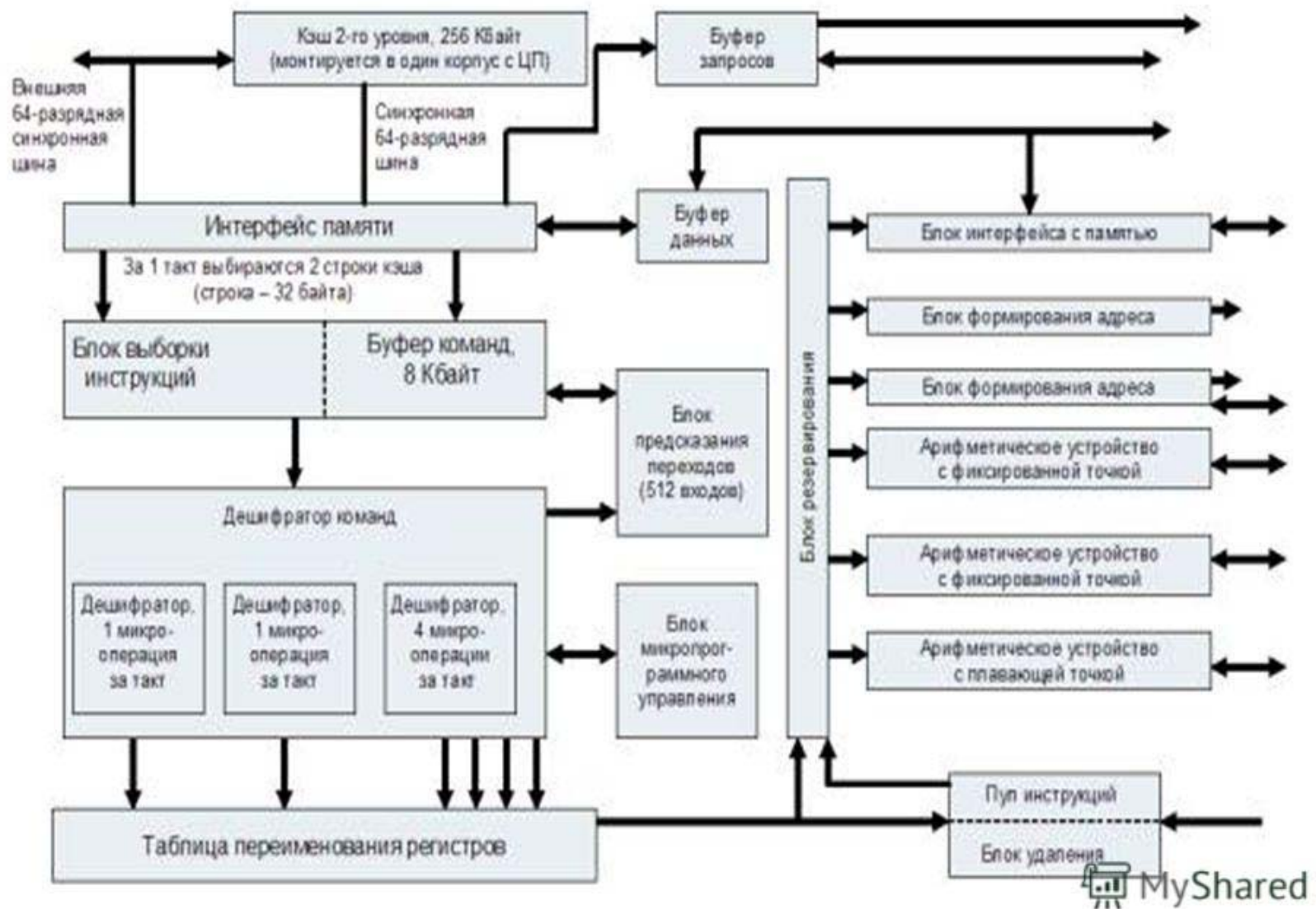


- пятиступенчатый конвейер для обработки команд:
- PF (Prefetch) - предвыборка команд;
  - D1 (Instruction Decode) - декодирование команды;
  - D2 (Address Generate) - формирование адреса;
  - EX (Execute) - выполнение команды в АЛУ и доступ к кэш-памяти;
  - WB (Write Back) - обратная запись.

Таблица 5.2. Конвейеризация команд в МП Pentium

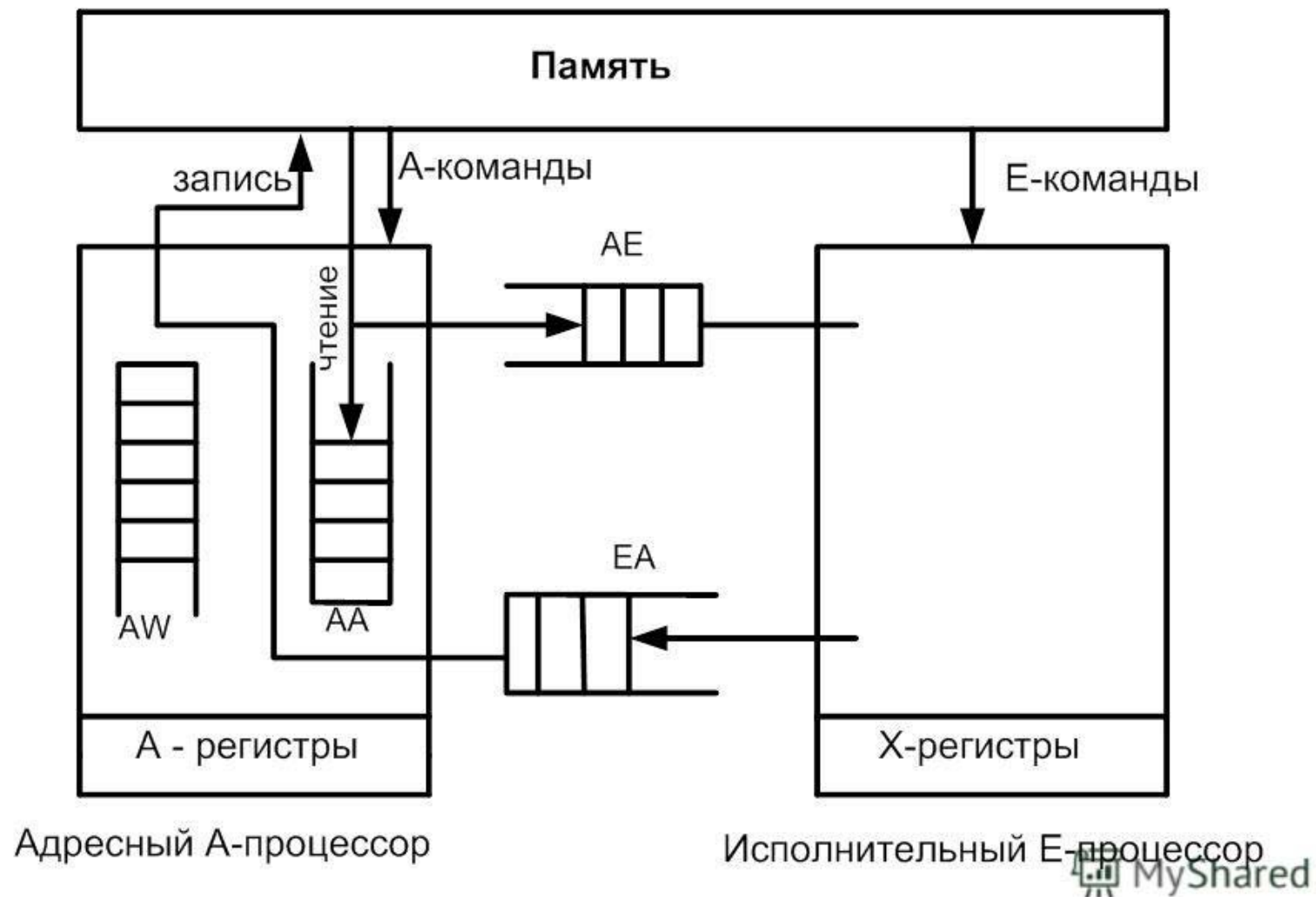
Этапы конвейеров	Циклы шины							
PF	K1	K3	K5	K7				
	K2	K4	K6	K8				
D1		K1	K3	K5	K7			
		K2	K4	K6	K8			
D2			K1	K3	K5	K7		
			K2	K4	K6	K8		
EX				K1	K3	K5	K7	
				K2	K4	K6	K8	
WB					K1	K3	K5	K7
					K2	K4	K6	K8





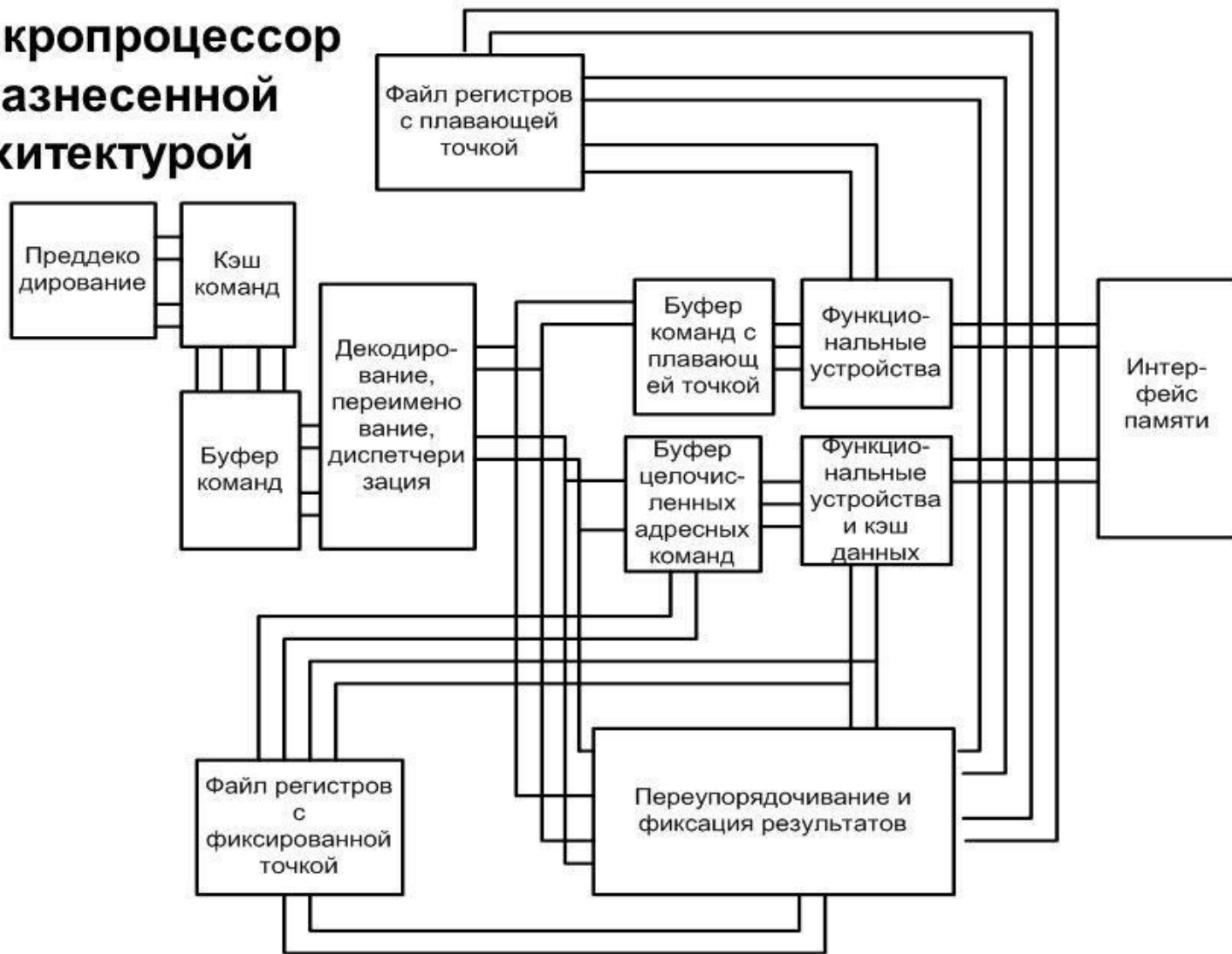
MyShared

## Микропроцессор с разнесенной архитектурой

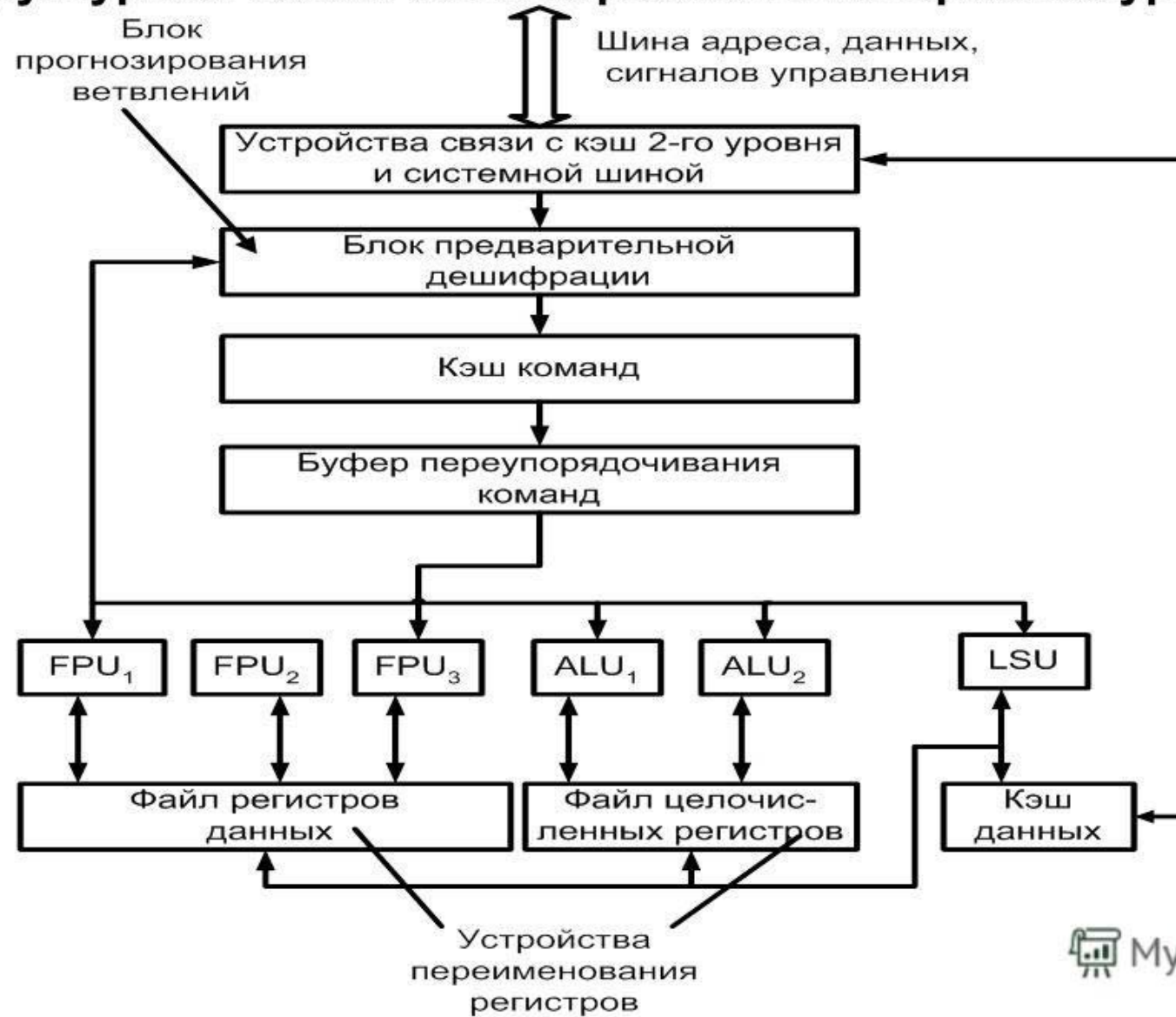


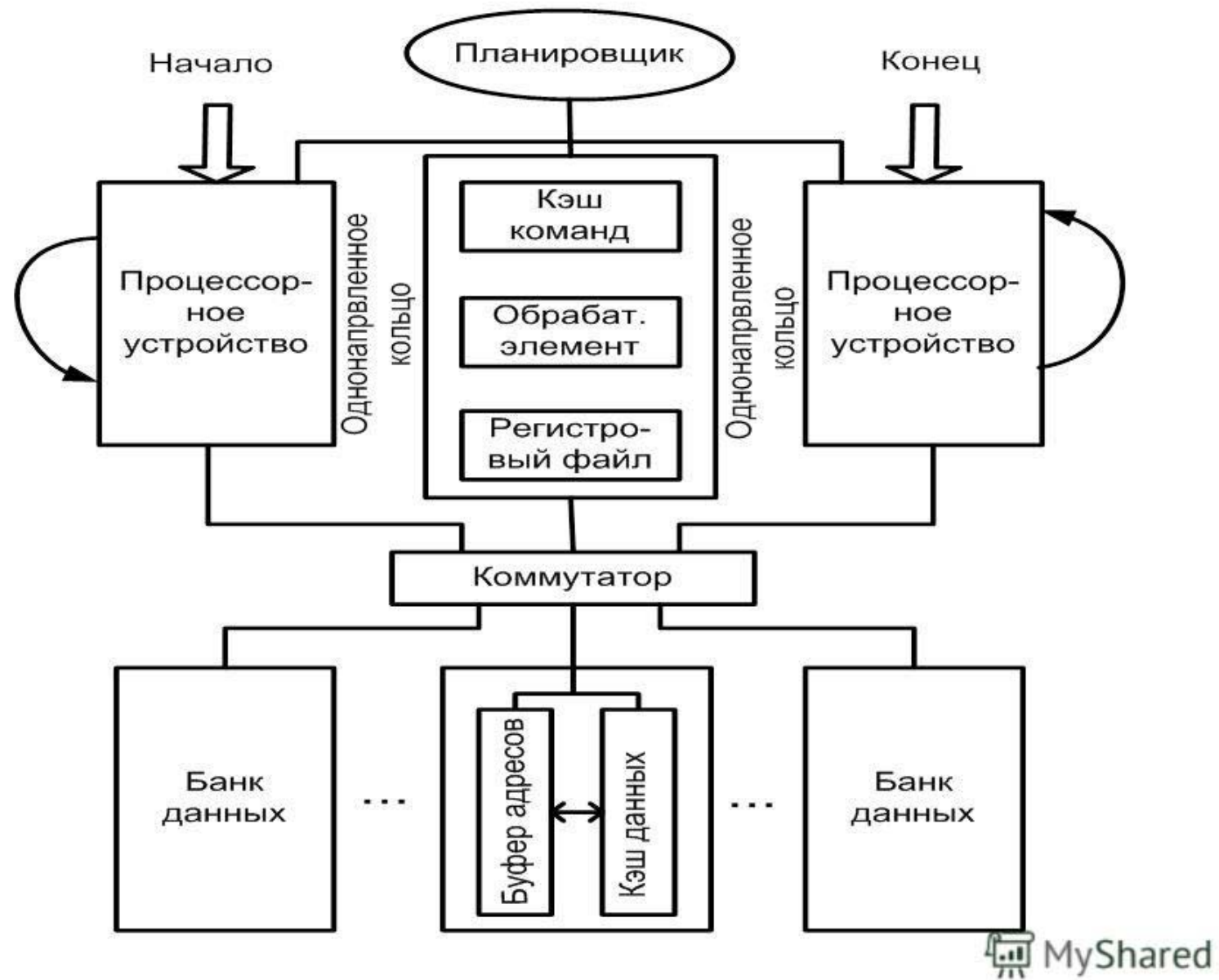


# Микропроцессор с разнесенной архитектурой

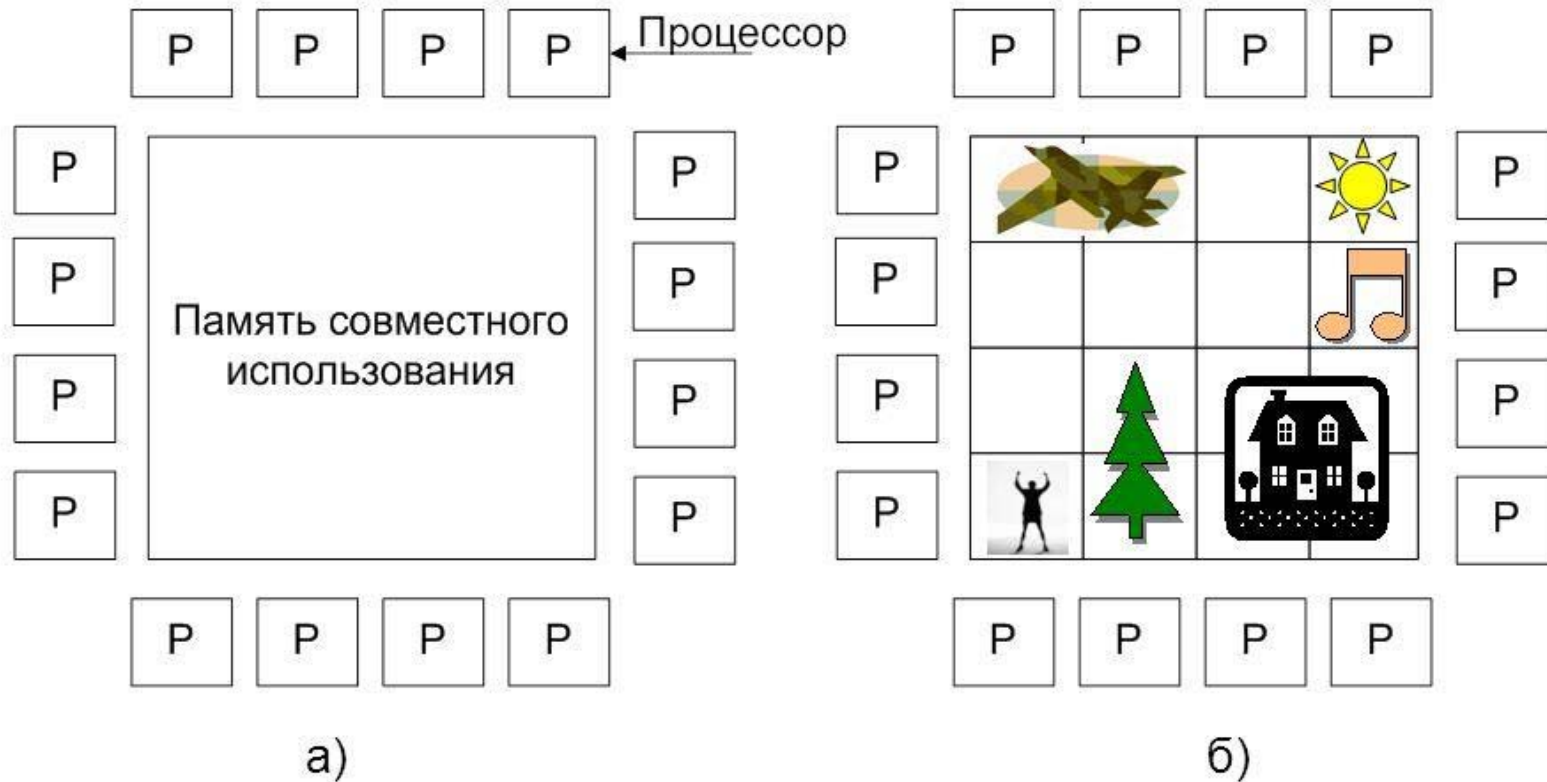


# Структурная схема МП с параллельной архитектурой





## Лекция 20. Мультипроцессоры



Мультипроцессор, содержащий 16 процессоров, которые разделяют общую память (а); изображение, разбитое на 16 секций, каждую из которых анализирует отдельный процессор (б).

# Мультикомпьютеры



Мультикомпьютер, содержащий 16 процессоров, каждый из которых имеет свою собственную память (а); битовое отображение рисунка предыдущего слайда, разделенное между 16 участками памяти (б)



## Лекция 21. Классификация компьютеров параллельного действия (классификация Флинна)

На основе числа потоков команд и потоков данных Флинн выделяет четыре класса архитектур:

**SISD, MISD, SIMD, MIMD.**

- 1) SISD (single instruction stream / single data stream) – одиночный поток команд и одиночный поток данных;
- 2) SIMD (single instruction stream / multiple data stream) – одиночный поток команд и множественный поток данных.
- 3) MISD (multiple instruction stream / single data stream) – множественный поток команд и одиночный поток данных.
- 4) MIMD (multiple instruction stream / multiple data stream) – множественный поток команд и множественный поток данных.



## Классификация компьютеров параллельного действия (классификация Флинна)

Потоки команд	Потоки данных	Названия	Примеры
1	1	SISD	Классическая машина фон Неймана
1	Много	SIMD	Векторный процессор, массивно параллельный процессор
Много	1	MISD	Не существует
Много	Много	MIMD	Мультипроцессор

# Расширенная классификация Флинна



## Расширенная классификация Флинна

1. **UMA** - Uniform Memory Access – архитектура с однородным доступом к памяти;
2. **NUMA** - Non Uniform Memory Access – архитектура с неоднородным доступом к памяти;
3. **COMA** - Cache Only Memory Access – архитектура с доступом только к кэш-памяти.

Так как мультимпьютеры не имеют прямого доступа к отдельным модулям памяти, они иногда называются машинами NORMA- NO (Remote Memory Access) – без доступа к отдаленным модулям памяти.

Мультимпьютеры можно разделить на две категории.

Первая категория содержит процессоры MPP (Massively Parallel Processors – процессоры с массивным параллелизмом).

Вторая категория мультимпьютеров включает рабочие станции, которые связываются с помощью уже имеющихся технологий соединений. Эти примитивные машины называются NOW (Network of Workstations – сеть рабочих станций) и COW (Cluster of Workstations – кластер рабочих станций).





## Лекция 22. Транспьютеры. Архитектура

Транспьютер (transputer) — элемент построения многопроцессорных систем, выполненный на одном кристалле сверхбольшой интегральной схемы, продукт английской компании INMOS Ltd (ныне - подразделение [STMicroelectronics](#)). Другая интерпретация - транспьютер = передатчик (transfer) + вычислитель (computer) - указывает на присутствие встроенных скоростных устройств ввода/вывода для обмена данными с соседними.



32 битный транспьютерный чип IMST414B