

**MYCY**

2020

# Лекция 1

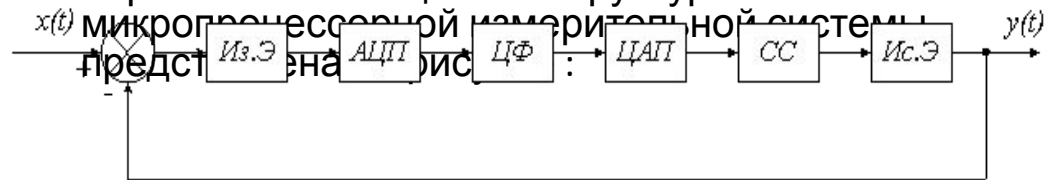
Элементы дискретных систем

# Общие сведения о микропроцессорных системах управления

В последние годы методы цифровой обработки сигналов приобретают все большую важность, так как теперь не только заменяют классические аналоговые методы во многих традиционных областях техники, но и применяются во многих новых областях. Это объясняется тем, что цифровые системы, по сравнению с аналоговыми, обладают рядом преимуществ:

- повышенная чувствительность;
- большая надежность;
- отсутствие дрейфа;
- более высокая устойчивость к шумам и возмущениям;
- удобство программирования.

- Из всех методов, используемых для цифровой обработки сигналов, наиболее важным является цифровая фильтрация, что объясняется следующими причинами:
- - наличие эффективных и относительно простых методов построения фильтров;
- - большими успехами в области технологии микросхем с высокой степенью интеграции;
- - успехами в части разработки технических средств и программного обеспечения ЭВМ, особенно с появлением микропроцессоров.
- Одним из существенных преимуществ цифровых фильтров является их большая гибкость по сравнению с аналоговыми. Программа цифрового фильтра может быть изменена или приспособлена к характеристикам объекта без каких-либо изменений в аппаратной части, что дает возможность использовать одну и ту же аппаратную основу для различных систем управления.
- Устройство цифровой обработки сигналов преобразует входной сигнал с помощью средств вычислительной техники, в соответствии с заданной задачей и заданным алгоритмом обработки. Обобщенная структурная схема



## Дискретизация

Сигнал с измерительного элемента (Из.Э) подается на аналого-цифровой преобразователь (АЦП), который осуществляет дискретизацию входного аналогового сигнала  $x(t)$  (рисунок а) по времени, квантование и кодирование выборочных значений сигнала, следующих с периодом дискретизации  $T$ . После дискретизации получается дискретный сигнал  $X_d(nT)$  (рисунок), где  $n$  – номер отсчета. При кодировании каждое значение  $X_d(nT)$  представляется кодом, состоящим из конечного числа двоичных разрядов. Таким образом, на выходе АЦП имеем цифровой сигнал  $X(nT)$ .

Сигнал с АЦП подается на цифровой фильтр. На выходе цифрового фильтра формируется цифровой сигнал  $Y(nT)$ .

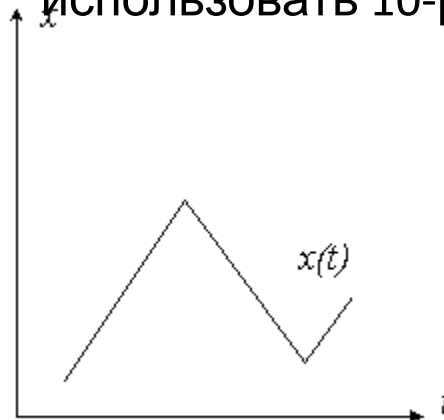
Преобразование сигналов выполняется в соответствии с алгоритмом.

$$Y(nT) = A[x(nT)] \quad (2.1)$$

Алгоритм  $A$  является основой синтеза структуры цифрового фильтра.

- Цифро-аналоговый преобразователь (ЦАП) и схема сглаживания (СС) служат для преобразования последовательности чисел в выходной аналоговый сигнал, подаваемый на исполнительный элемент (Ис.Э). ЦАП и АЦП являются соответственно выходными и входными узлами для ЦФ. К основным параметрам ЦАП и АЦП относится число двоичных разрядов, которое определяет точность вычислений. Чтобы обеспечить точность преобразования 0,1%, необходимо использовать 10-разрядные АЦП.

а)



б)

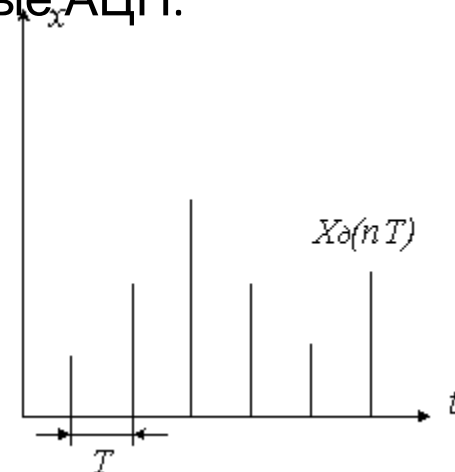
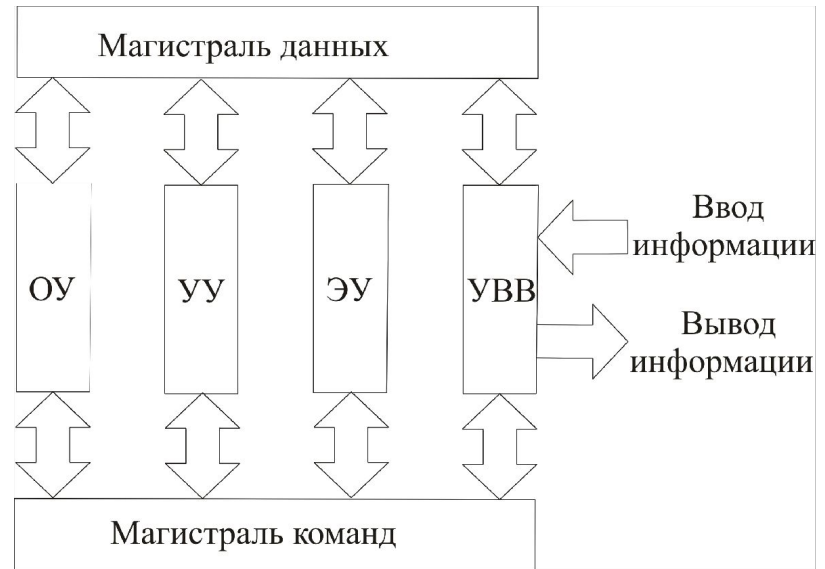
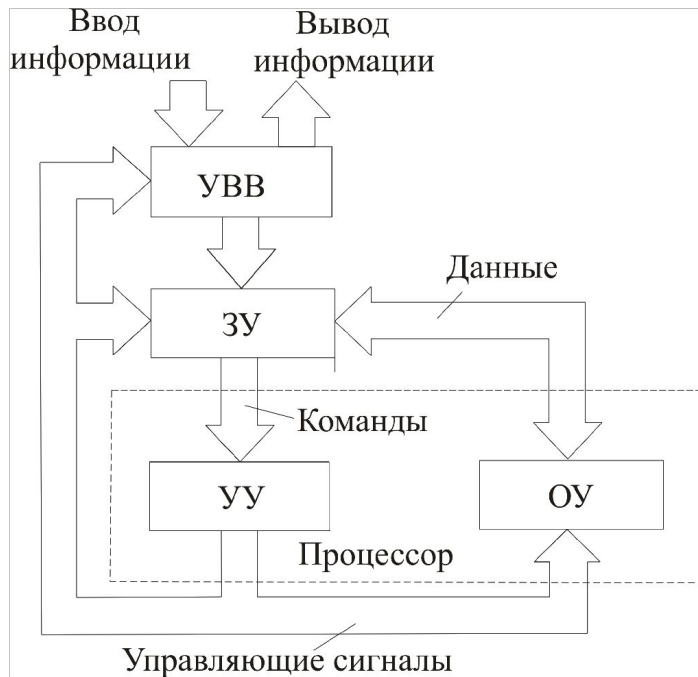


Рис.2.1. Классическая а) и магистральная б) структуры цифровых систем.

структуры цифровых систем.



# Потребляемая мощность

- Основными параметрами цифровых систем являются *мощность* потребляемая от источников питания, и *производительность*.
- Потребляемая мощность определяется выражением

- $$P = \sum_i E_i I_{\pi i}$$

- где  $E_i$  - напряжение и  $I_{\pi i}$  - средний ток питания для  $i$ -го узла (блока); сумма берется по всем узлам (блокам), входящим в систему.
- Единицей производительности является скорость выполнения арифметических или других операций: число операций, выполненных в секунду (оп/с).

# Производительность универсальных ЭВМ

- Для сложения или вычитания двоичных чисел («короткие» операции) требуется обычно 4-5 тактов машинного времени, а для умножения или деления («длинные» операции) - на 1-2 порядка больше. Поэтому при расчете производительности используются различные способы усреднения. Например, производительность универсальных ЭВМ часто оценивается по формуле

- $$W_{\text{ПР}} = 1 / (0,7t_k + 0,3t_{\text{Д}})$$

- где  $t_k$  - времена выполнения «коротких» и «длинных» операций; 0,7 и 0,3 - коэффициенты, учитывающие относительную частоту выполнения этих операций при реализации типовых алгоритмов (операции типа «сложение» составляют около 70%, типа «умножение» - около 30%).

- Как показатель качества системы часто используют отношение (скорость/мощность), которое определяет научно-технический уровень ее структурной, схемотехнической и конструкторско-технологической разработок.



- Важнейшей характеристикой системы является также ее надежность, определяемая *временем безотказной работы*  $T_0$  [ч] или *средней частотой отказов*  $k$  [от/ч] =  $1/T_0$ . Для многих областей применения весьма важными являются такие параметры, как *масса и габариты*.

- Для больших цифровых систем, например высокопроизводительных вычислительных комплексов, характерен переход к распределенной структуре. При этом каждое из устройств системы в миниатюре повторяет структуру целой цифровой системы.
- Так, например, УВВ включает в себя специализированный процессор, выполняющий предварительную подготовку информации для обработки, и собственное ЗУ.

- Особым классом являются *мультипроцессорные системы*, процессорная часть которых состоит из нескольких процессоров. Такая структура обеспечивает повышение производительности систем за счет распределения между различными процессорами отдельных этапов решения задачи и одновременного их выполнения.

- . Мини- и микро-ЭВМ обычно имеют *магистральную структуру* (рис. 2.1, б), при которой обмен информацией между всеми устройствами производится по одной или нескольким общим линиям связи - *магистралям*, В результате достигается относительная простота системы, однако производительность системы снижается, так как одновременно могут обмениваться информацией только два устройства.

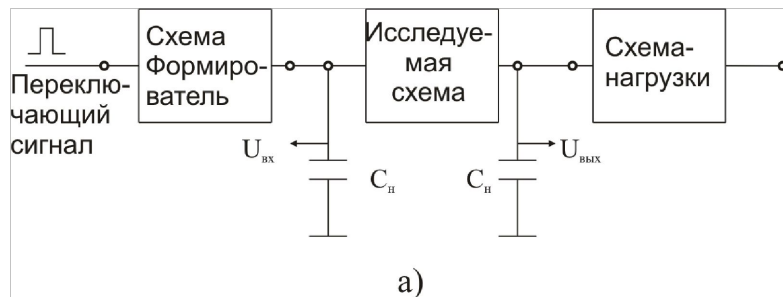
# Классификация и основные параметры цифровых микросхем

- Современная технология позволяет изготавливать микросхемы, содержащие десятки тысяч логических элементов. Сложность цифровой микросхемы характеризуется степенью функциональной интеграции:

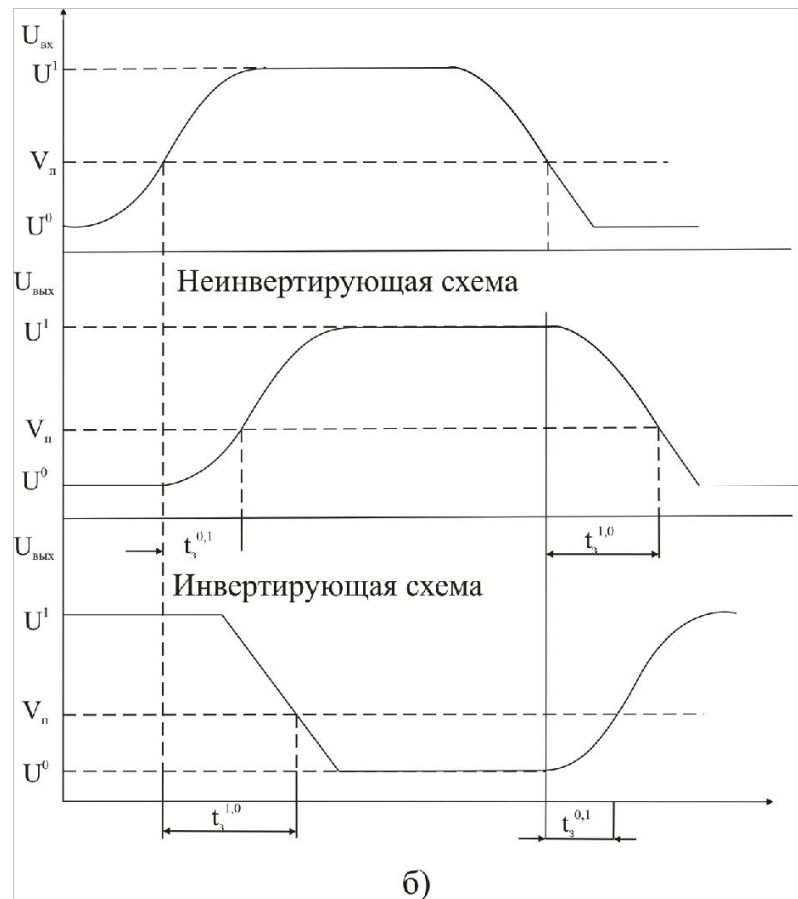
$$K_{и} = \lg N_{эл}$$

- где  $N$  - число элементов И - НЕ либо ИЛИ - НЕ, расположенных на кристалле микросхемы.

# Логическая схема измерения (а) и их переходные характеристики (б).



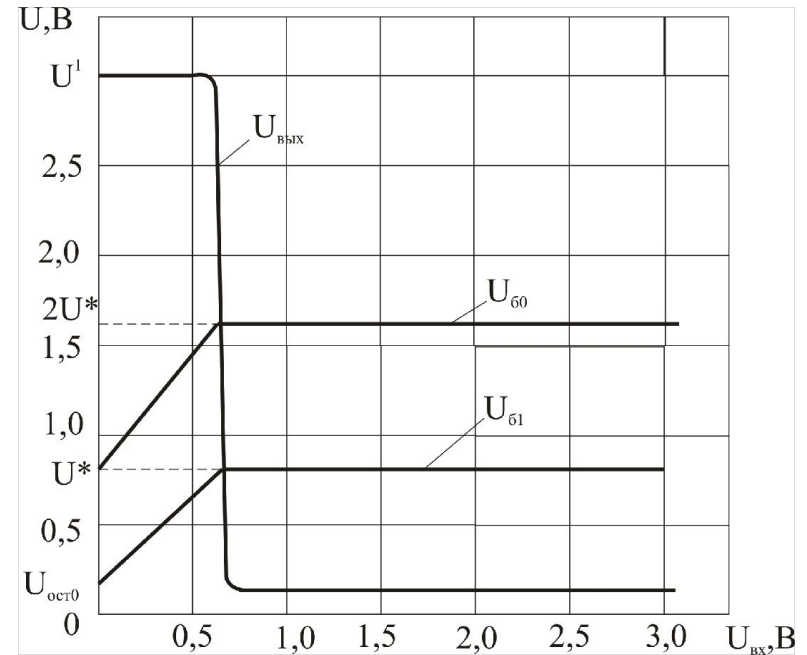
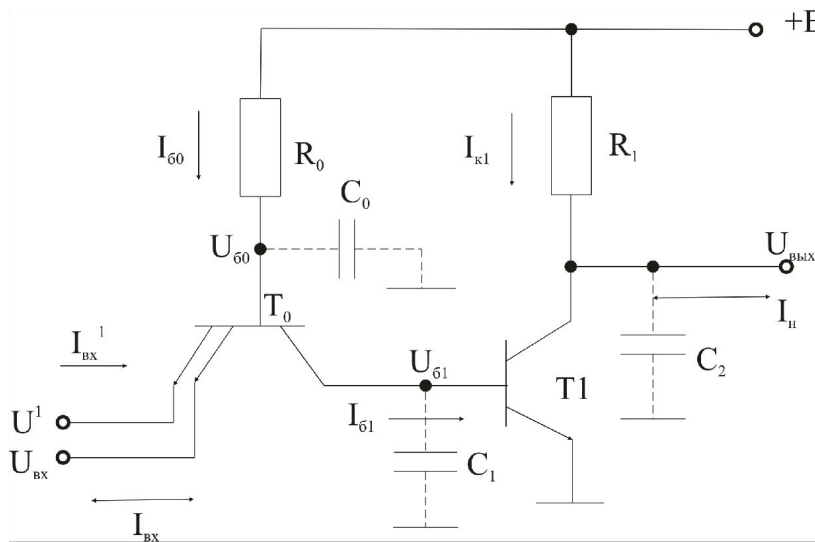
а)



б)

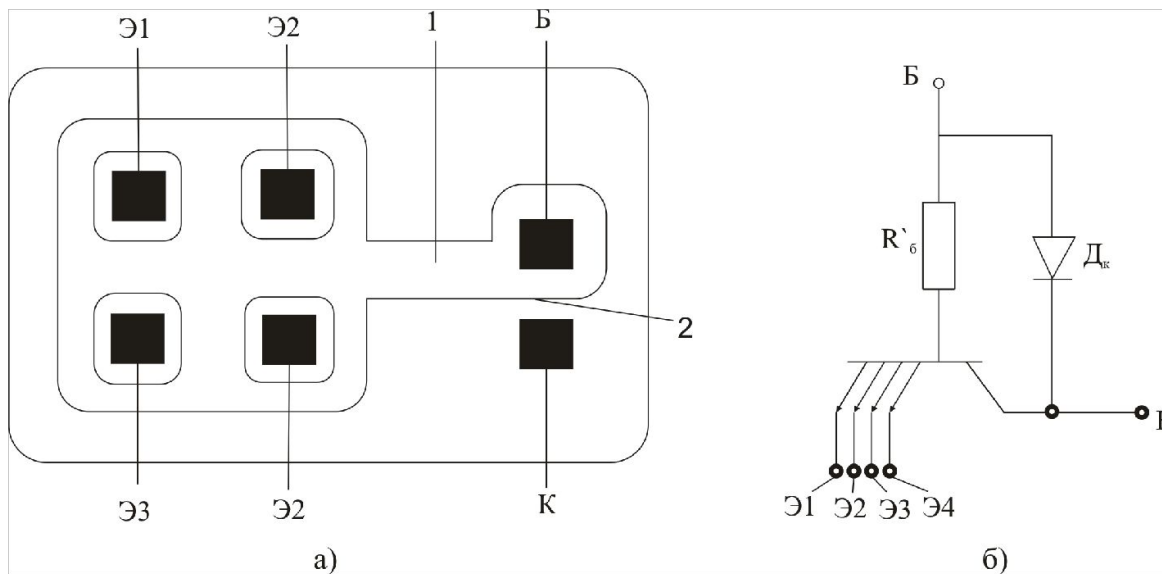
# Элементы транзисторно-транзисторной логики (ТТЛ)

- Типовая передаточная характеристика и зависимости напряжений на базах транзисторов  $T_0$ ,  $T_1$  от потенциала  $U_{вх}$  при  $E=3В$ ,  $T=+20^{\circ}C$ ,  $n=1$ .



# Топология одного из используемых вариантов МЭТ

- Многоэмиттерный транзистор
- а) топология; б) эквивалентная схема.





# Переходные процессы в элементе ТТЛ с простым инвертором.

Процесс переключения элемента

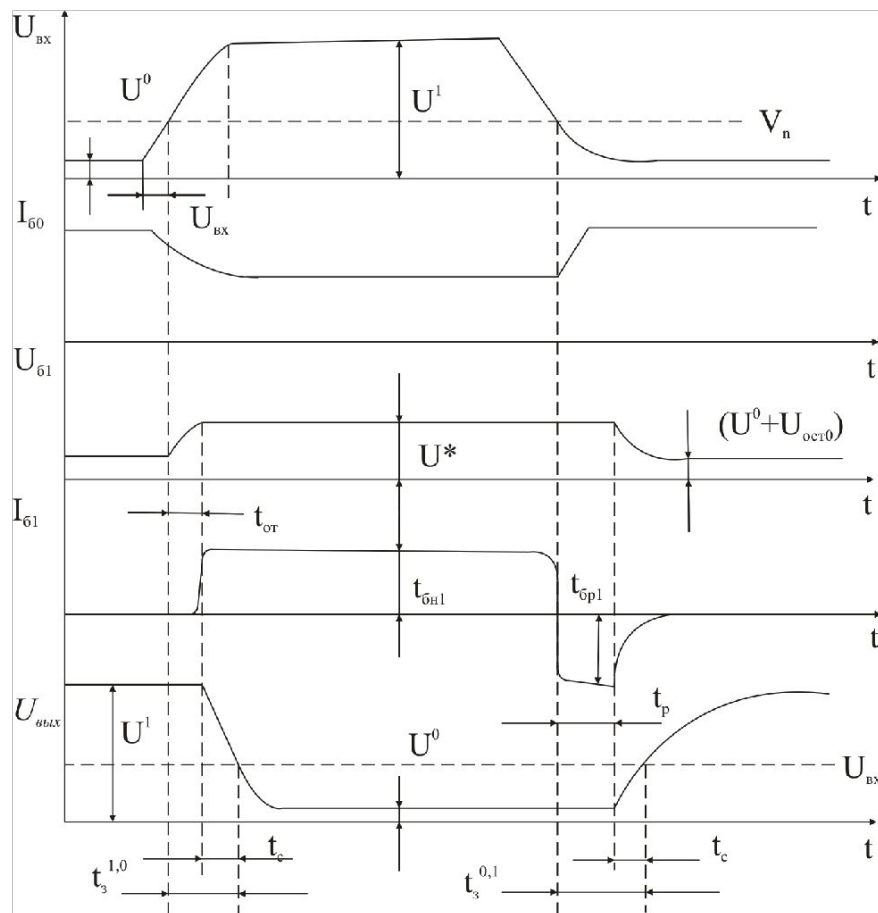
состоит из нескольких этапов:

**-Спад выходного потенциала до порога переключения  $V_n$ ;**

**-Рассасывание избыточного Заряда;**

**-Нарастание выходного**

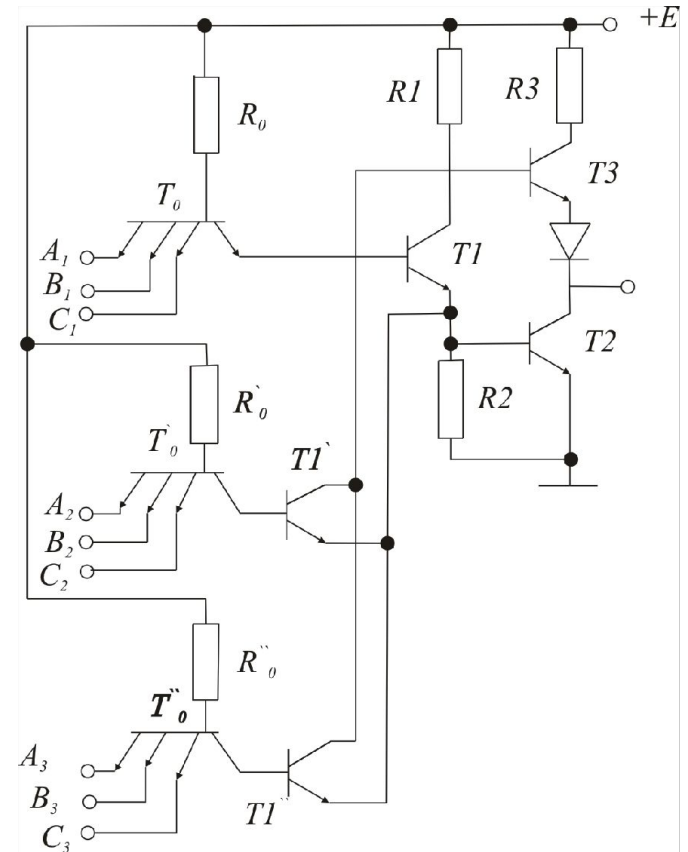
**потенциала**





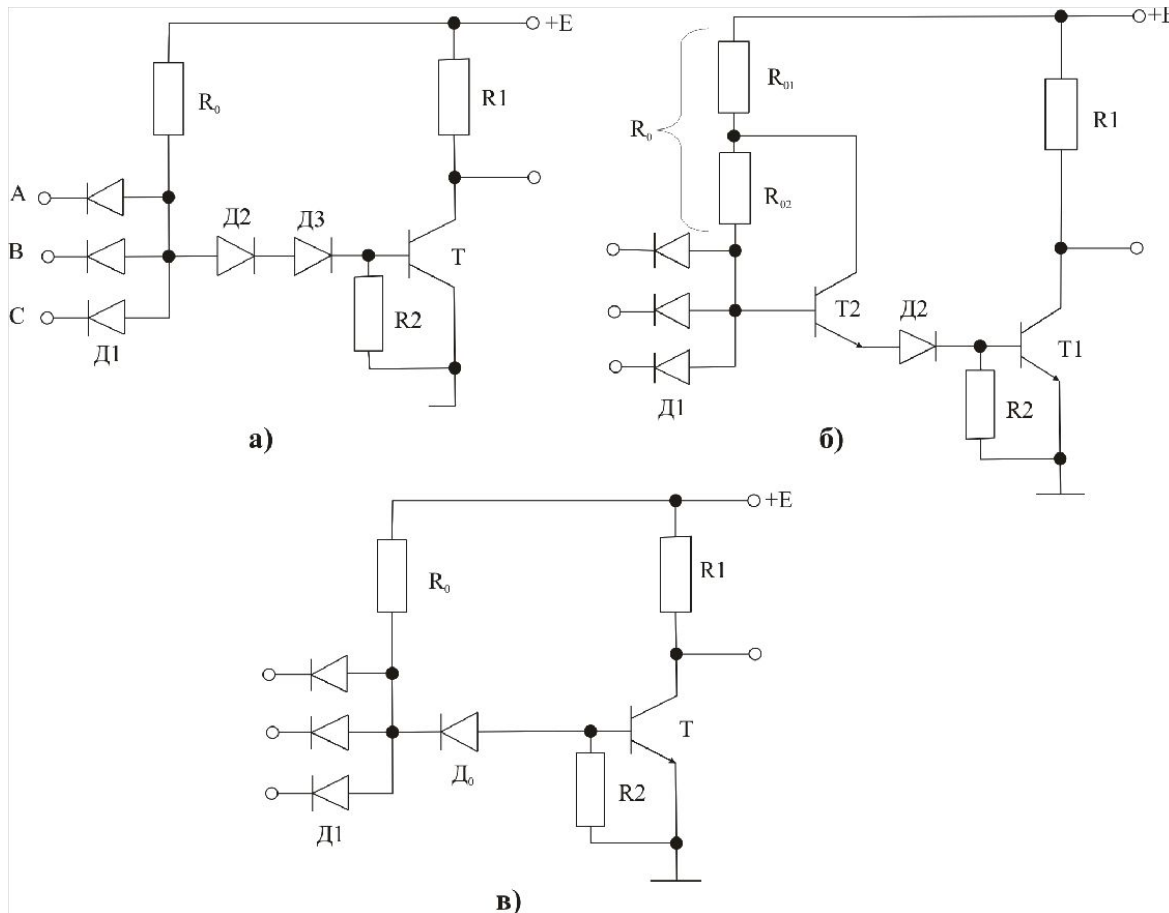
# Элемент И-ИЛИ-НЕ ТТЛ

$$W = A_1 B_1 C_1 + A_2 B_2 C_2 + \dots + A_k B_k C_k$$



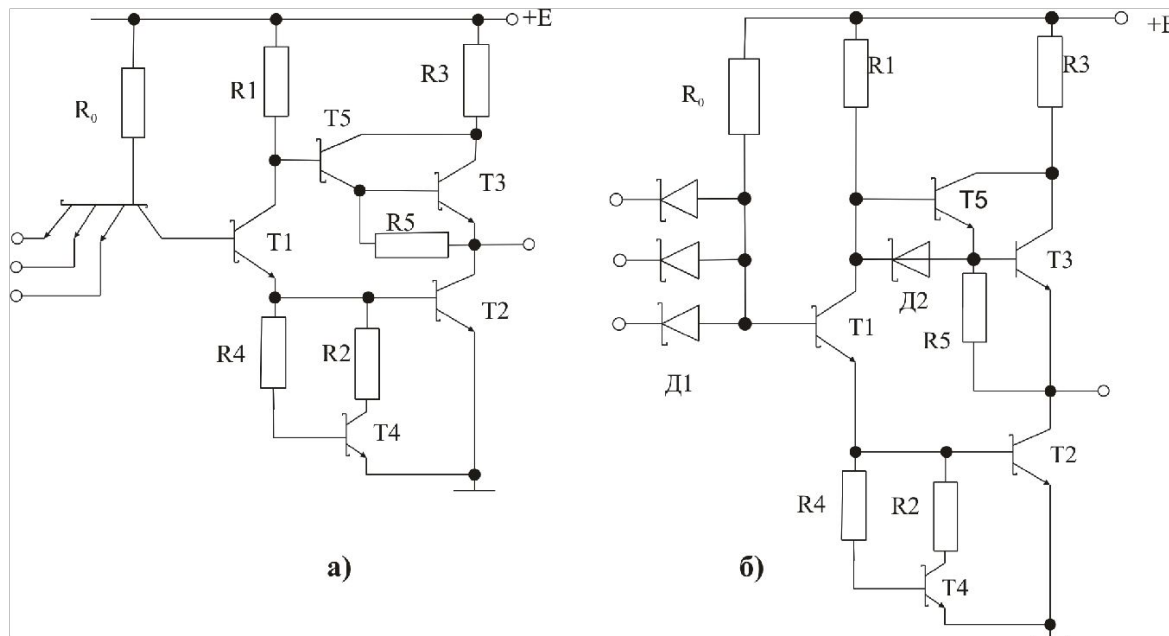
# Элементы И-НЕ ДТЛ.

- а) типовой, б) с повышенным коэффициентом разветвления в) повышенной помехоустойчивостью



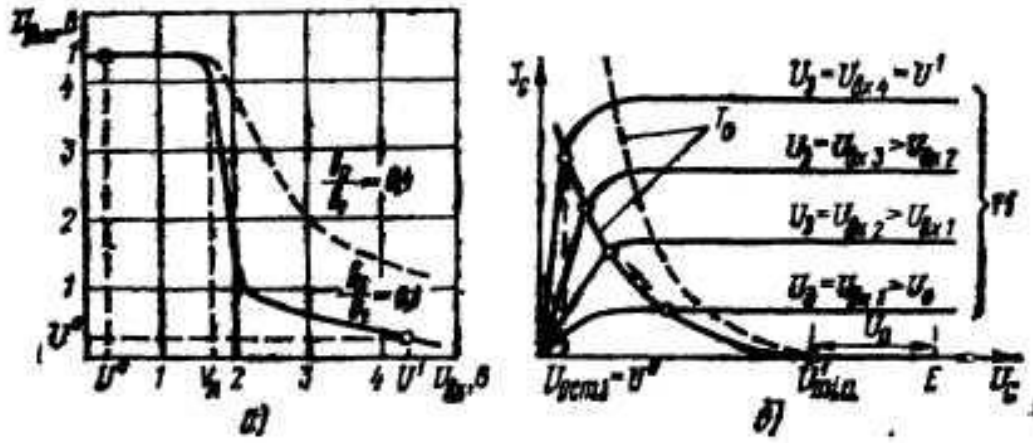
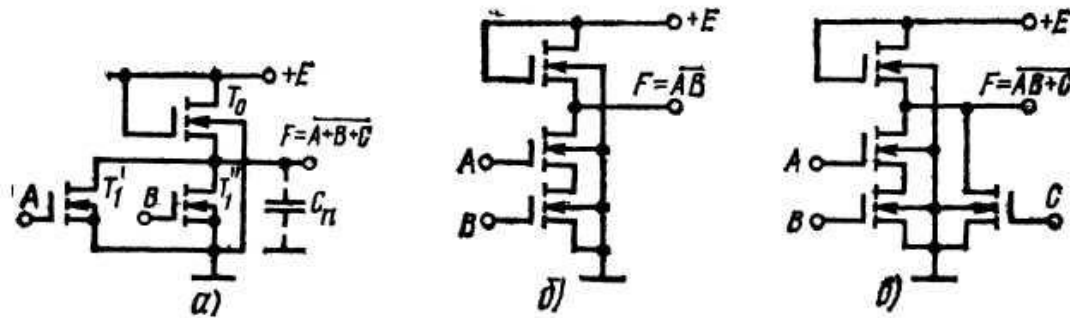
# Элементы И-НЕ ТШЛ (а) и ДТШЛ (б).

- Повышение быстродействия в элементах ТШЛ обусловлено двумя причинами: во-первых, не учитывается время рассасывания избыточного заряда в транзисторах Шотки, т. е.  $t_p \approx 0$ ; во-вторых, уменьшаются времена спада и нарастания  $t_c$ ,  $t_n$  так как транзисторы Шотки, не легированные золотом, имеют в 2-3 раза большие значения коэффициентов, чем обычные транзисторы в микросхемах ТТЛ, легируемых золотом. В микросхемах ТШЛ достигаются значения  $U = 2-3$  нс при  $P = 10-20$  мВт, а работа переключения составляет  $A_p = 20-50$  пДж.



# Элементы МДПТЛ

- а) ИЛИ-НЕ; б) И-НЕ; в) И-ИЛИ-НЕ.



## **Лекция 2**

**Непрерывные (аналоговые)  
или цифровые СУ ?  
Какое решение выбрать?**

**Произведения всех действительно даровитых голов отличаются от остальных характером решительности и определённости, и вытекающими из них отчётливостью и ясностью. Ибо такие головы всегда определённо и ясно сознают, что они хотят выразить, - всё равно, будет ли это проза, стихи или звуки. Этой решительности и ясности недостаёт прочим, и они тотчас же распознаются по этому недостатку. Характеристический признак первостепенных умов есть непосредственность всех их суждений и приговоров. Всё, что они производят, есть результат их самособственного мышления, который повсюду обнаруживается как таковой уже в самом изложении.**

**Артур Шопенгауэр**



Проведем сравнение непрерывных и дискретных систем управления по трем группам критериев:

Проведем сравнение непрерывных и дискретных систем управления по трем группам критериев:

**Сравнение с точки зрения самого процесса управления**

Устойчивость:

при :

$$x(t) = x_{\text{собств}}(t) + x_{\text{вын}}(t),$$

при

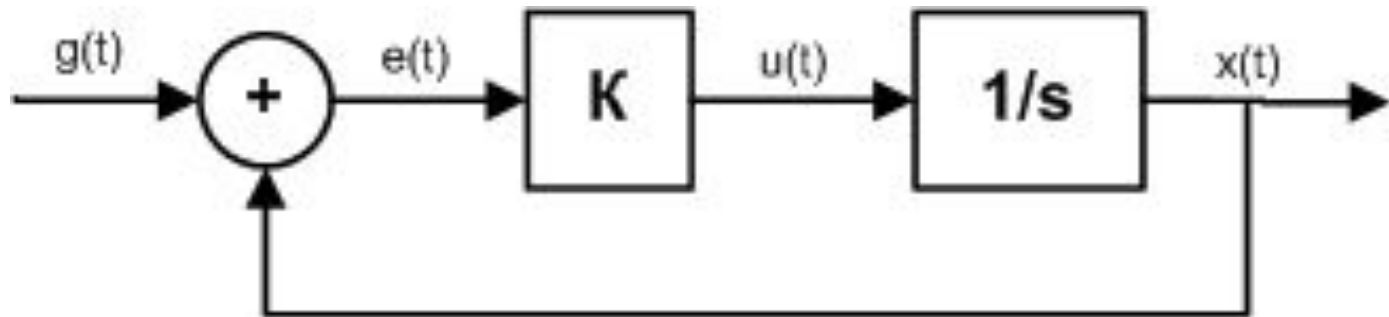
$$t \rightarrow \infty \quad x_{\text{собств}}(t) \rightarrow 0$$

Точность:  $\varepsilon_{\text{устан}} = \lim_{t \rightarrow \infty} \varepsilon(t)$

Качество процесса управления, т.е. параметры переходного процесса: перерегулирование (должно быть по возможности меньше) и время переходного процесса (также должно быть по возможности меньше).

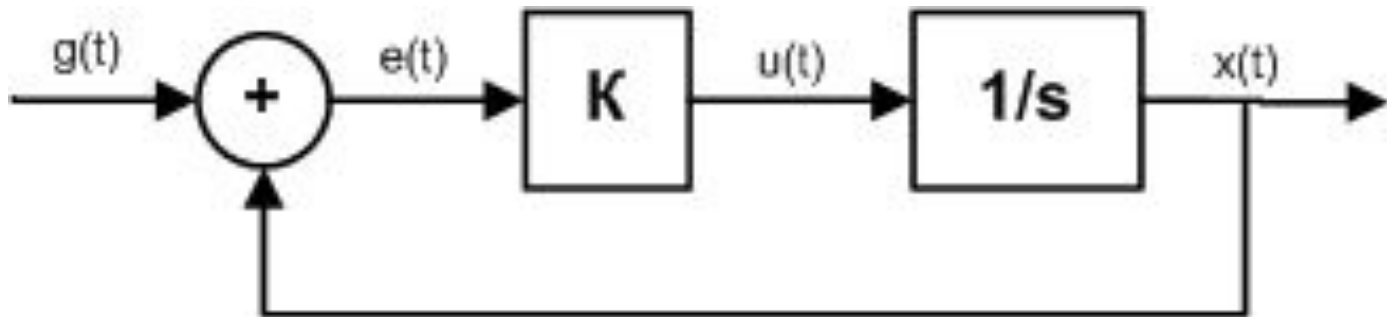
# Сравнение по общетехническим характеристикам

Масса и габариты;  
Энергопотребление;  
Надежность.



# Сравнение по технико-экономическим параметрам

Стоимость разработки и изготовления;  
Стоимость модернизации (изменение алгоритма управления).



# Сравнение с точки зрения процесса управления

## Устойчивость

$$W(s) = \frac{k}{s} \quad \Phi(s) = \frac{1}{Ts + 1}, \quad T = \frac{1}{k}$$

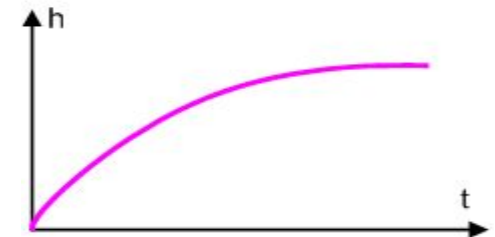
Точность : Система имеет астатизм 1-го порядка, следовательно, установившаяся ошибка равна нулю, если  $g(t) = \text{const}$ . Если ,

$$g(t) = g_0 t$$

ошибка обратно пропорциональна коэффициенту усиления  $k$

## Переходной процесс

$$h(t) = 1 - e^{-\frac{t}{T}}$$



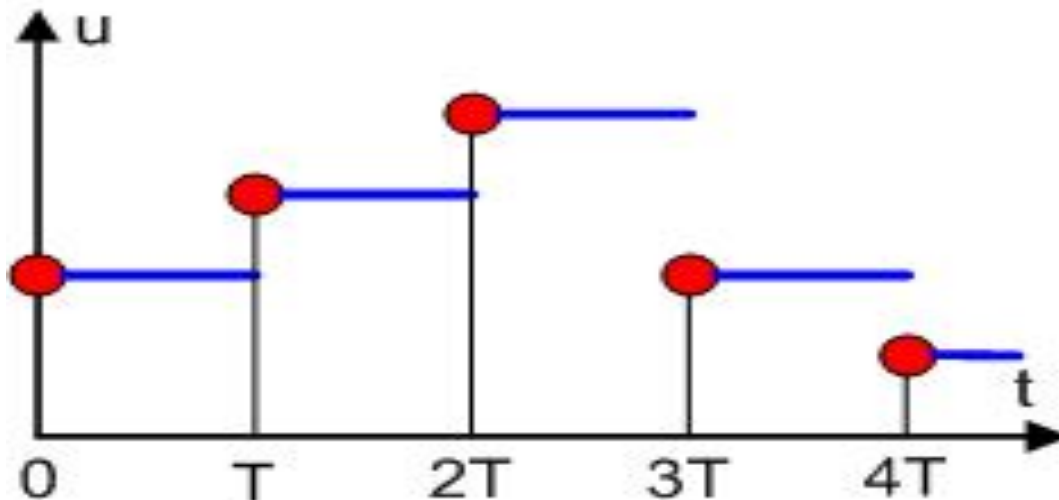
- Перерегулирования нет, уменьшая постоянную времени  $T$  мы можем добиться уменьшения времени переходного процесса.
- Теперь рассмотрим вариант дискретной организации блока управления.

Так как значения управляющего сигнала на выходе блока управления определены лишь в дискретные моменты времени, необходимо использовать экстраполяцию для определения значения на всем интервале

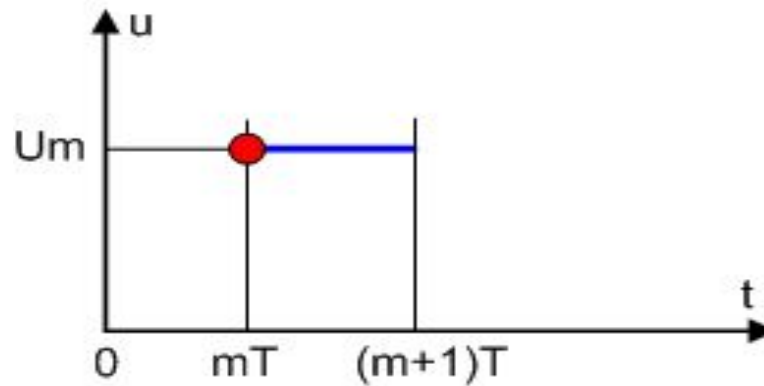
# Дискретная организация блока управления

- Будем считать, что в течение периода

$$u(t) = \text{const}$$



Рассмотрим произвольно взятый интервал времени



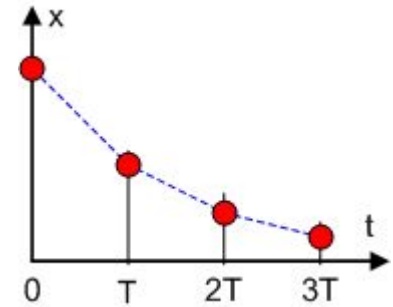
Можем записать следующие соотношения:  $u(t) = u_m = \text{const}$ ,  $u(t) = k \cdot \varepsilon(t)$ ,

$$\varepsilon(t) = g(t) - x(t). \quad x(mT) = -k \cdot x(mT) \cdot mT$$

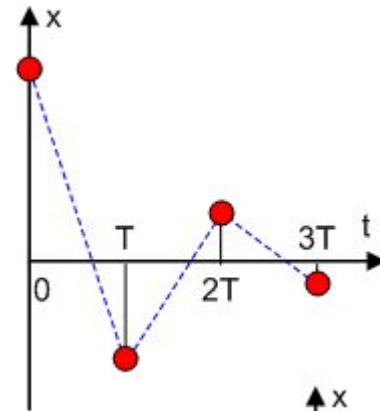
Для произвольного момента времени:  $x[nT] = x_0 (1 - kT)^n$ ,

где  $x_0$  определяется начальными условиями.

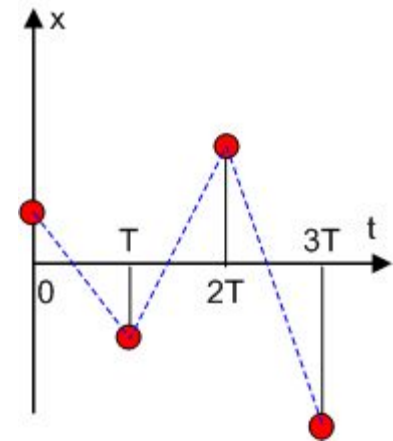
- Для  $kT < 1$   $kT = 0.5$



$$1 < kT < 2 \quad kT = 1.5$$



$$kT > 2 \quad kT = 2.5$$





- Таким образом, мы видим, что в цифровой системе устойчивость, точность и качество управления зависят от параметров системы, и прежде всего, от значения *(периода дискретизации, который определяет  $kT$  временем работы алгоритма управления)*. В зависимости от значения величины система может стать неустойчивой, чем больше значение этой величины, тем хуже вид переходного процесса. Существуют ограничения на значение  $kT$ , то есть существует предельное значение  $kT$ , при превышении которого система теряет устойчивость. Следовательно существует ограничение на значение коэффициента усиления  $K$ . Если же предположить, что фиксирован коэффициент усиления  $K$ , показатели системы ухудшаются при увеличении периода дискретизации  $kT$ , и мы можем сказать, что при увеличении  $kT$  выше некоего предельного значения, система также теряет устойчивость.

- На основании этого можно сделать вывод, что при использовании линейных алгоритмов управления, цифровая система всегда хуже непрерывной системы с точки зрения процесса управления. Одна из причин такого положения заключается в том, что в дискретной системе сигнал обратной связи вводится в дискретные моменты времени, следовательно в течение интервала времени система существует без обратной связи.

# Параметры сравнения

## Непрерывная система

## Дискретная система

**Масса и габариты**

Приблизительно одинаковы

**Энергопотребление**

Хуже

Лучше

**Надежность**

Хуже

Лучше

**Сравнение по технико-экономическим характеристикам**

**Стоимость разработки**

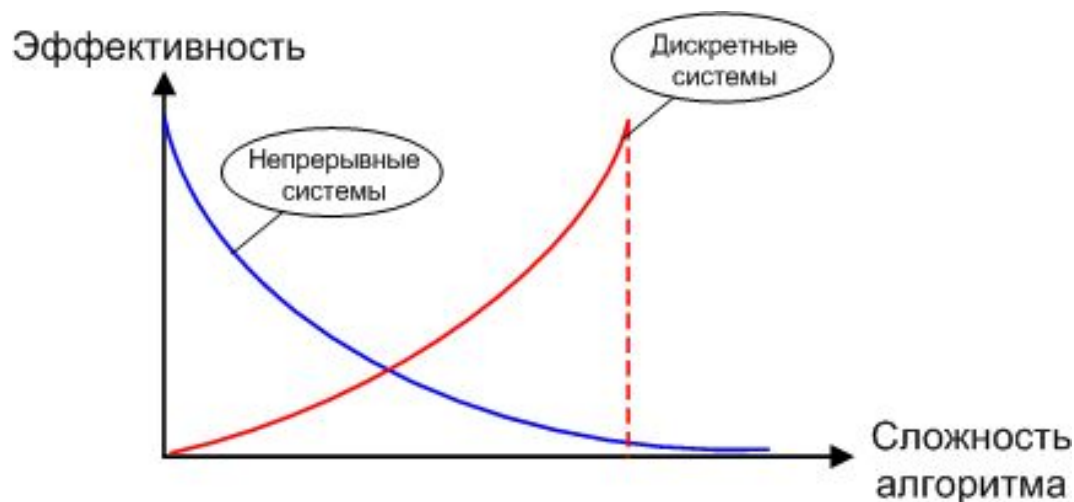
Приблизительно одинакова

**Стоимость модернизации**

Выше

Ниже

# Зависимость эффективности непрерывной и дискретной реализации блока управления от сложности реализуемого алгоритма



Из графика видно, что по мере усложнения алгоритма, эффективность непрерывной системы уменьшается, так как возрастает число включенных в нее электронных элементов, а следовательно, усложняется конструкция, увеличиваются масса, габариты, стоимость, уменьшается точность и общая надежность.

# Выводы

- Для дискретной системы **усложнение** алгоритма **приводит** лишь к **изменению программы**, что не влияет ни на массу и габариты, ни на стоимость технической реализации, так как не меняется конструкция самого блока управления.
- Правда, при дальнейшем усложнении алгоритма наступает критический момент, когда эффективность дискретной системы резко падает. Это связано с чрезмерным усложнением программы, сложностью ее отладки и уменьшением общей надежности системы.

- **Вывод:** Дискретная система управления имеет два основных преимущества по сравнению с непрерывной системой:
- Простота модернизации (изменения алгоритма);
- Большая эффективность при использовании сложных (нелинейных, адаптивных) алгоритмов управления.



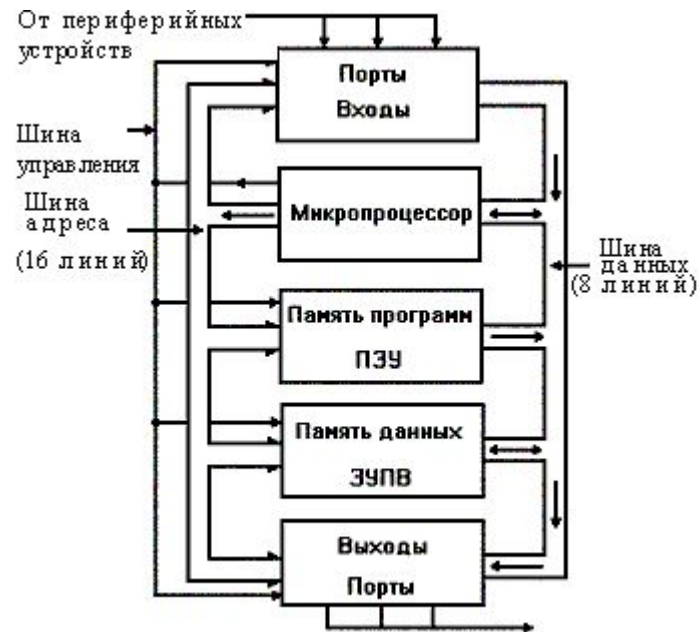
# Лекция 3

Микропроцессорные системы  
управления  
Структура. Требования.  
Типы МК.



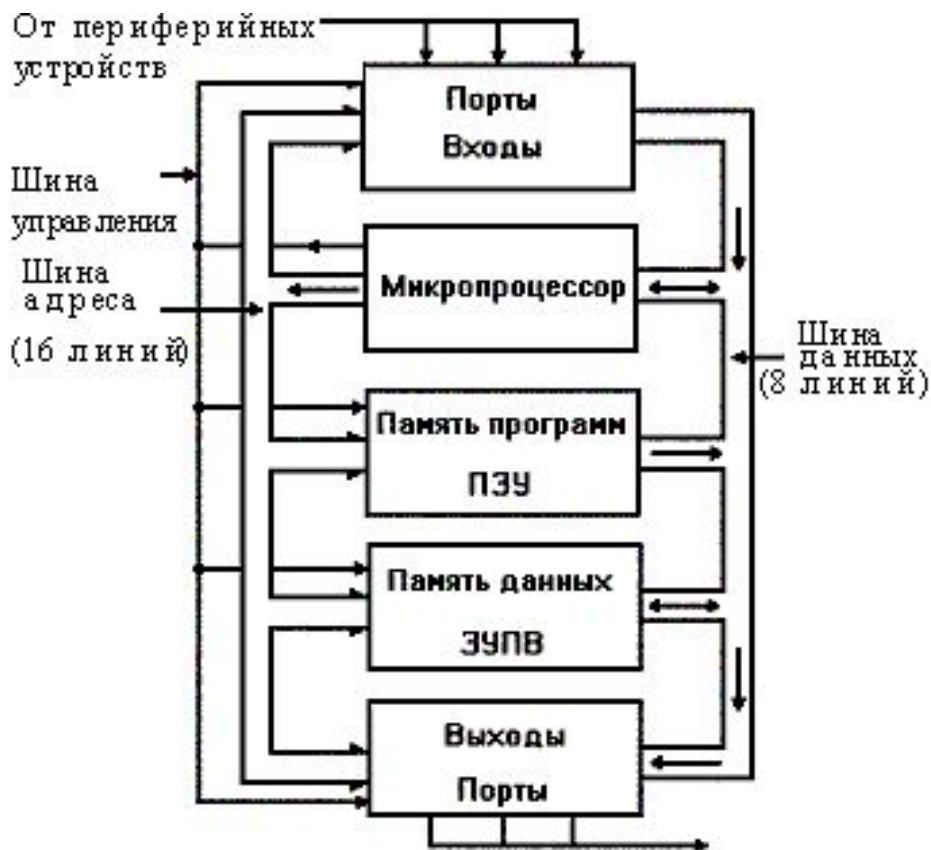
# Структура типового микропроцессора

Архитектура типичной небольшой вычислительной системы на основе микроЭВМ показана на рис. Такая микроЭВМ содержит все 5 основных блоков цифровой машины: устройство ввода информации, управляющее устройство (УУ), арифметико-логическое устройство (АЛУ) (входящие в состав микропроцессора), запоминающие устройства (ЗУ) и устройство вывода информации.



## МП

Микропроцессор координирует работу всех устройств цифровой системы с помощью шины управления (ШУ). Помимо ШУ имеется 16-разрядная адресная шина (ША), которая служит для выбора определенной ячейки памяти, порта ввода или порта вывода. По 8-разрядной информационной шине или шине данных (ШД) осуществляется двусторонняя пересылка данных к микропроцессору и от микропроцессора. Важно отметить, что МП может посылать информацию в память микро ЭВМ или к одному из портов вывода, а также получать информацию из памяти или от одного из портов ввода.



## Управляющие блоки

Основные требования, которые потребители предъявляют к управляющим блокам приборов (микроконтроллерам) можно сформулировать следующим образом:

В отличие от универсальных компьютеров к управляющим контроллерам, как правило, не предъявляются высокие требования к производительности и программной совместимости.

- низкая стоимость,
- высокая надежность,
- высокая степень миниатюризации,
- малое энергопотребление,
- работоспособность в жестких условиях эксплуатации;
- достаточная производительность для выполнения всех требуемых функций.
- В отличие от универсальных компьютеров к управляющим контроллерам, как правило, не предъявляются высокие требования к производительности и программной совместимости.
- Выполнение всех этих довольно противоречивых условий одновременно затруднительно, поэтому развитие и совершенствование техники пошло по пути специализации и в настоящее время количество различных моделей управляющих микроконтроллеров чрезвычайно велико.

# Классификация микропроцессорных систем

## Архитектура микропроцессора

Микропроцессор характеризуется:

1) тактовой частотой, определяющей максимальное время выполнения переключения элементов в ЭВМ;  
2) разрядностью, т.е. максимальным числом одновременно обрабатываемых двоичных разрядов.

Разрядность МП обозначается  $m/n/k/$  и включает:

$m$  - разрядность внутренних регистров, определяет принадлежность к тому или иному классу процессоров;  $n$  - разрядность шины данных, определяет скорость передачи информации;  
 $k$  - разрядность шины адреса, определяет размер адресного пространства. Например, МП i8088 характеризуется значениями  $m/n/k=16/8/20$ ;

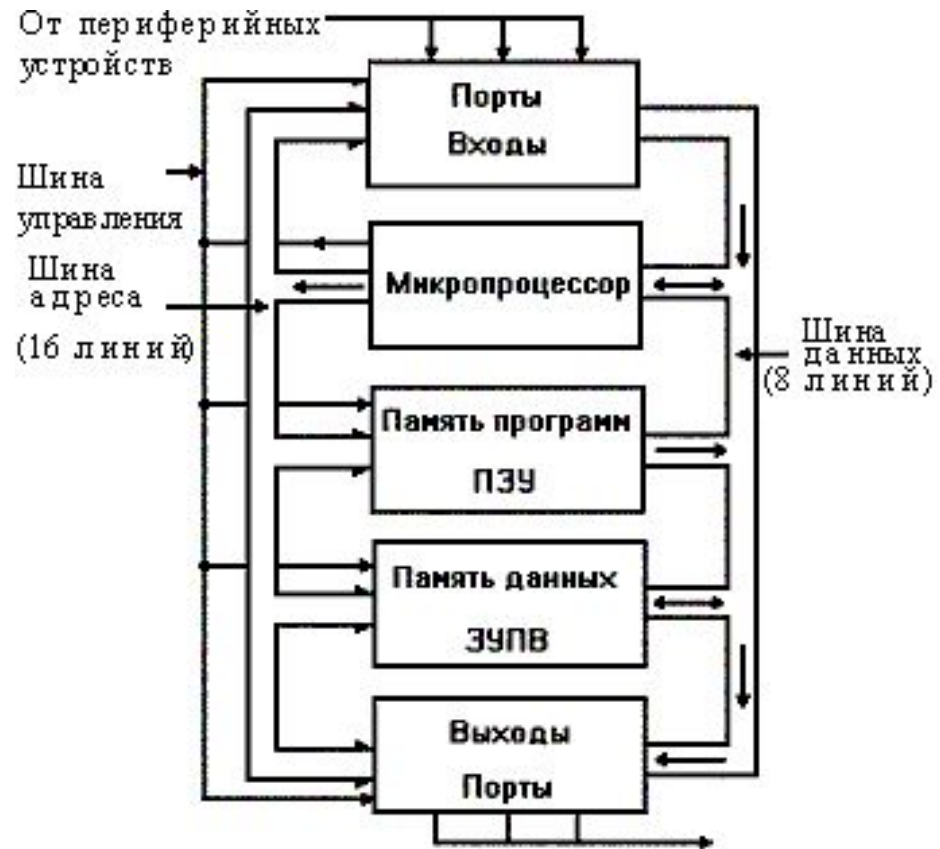
3) архитектурой. Понятие архитектуры микропроцессора включает в себя систему команд и способы адресации, возможность совмещения выполнения команд во времени, наличие дополнительных устройств в составе микропроцессора, принципы и режимы его работы. Выделяют понятия микроархитектуры и макроархитектуры.

- Микроархитектура микропроцессора - это аппаратная организация и логическая структура микропроцессора, регистры, управляющие схемы, арифметико-логические устройства, запоминающие устройства и связывающие их информационные магистрали.
- Макроархитектура - это система команд, типы обрабатываемых данных, режимы адресации и принципы работы микропроцессора.
- В общем случае под архитектурой ЭВМ понимается абстрактное представление машины в терминах основных функциональных модулей, языка ЭВМ, структуры данных.

•

## ПЗУ

Постоянное запоминающее устройство (ПЗУ) в микроЭВМ содержит некоторую программу (на практике программу инициализации ЭВМ). Программы могут быть загружены в запоминающее устройство с произвольной выборкой (ЗУПВ) и из внешнего запоминающего устройства (ВЗУ). Это программы пользователя.



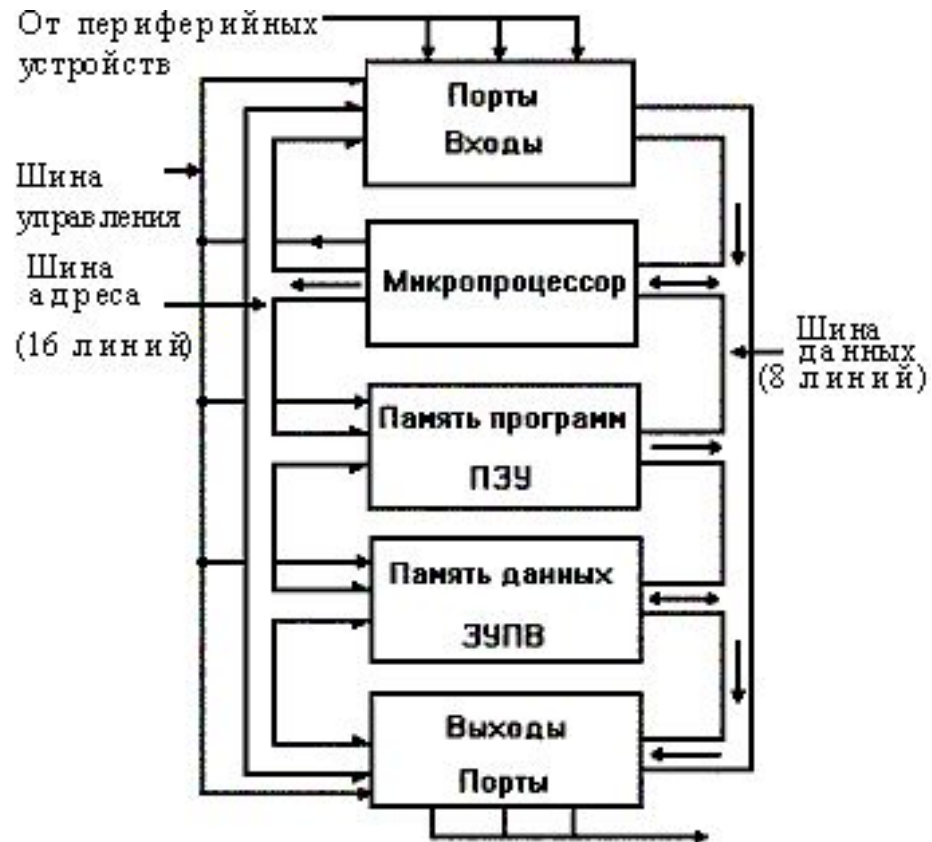
В настоящее время выпускается целый ряд типов МК. Все эти приборы можно условно разделить на три основных класса:

8-разрядные МК для встраиваемых приложений;

16- и 32-разрядные МК;

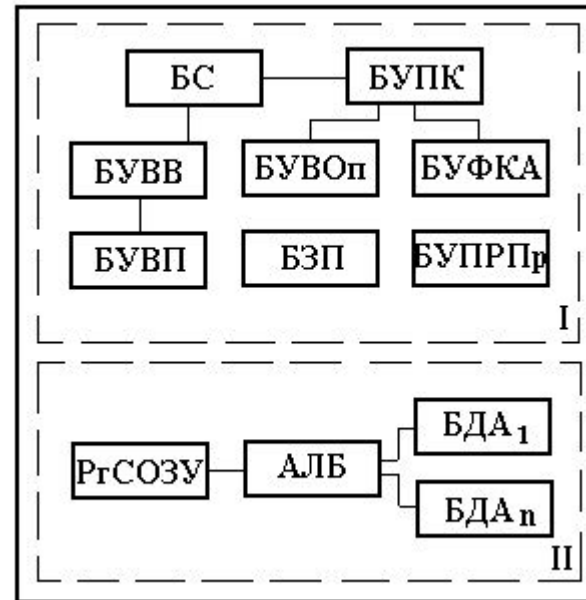
цифровые сигнальные процессоры (DSP).

Наиболее распространенным представителем семейства МК являются 8-разрядные приборы, широко используемые в промышленности, бытовой и компьютерной технике. Они прошли в своем развитии путь от простейших приборов с относительно слаборазвитой периферией до современных многофункциональных контроллеров, обеспечивающих реализацию сложных алгоритмов управления в реальном масштабе времени. Причиной жизнеспособности 8-разрядных МК является использование их для управления реальными объектами, где применяются, в основном, алгоритмы с преобладанием логических операций, скорость обработки которых практически не зависит от разрядности процессора.



# Логическая структура микропроцессора

Логическая структура микропроцессора, т. е. конфигурация составляющих микропроцессор логических схем и связей между ними, определяется функциональным назначением. Именно структура задает состав логических блоков микропроцессора и то, как эти блоки должны быть связаны между собой, чтобы полностью отвечать архитектурным требованиям. Срабатывание электронных блоков микропроцессора в определенной последовательности приводит к выполнению заданных архитектурой микропроцессора функций, т. е. к реализации вычислительных алгоритмов.



## Типы МК

Росту популярности 8-разрядных МК способствует постоянное расширение номенклатуры изделий, выпускаемых такими известными фирмами, как Motorola, Microchip, Intel, Zilog, Atmel и многими другими.

Современные 8-разрядные МК обладают, как правило, рядом отличительных признаков.

Перечислим основные из них:

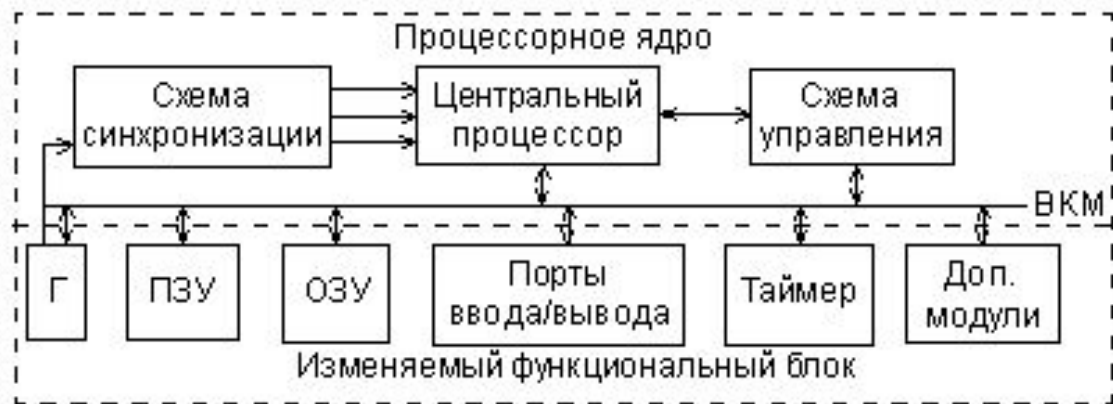
- модульная организация, при которой на базе одного процессорного ядра (центрального процессора) проектируется ряд (линейка) МК, различающихся объемом и типом памяти программ, объемом памяти данных, набором периферийных модулей, частотой синхронизации;
- использование закрытой архитектуры МК, которая характеризуется отсутствием линий магистралей адреса и данных на выводах корпуса МК. Таким образом, МК представляет собой законченную систему обработки данных, наращивание возможностей которой с использованием параллельных магистралей адреса и данных не предполагается;
- использование типовых функциональных периферийных модулей (таймеры, процессоры событий, контроллеры последовательных интерфейсов, аналого-цифровые преобразователи и др.), имеющих незначительные отличия в алгоритмах работы в МК различных производителей;
- расширение числа режимов работы периферийных модулей, которые задаются в процессе инициализации регистров специальных функций МК.



## Типы МК

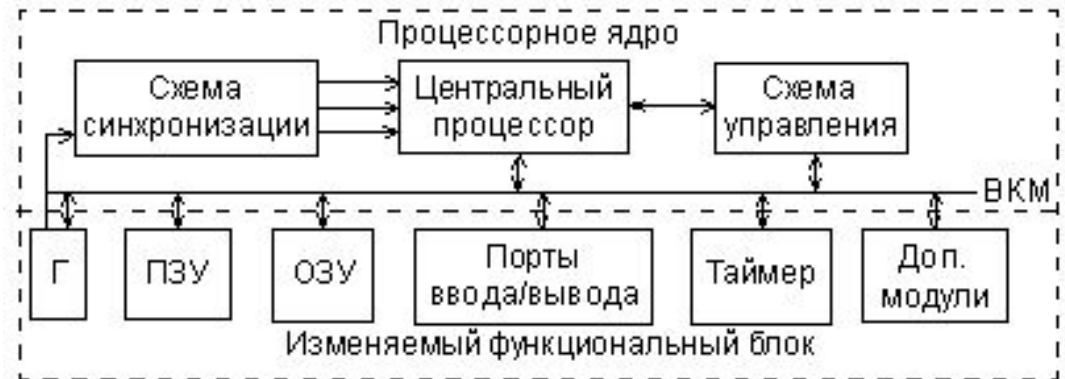
При модульном принципе построения все МК одного семейства содержат процессорное ядро, одинаковое для всех МК данного семейства, и изменяемый функциональный блок, который отличает МК разных моделей. Структура модульного МК приведена на рис.2.4.

- Процессорное ядро включает в себя:
- центральный процессор;
- внутреннюю контроллерную магистраль (ВКМ) в составе шин адреса, данных и управления;
- схему синхронизации МК;
- схему управления режимами работы МК, включая поддержку режимов пониженного энергопотребления, начального запуска (сброса) и т.д.



Изменяемый функциональный блок включает в себя модули памяти различного типа и объема, порты ввода/вывода, модули тактовых генераторов (Г), таймеры. В относительно простых МК модуль обработки прерываний входит в состав процессорного ядра. В более сложных МК он представляет собой отдельный модуль с развитыми возможностями. В состав изменяемого функционального блока могут входить и такие дополнительные модули как компараторы напряжения, аналого-цифровые преобразователи (АЦП) и другие. Каждый модуль проектируется для работы в составе МК с учетом протокола ВКМ. Данный подход позволяет создавать разнообразные по структуре МК в пределах одного семейства. Структура процессорного ядра МК

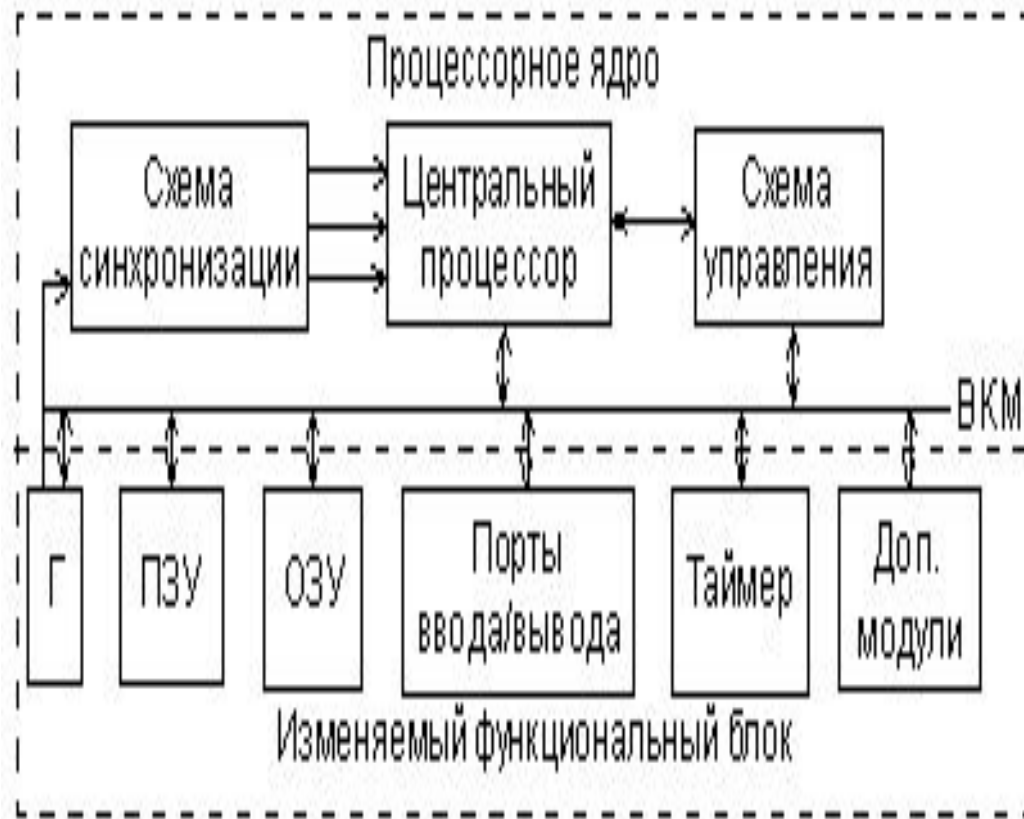
- Основными характеристиками, определяющими производительность процессорного ядра МК, являются:
- набор регистров для хранения промежуточных данных;
- система команд процессора;
- способы адресации операндов в пространстве памяти;
- организация процессов выборки и исполнения команды.



В МПС микропроцессор

выполняет следующие функции:

- выборку команд программы из основной памяти;
- дешифрацию команд;
- выполнение арифметических, логических и других операций, закодированных в командах;
- управление пересылкой информации между регистрами и основной памятью, между устройствами ввода/вывода;
- обработку сигналов от устройств ввода/вывода, в том числе реализацию прерываний с этих устройств;
- управление и координацию работы основных узлов МП.



## Архитектура

С точки зрения системы команд и способов адресации операндов процессорное ядро современных 8-разрядных МК реализует один из двух принципов построения процессоров:

процессоры с CISC-архитектурой, реализующие так называемую полную систему команд (Complicated Instruction Set Computer);

- CISC-процессоры выполняют большой набор команд с развитыми возможностями адресации, давая разработчику возможность выбрать наиболее подходящую команду для выполнения необходимой операции. В применении к 8-разрядным МК процессор с CISC-архитектурой может иметь однобайтовый, двухбайтовый и трехбайтовый (редко четырехбайтовый) формат команд. Время выполнения команды может составлять от 1 до 12 циклов. К МК с CISC-архитектурой относятся МК фирмы Intel с ядром MCS-51, которые поддерживаются в настоящее время целым рядом производителей, МК семейств HC05, HC08 и HC11 фирмы Motorola и ряд других.

## Архитектура

Процессоры с RISC-архитектурой, реализующие сокращенную систему команд (Reduced Instruction Set Computer).

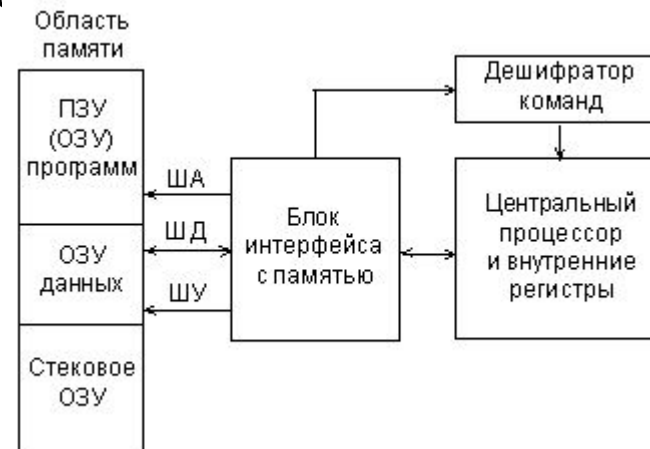
- В процессорах с RISC-архитектурой набор исполняемых команд сокращен до минимума. Для реализации более сложных операций приходится комбинировать команды. При этом все команды имеют формат фиксированной длины (например, 12, 14 или 16 бит), выборка команды из памяти и ее исполнение осуществляется за один цикл (такт) синхронизации. Система команд RISC-процессора предполагает возможность равноправного использования всех регистров процессора. Это обеспечивает дополнительную гибкость при выполнении ряда операций. К МК с RISC-процессором относятся МК AVR фирмы Atmel, МК PIC16 и PIC17 фирмы Microchip и другие.
- **На первый взгляд, МК с RISC-процессором должны иметь более высокую производительность по сравнению с CISC МК при одной и той же тактовой частоте внутренней магистрали. Однако на практике вопрос о производительности более сложен и неоднозначен.**

## Архитектура

С точки зрения организации процессов выборки и исполнения команды в современных 8-разрядных МК применяется одна из двух уже упоминавшихся архитектур МПС: фон-неймановская (принстонская) или гарвардская.

Основной особенностью **фон-неймановской архитектуры** является использование общей памяти для хранения программ и данных, как показано на рис.

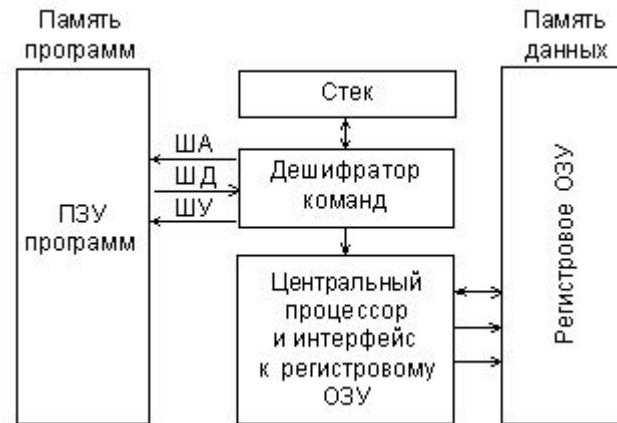
**Основное преимущество архитектуры Фон-Неймана** – упрощение устройства МПС, так как реализуется обращение только к одной общей памяти. Кроме того, использование единой области памяти позволяло оперативно перераспределять ресурсы между областями программ и данных, что существенно повышало гибкость МПС с точки зрения разработчика программного обеспечения. Размещение стека в общей памяти облегчало доступ к его содержимому. Неслучайно поэтому фон-неймановская архитектура стала основной архитектурой универсальных компьютеров, вклю



# Архитектура

Основной особенностью гарвардской архитектуры является использование отдельных адресных пространств для хранения команд и данных, как показано на рис.

- Гарвардская архитектура почти не использовалась до конца 70-х годов, пока производители МК не поняли, что она дает определенные преимущества разработчикам автономных систем управления.



## Архитектура

Гарвардская архитектура обеспечивает потенциально более высокую скорость выполнения программы по сравнению с фон-неймановской за счет возможности реализации параллельных операций. Выборка следующей команды может происходить одновременно с выполнением предыдущей, и нет необходимости останавливать процессор на время выборки команды. Этот метод реализации операций позволяет обеспечивать выполнение различных команд за одинаковое число тактов, что дает возможность более просто определить время выполнения циклов и критичных участков программы.

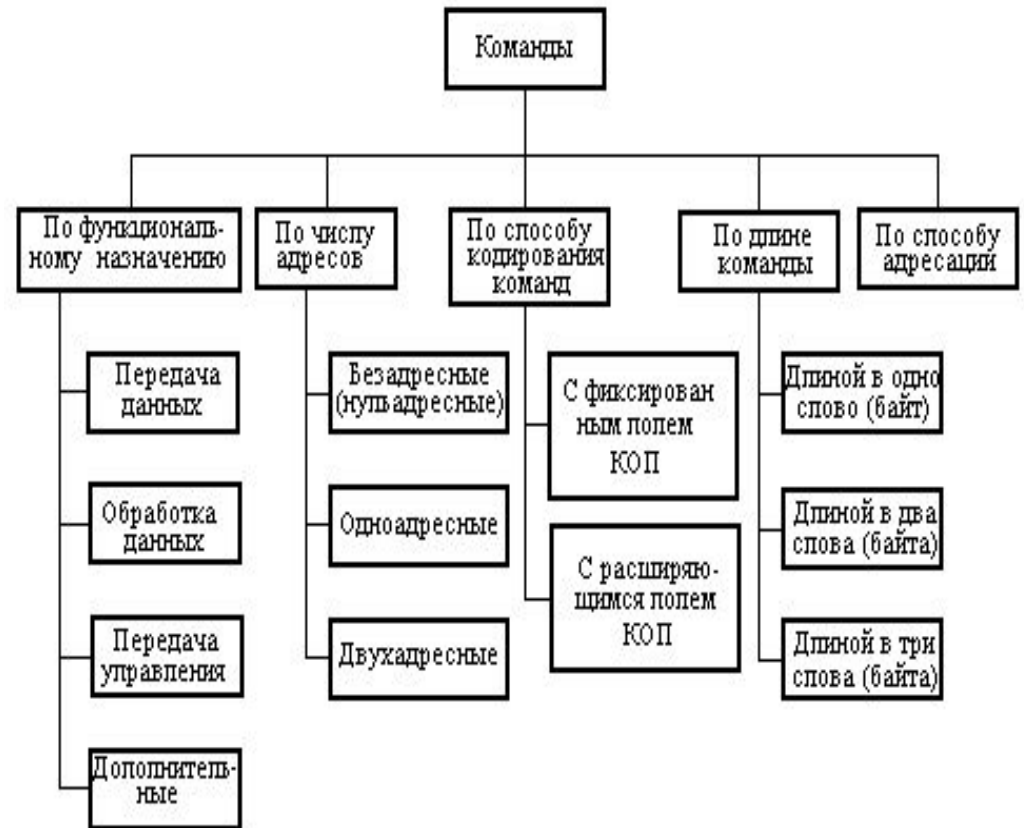
- По опыту использования МПС для управления различными объектами, для реализации большинства алгоритмов управления такие преимущества фон-неймановской архитектуры как гибкость и универсальность не имеют большого значения. Анализ реальных программ управления показал, что необходимый объем памяти данных МК, используемый для хранения промежуточных результатов, как правило, на порядок меньше требуемого объема памяти программ. В этих условиях использование единого адресного пространства приводило к увеличению формата команд за счет увеличения числа разрядов для адресации операндов. Применение отдельной небольшой по объему памяти данных способствовало сокращению длины команд и ускорению поиска информации в памяти данных.



## Архитектура

- Большинство производителей современных 8-разрядных МК используют гарвардскую архитектуру. Однако гарвардская архитектура является недостаточно гибкой для реализации некоторых программных процедур. Поэтому сравнение МК, выполненных по разным архитектурам, следует проводить применительно к конкретному приложению.
- В настоящее время наиболее яркими представителями микроконтроллеров SISC и RISC, имеющих соответственно фон-неймановскую и гарвардскую архитектуры являются микроконтроллеры i8051 и AVR – микроконтроллеры фирмы Atmel, которые по ряду характеристик превзошли очень известные PIC – микроконтроллеры. Поэтому рассмотрим организацию и устройство вышеперечисленных представителей.

# Классификация команд.



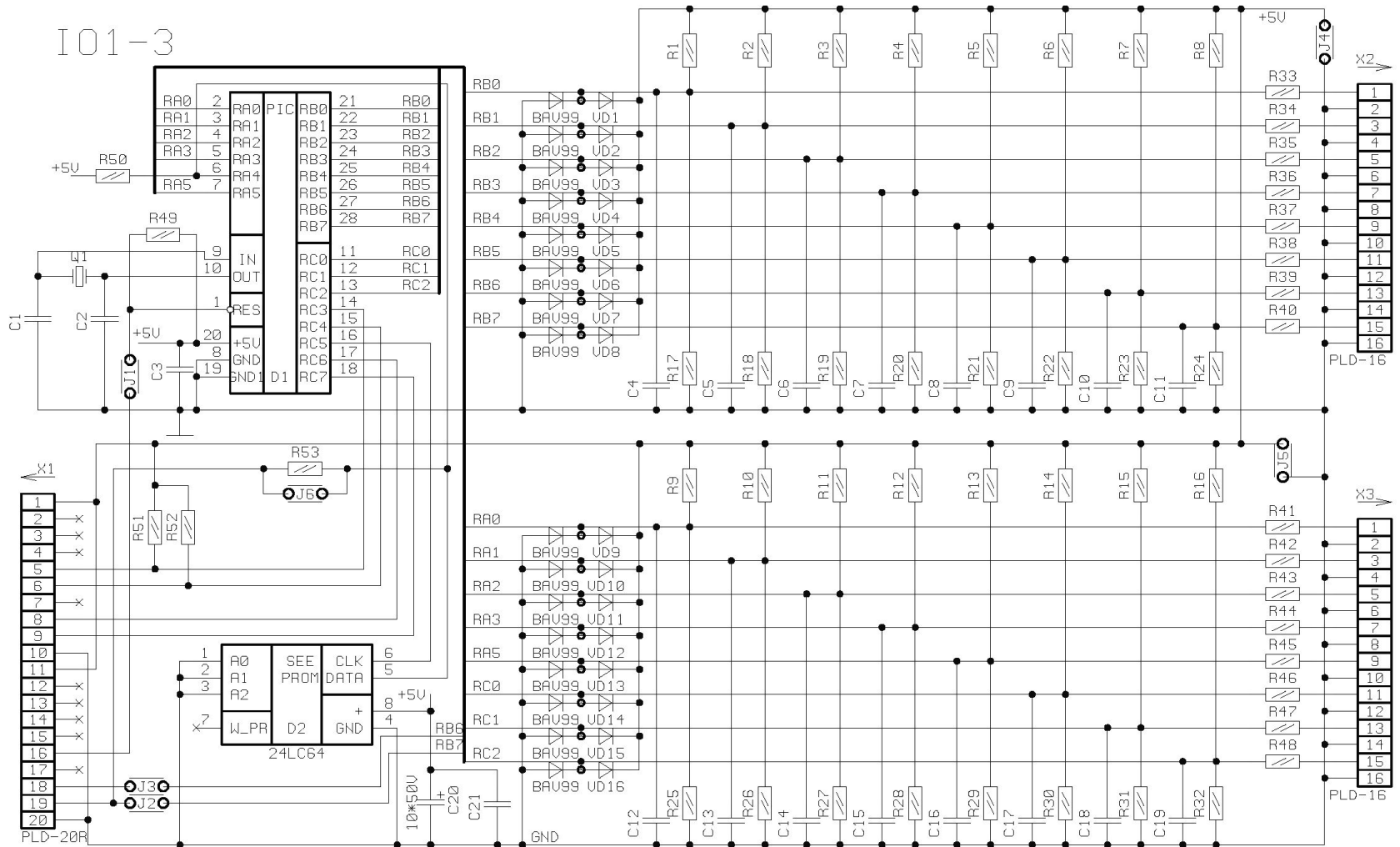
## Типы архитектур

Существует несколько подходов к классификации микропроцессоров по типу архитектуры. Так, выделяют МП с CISC (Complete Instruction Set Computer) архитектурой, характеризующейся полным набором команд, и RISC (Reduce Instruction Set Computer) архитектурой, которая определяет систему с сокращенным набором команд одинакового формата, выполняемых за один такт МП.

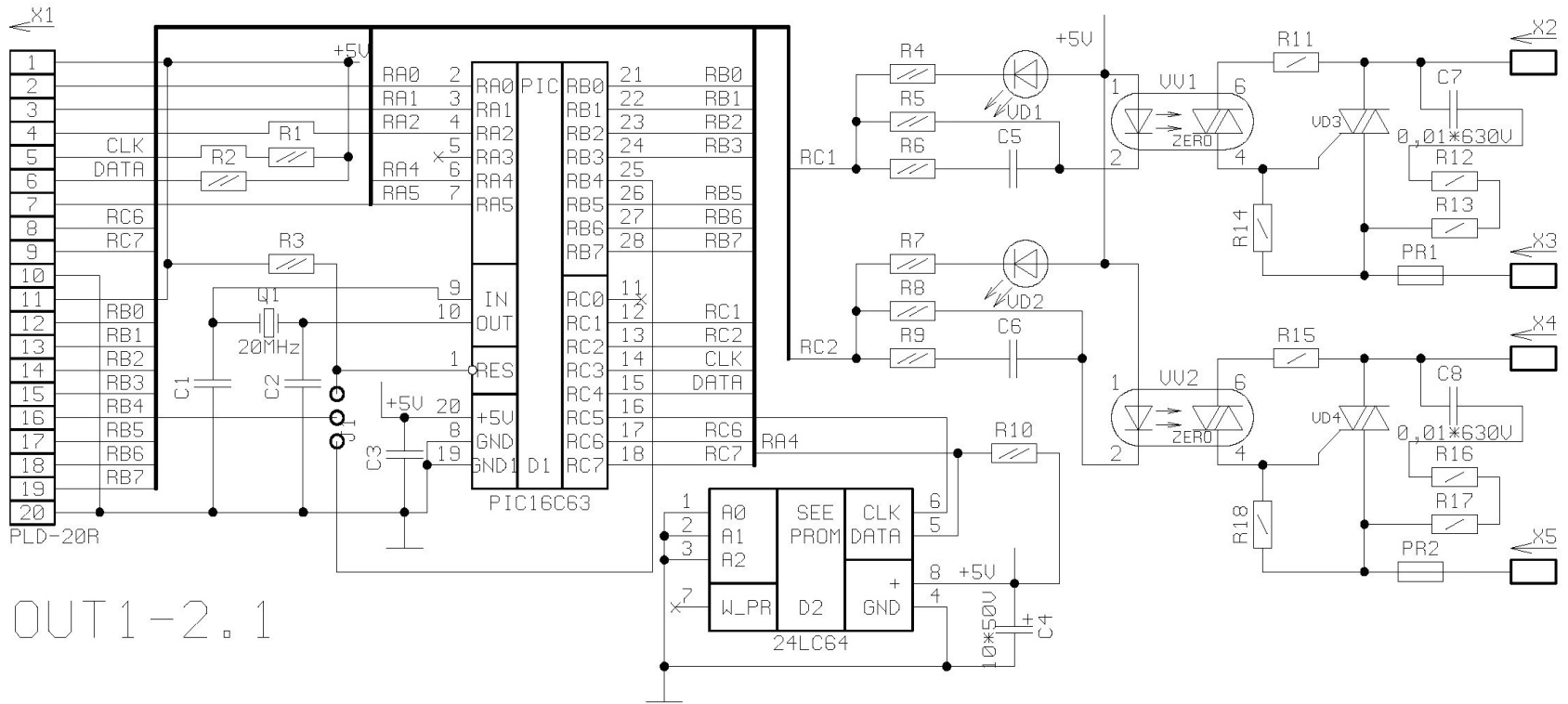
Определяя в качестве основной характеристики МП разрядность, выделяют следующие типы МП архитектуры:

- с фиксированной разрядностью и списком команд (однокристальные);
- с наращиваемой разрядностью (секционные) и микропрограммным управлением.

# Принципиальная электрическая схема модуля MSU42-3.

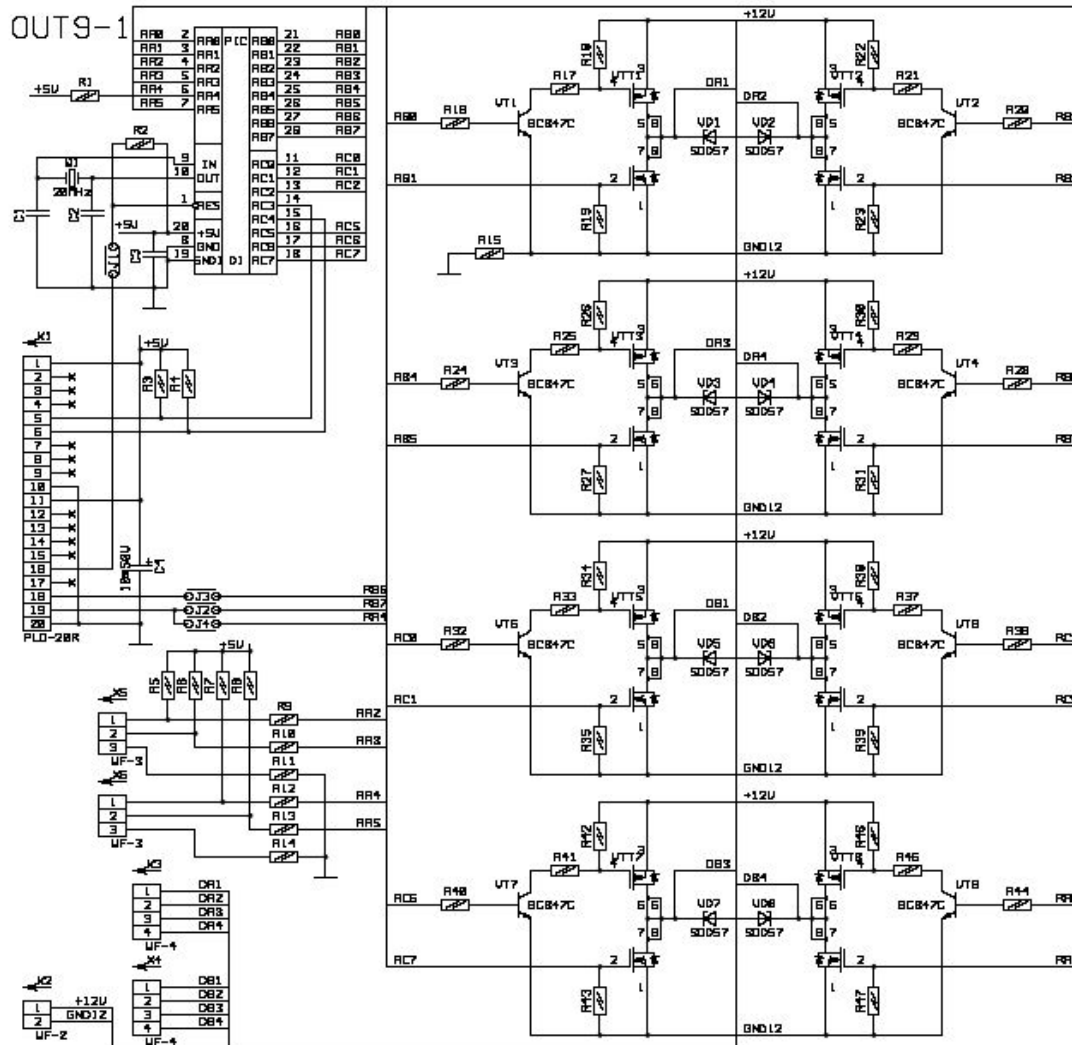


# Принципиальная электрическая схема модуля OUT1-X.



OUT1-2.1

# Принципиальная электрическая схема модуля OUT9-1.X.



# Лекция 4

Современные  
микроконтроллеры

## Знакомство с АРМ

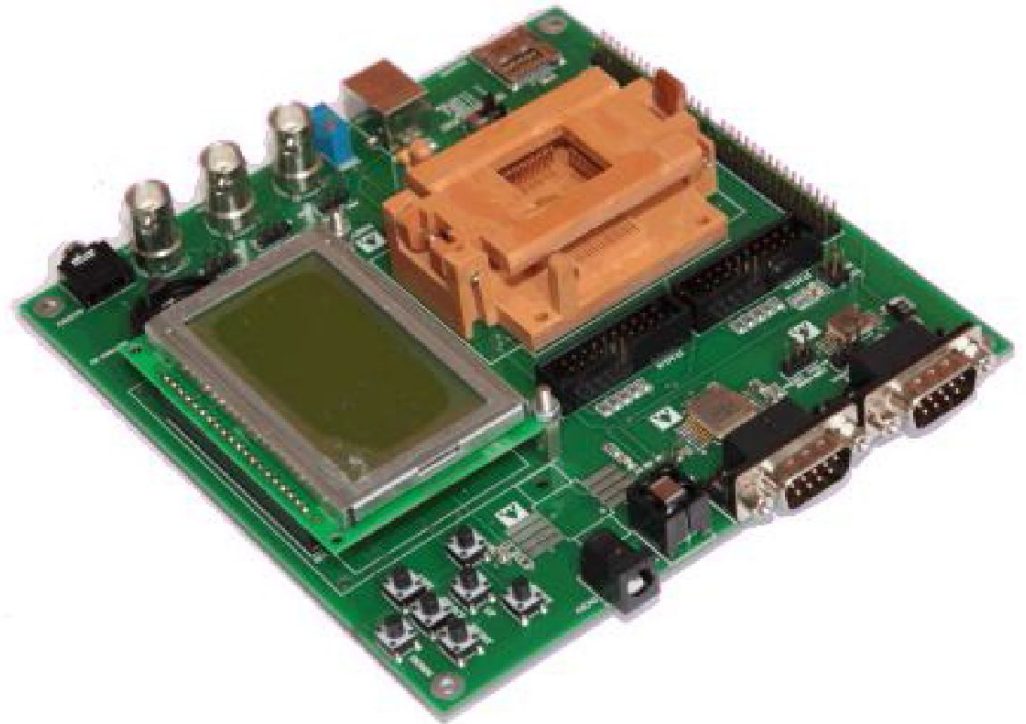
- разрядность шины данных, бит 32
- процессорное ядро ARM32 Cortex M3
- объем Flash памяти программ, кбайт 128
- объем статической оперативной памяти данных, кбайт 32
- тактовая частота процессорного ядра, МГц 80
- ток потребления в активном режиме при тактовой частоте ядра 80МГц, мА 120
- напряжение питания, В 2,2...3,6
- интерфейсы для программирования и отладки JTAG и SWD

- В состав МК входят следующие периферийные устройства:
- 43 линии ввода-вывода общего назначения, объединенные в 6 портов;
- 2 АЦП – аналого-цифровых преобразователя (12 бит, 8 каналов);
- 1 ЦАП – цифро-аналоговый преобразователь (12 бит);
- ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ. Лабораторная работа №1 2
- контроллер DMA – прямого доступа к памяти;
- 3 аппаратных таймера-счетчика общего назначения (каждый по 16 бит);
- часы реального времени;
- 1 интерфейс USB 2.0;
- 2 интерфейса USART;
- 2 интерфейса SPI;
- 1 интерфейс I2C;
- 2 интерфейс CAN.



# Отладочная плата 1986EvBrd\_64

Отладочная плата 1986EvBrd\_64, выпускаемая фирмой «Миландр», предназначена для ознакомления с возможностями микроконтроллера K1986BE92Q1 и отладки программного обеспечения для него. Внешний вид отладочной платы показан на рисунке 1.

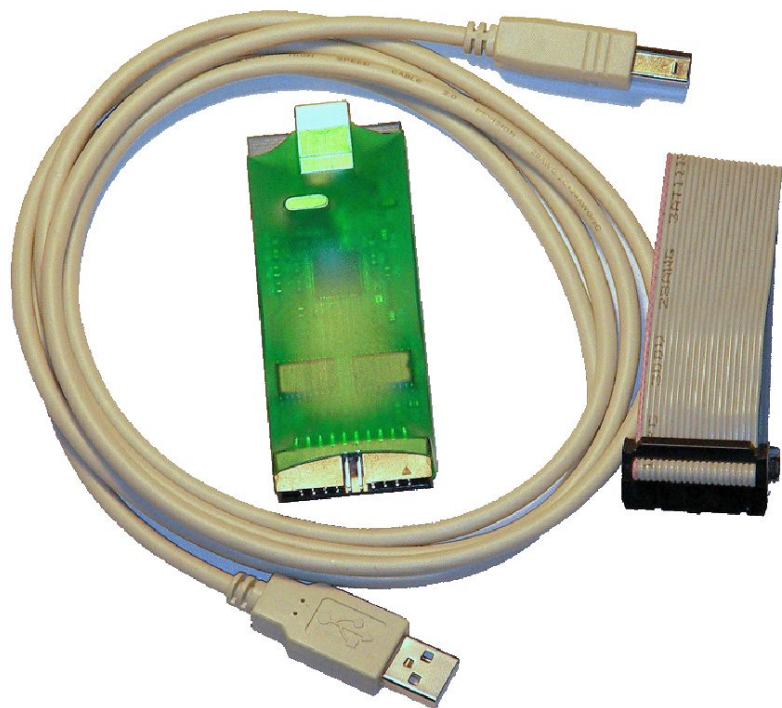


## Программатор

Для загрузки программ, написанных с помощью персонального компьютера, во Flash-память МК, а также для их отладки необходим программатор-отладчик. Нами будет

использован программатор-отладчик (далее просто программатор) MT-Link отечественной фирмы MT System (<http://www.mt-system.ru/>), показанный на рисунке . Он является

аналогом известного программатора J-Link фирмы IAR-Systems. Программатор подключается к компьютеру с помощью USB-кабеля и использует интерфейсы для внутрисхемной отладки SWD – Serial Wire Debug или JTAG. На плате предусмотрено два разъема для подключения программатора (JTAG-A и JTAG-B).  
MT-Link



## Отображение информации

Для отображения буквенно-цифровой и графической информации на плате предусмотрен монохромный жидкокристаллический индикатор размером 128x64 точки. При отладке приложений удобно пользоваться двумя светодиодами красного цвета, подключенными к выводам МК. Для ввода информации можно использовать пять механических кнопок общего назначения: «LEFT», «RIGHT», «UP», «DOWN» и «SELECT». Для сброса и перезапуска МК предназначена кнопка «RESET». На плате также предусмотрен целый ряд иных компонентов, которые будут постепенно изучаться в последующих лабораторных работах.

- Среда программирования Keil  $\mu$ Vision
- Интегрированная среда программирования Keil  $\mu$ Vision MDK-ARM предназначена для написания и отладки программ для микроконтроллеров семейства ARM32 с помощью языков Си, С++ и ассемблера. В рамках нашего курса будем писать программы лишь на языке Си.
- В состав среды входят все необходимые для этого средства: специализированный текстовый редактор с семантической подсветкой кода, компилятор, ассемблер, компоновщик, отладчик и т.д.
- Среда программирования поддерживает практически все выпускаемые в мире МК с архитектурой ARM32.
- Keil  $\mu$ Vision посредством драйверов может работать с
- различными внутрисхемными программаторами-отладчиками, в том числе и с MT-LINK.
- Внешним видом среда Keil  $\mu$ Vision напоминает среду Microsoft Visual Studio, поэтому программистам, знакомым с Visual Studio, нетрудно будет освоиться и со средой Keil. К сожалению, русифицированной версии Keil  $\mu$ Vision пока не появилось. Весь интерфейс
- организован на английском языке.

# Программная модель внешнего устройства



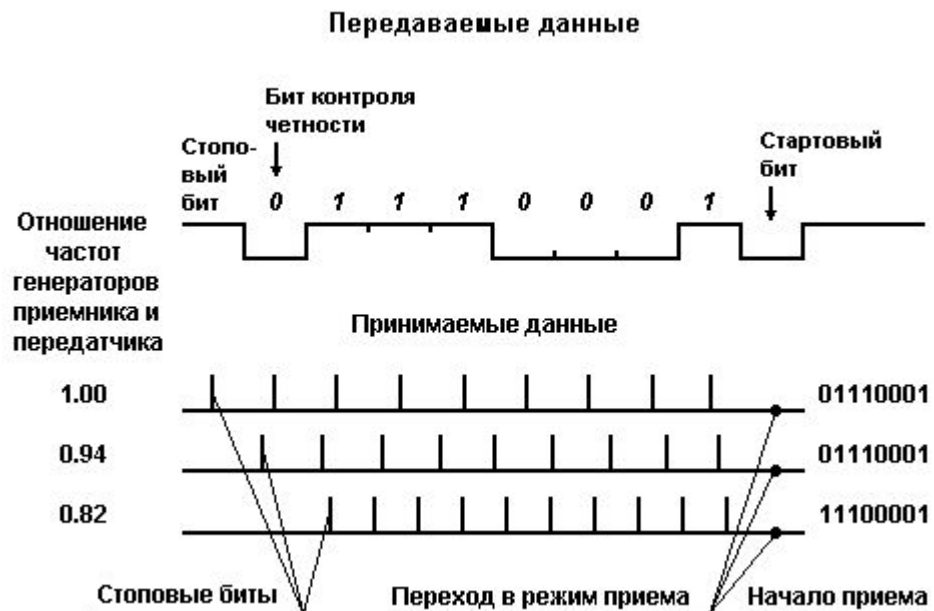
# Форматы передачи данных

Существуют два способа передачи слов информации по линиям данных: параллельный, когда одновременно пересылаются все биты слова, и последовательный, когда биты слова пересылаются поочередно, начиная, например, с его младшего разряда.

Формат асинхронной последовательной передачи данных

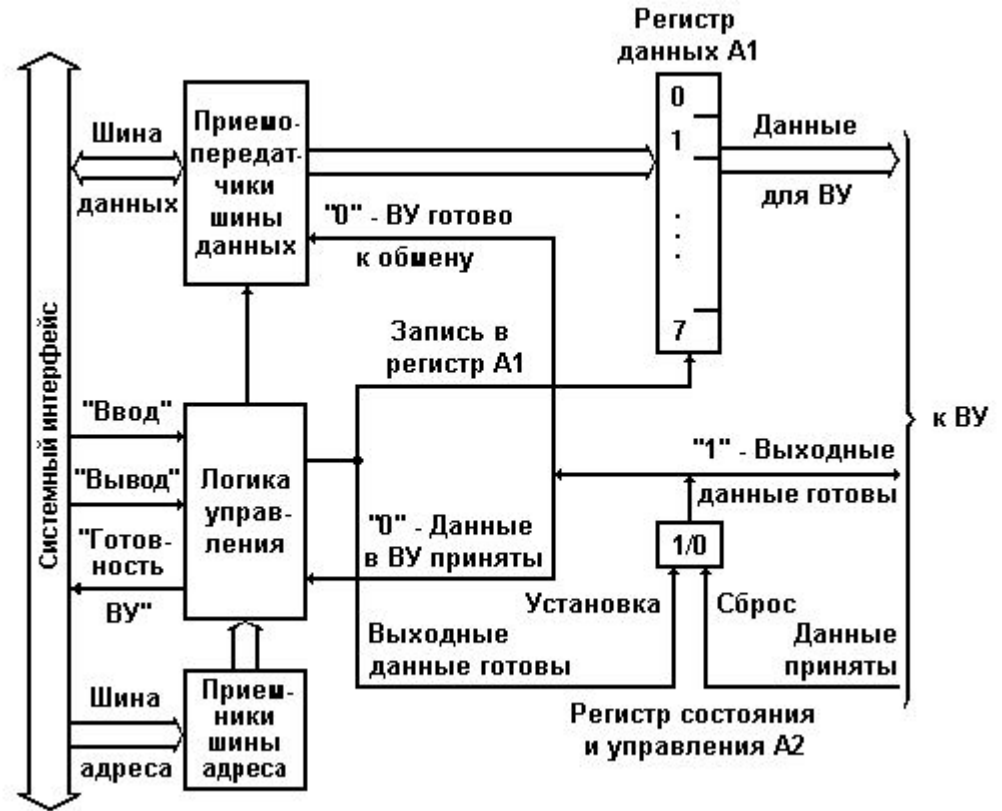


Ошибка из-за рассинхронизации генераторов передатчика и приемника

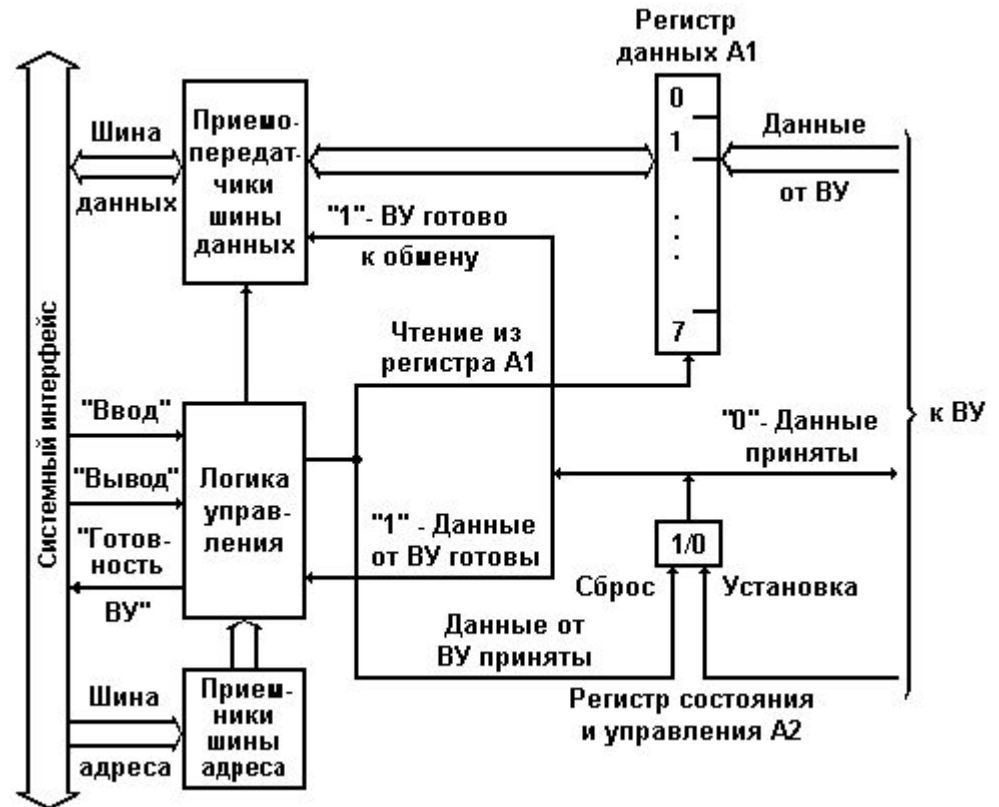


# Параллельная передача данных

В простом контроллере ВУ, обеспечивающем побайтную передачу данных на внешнее устройство в шине связи с ВУ используются всего два управляющих сигнала: "Выходные данные готовы" и "Данные приняты".



Блок-схема простого контроллера ВУ, обеспечивающего побайтный прием данных из ВУ, приведена на рис. В этом контроллере при взаимодействии с внешним устройством также используются два управляющих сигнала: "Данные от ВУ готовы" и "Данные приняты".

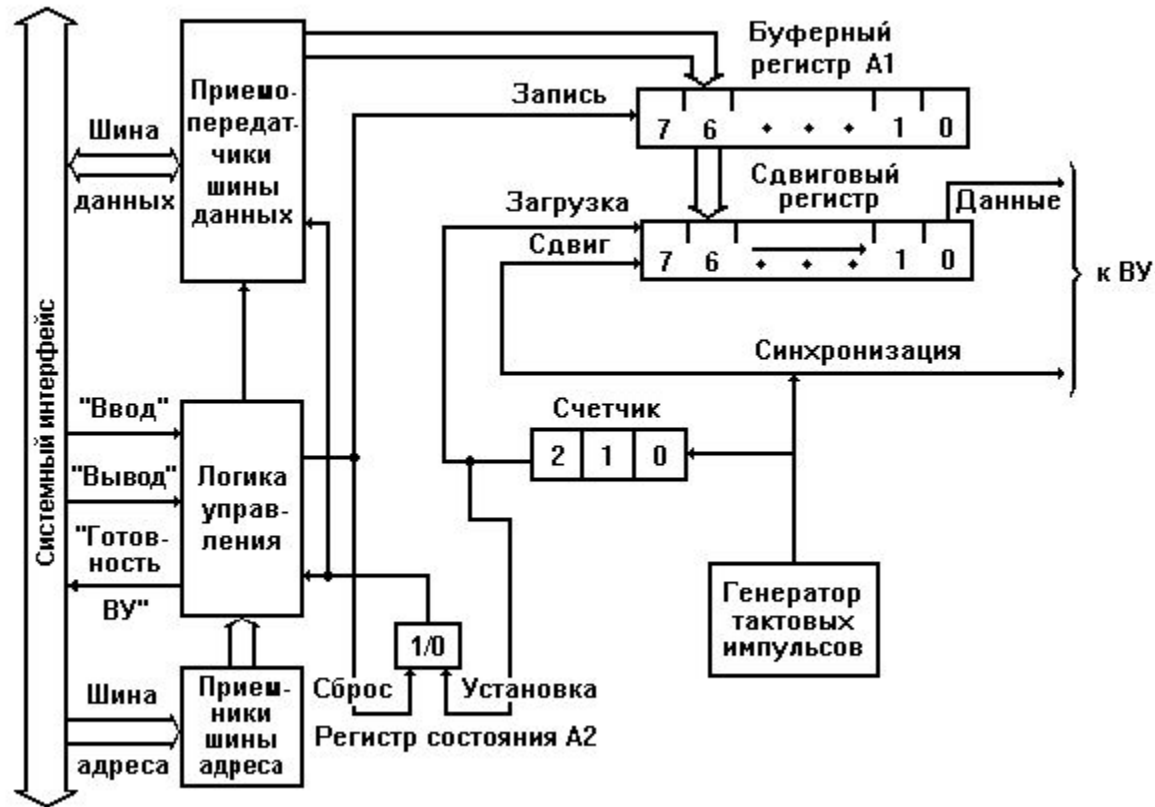






# Асинхронный последовательный интерфейс

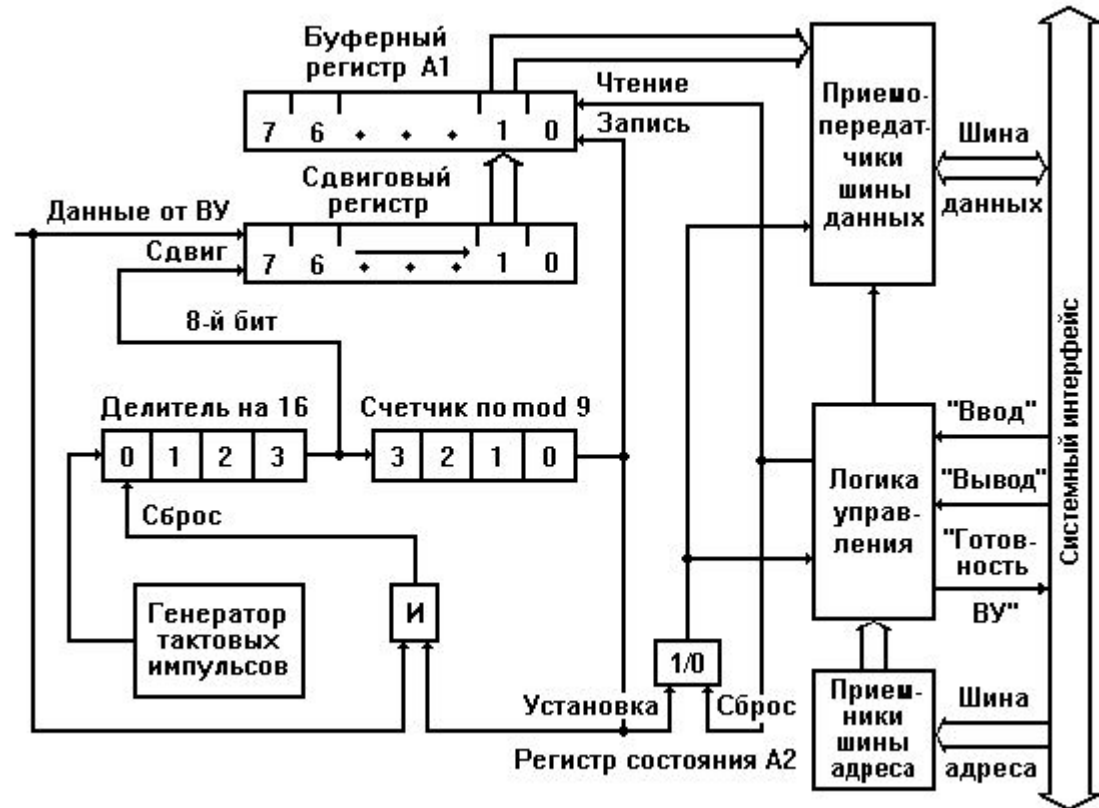
Организация асинхронного последовательного обмена данными с внешним устройством осложняется тем, что на передающей и приемной стороне последовательной линии связи используются настроенные на одну частоту, но физически разные генераторы тактовых импульсов и, следовательно, общая синхронизация отсутствует. Рассмотрим на примерах организацию контроллеров последовательных интерфейсов для последовательных асинхронных передачи и приема данных. Простейший контроллер для асинхронной передачи данных в ВУ по последовательной линии связи представлен на рис. Он предназначен для передачи данных в формате с двумя стоповыми битами.



# Контроллер последовательного асинхронного приема

Уровень логической единицы поступает по линии "Данные" в контроллер для асинхронного приема данных рис. Этот уровень создает условия для выработки сигнала, запрещающего работу делителя частоты генератора тактовых импульсов.

Действительно, после приема предыдущего байта данных счетчик импульсов сдвига (счетчик по mod 2) находится в нулевом состоянии и на вентиль И поступают два единичных сигнала: со счетчика сдвигов и из линии "Данные". На выходе вентилей И вырабатывается сигнал сброса делителя частоты сигналов тактового генератора, запрещающий формирование импульсов сдвига.



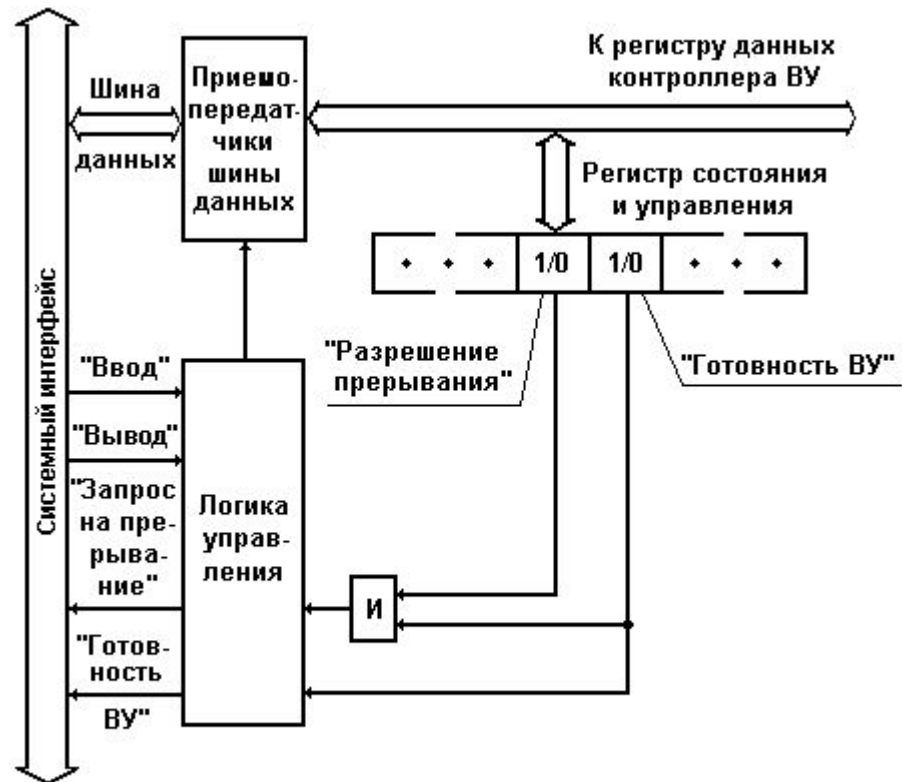
# Способы обмена информацией в микропроцессорной системе

В ЭВМ применяются три режима ввода/вывода: программно-управляемый ВВ (называемый также программным или нефорсированным ВВ), ВВ по прерываниям (форсированный ВВ) и прямой доступ к памяти. Первый из них характеризуется тем, что инициирование и управление ВВ осуществляется программой, выполняемой процессором, а внешние устройства играют сравнительно пассивную роль и сигнализируют только о своем состоянии, в частности, о готовности к операциям ввода/вывода. Во втором режиме ВВ иницируется не процессором, а внешним устройством, генерирующим специальный сигнал прерывания. Реагируя на этот сигнал готовности устройства к передаче данных, процессор передает управление подпрограмме обслуживания устройства, вызвавшего прерывание.



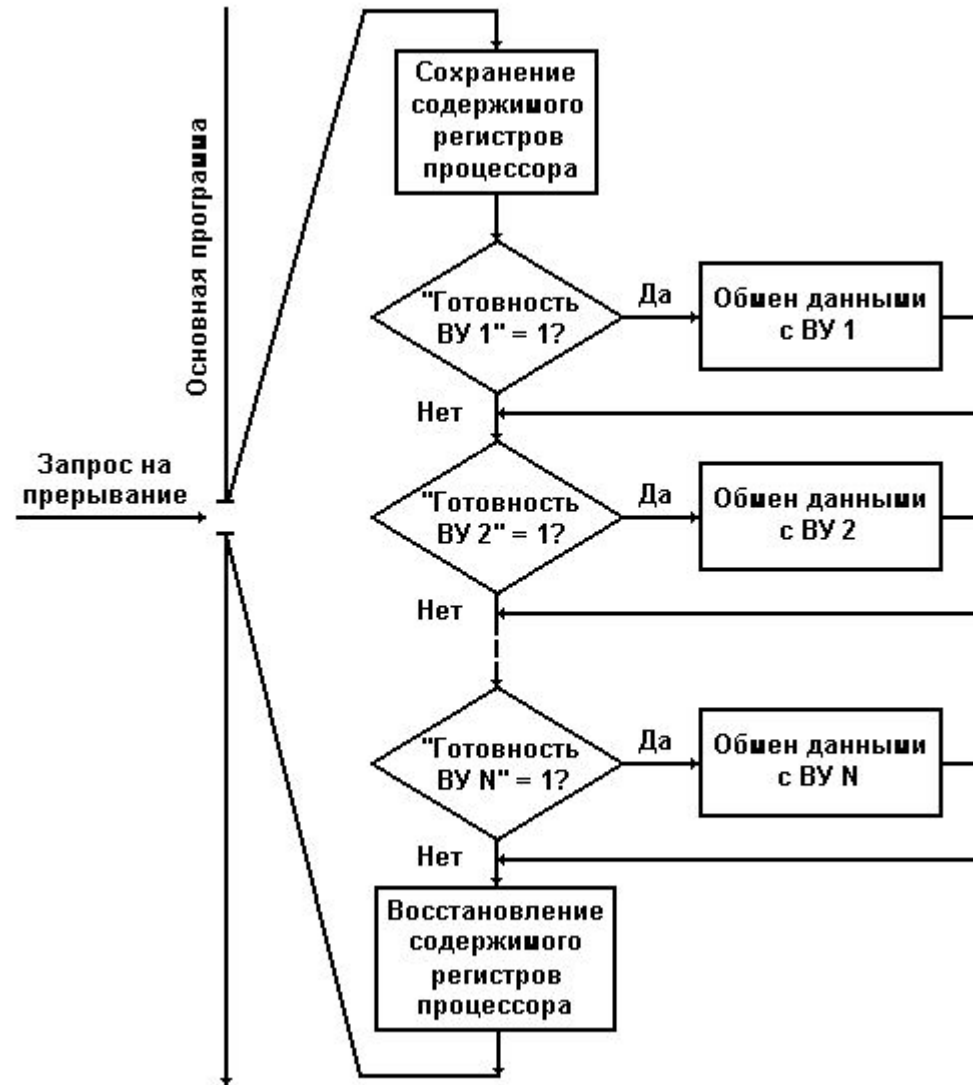
# Организация прерываний в микроЭВМ

Одной из разновидностей программно-управляемого обмена данными с ВУ в микроЭВМ является обмен с прерыванием программы, отличающийся от асинхронного программно-управляемого обмена тем, что переход к выполнению команд, физически реализующих обмен данными, осуществляется с помощью специальных аппаратных средств. Команды обмена данными в этом случае выделяют в отдельный программный модуль - подпрограмму обработки прерывания. Задачей аппаратных средств обработки прерывания в процессоре микроЭВМ как раз и является приостановка выполнения одной программы (ее еще называют основной программой) и передача управления подпрограмме обработки прерывания.



## Структура единой программы обработки прерываний и ее связь основной программой

Организация прерываний с программным опросом готовности предполагает наличие в памяти микроЭВМ единой подпрограммы обслуживания прерываний от всех внешних устройств. Структура такой подпрограммы приведена на рис.





Реализация приоритетов ВУ в микроЭВМ с векторной системой прерываний с интерфейсным вектором (ППР (вх.) - "Предоставление прерывания (входной)"; "ППР (вых.) - Предоставление прерывания (выходной)")



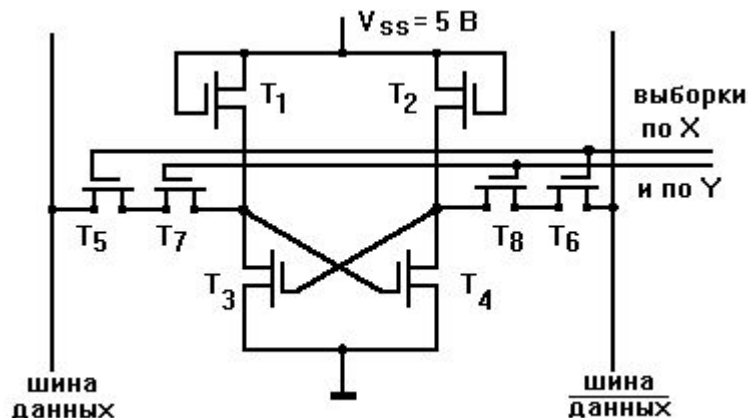




## Память в микропроцессорной системе

Для функционирования компьютерной системы необходимо наличие как оперативного запоминающего устройства (ОЗУ), так и постоянного запоминающего устройства (ПЗУ), обеспечивающего сохранение информации при выключении питания. ОЗУ может быть статическим и динамическим, а ПЗУ однократно или многократно программируемым.

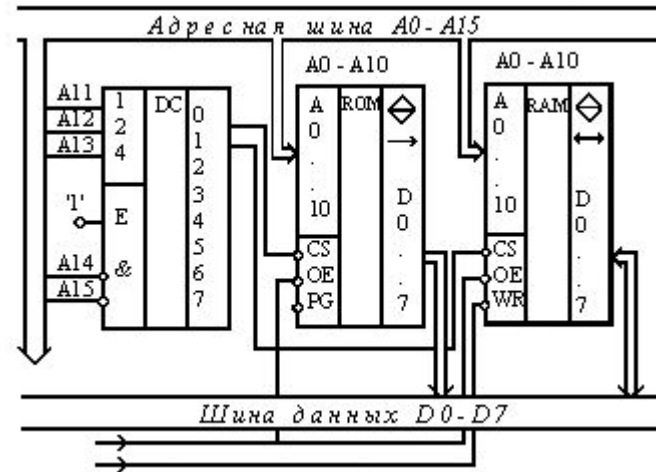
- Ячейка статического ОЗУ.



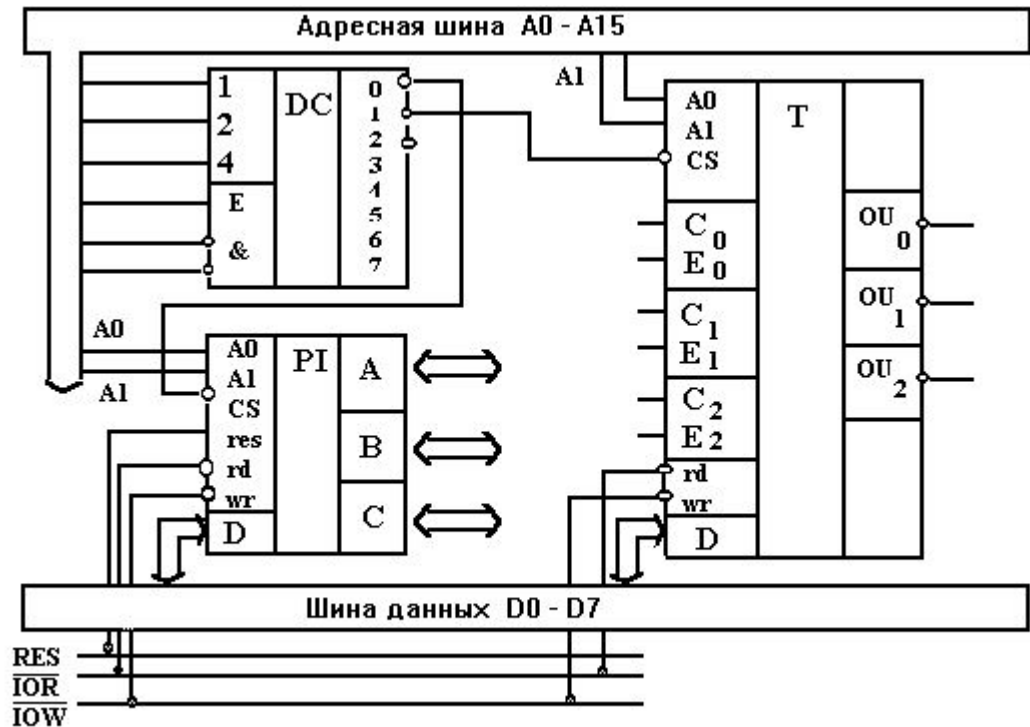
- Запоминающая ячейка динамического ОЗУ.



# Микросхемы ОЗУ (К573РУ9) и ППЗУ (К573РФ5) в составе микропроцессорной системы.



Пример взаимодействия программируемых ИМС параллельного порта (PI) КР580ВВ55А и таймера (Т) КР580ВИ53 с системной магистралью компьютера; DC - дешифратор К555ИД7.



# Буферная память

В вычислительных системах используются подсистемы с различным быстродействием, и, в частности, с различной скоростью передачи данных. Обычно обмен данными между такими подсистемами реализуется с использованием прерываний или канала прямого доступа к памяти. В первую очередь подсистема 1 формирует запрос на обслуживание по мере готовности данных к обмену. Однако обслуживание прерываний связано с непроизводительными потерями времени и при пакетном обмене производительность подсистемы 2 заметно уменьшается. Структурная схема буфера 64x4.



# Стековая память

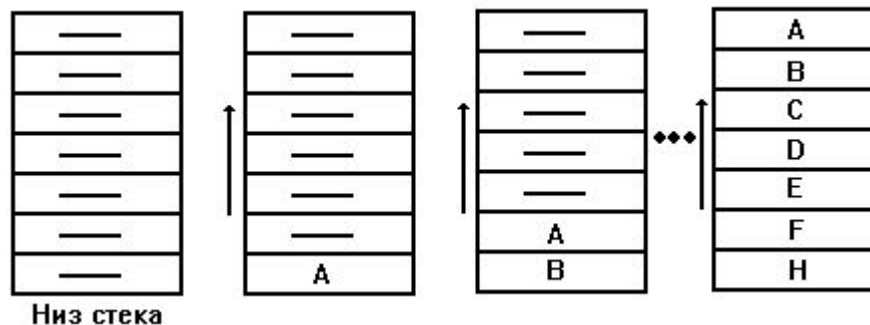
Перемещение данных при записи и считывании информации в стековой памяти подобно тому, как это имеет место в сдвигающих регистрах. С точки зрения реализации механизма доступа к стековой памяти выделяют аппаратный и аппаратно-программный (внешний) стеки.

Аппаратный стек представляет собой совокупность регистров, связи между которыми организованы таким образом, что при записи и считывании данных содержимое стека автоматически сдвигается.

Обычно емкость аппаратного стека ограничена диапазоном от нескольких регистров до нескольких десятков регистров, поэтому в большинстве МП такой стек используется для хранения содержимого программного счетчика и его называют стеком команд.

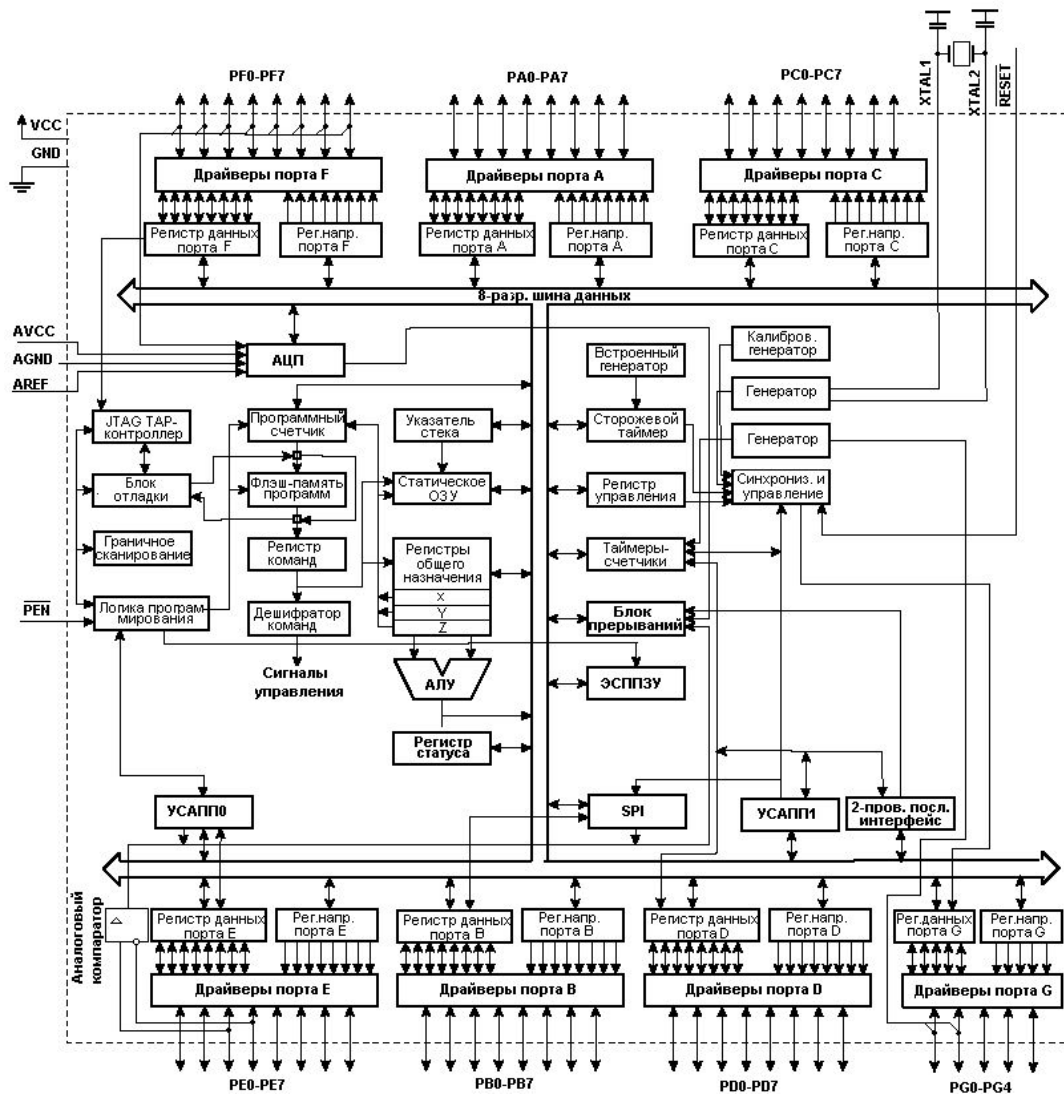
Основное достоинство аппаратного стека - высокое быстродействие, а недостаток - ограниченная емкость.

ты стековой памяти.



# Современные микроконтроллеры

8-разрядный AVR-микроконтроллер с внутрисистемной программируемой флэш-памятью емкостью 128 кбайт ATmega128, ATmega128L



## МК

Высокопроизводительный, маломощный 8-разрядный AVR-микроконтроллер Развитая RISC-архитектура

- 133 мощных инструкций, большинство из которых выполняются за один машинный цикл
- 32 8-разр. регистров общего назначения + регистры управления встроенной периферией
- Полностью статическая работа
- Производительность до 16 млн. операций в секунду при тактовой частоте 16 МГц
- Встроенное умножающее устройство выполняет умножение за 2 машинных цикла

Энергонезависимая память программ и данных

- Износостойкость 128-ми кбайт внутрисистемно перепрограммируемой флэш-памяти: 1000 циклов запись/стирание
- Опциональный загрузочный сектор с отдельной программируемой защитой

Внутрисистемное программирование встроенной загрузочной программой

- Гарантированная двухоперационность: возможность чтения во время записи
- – Износостойкость 4 кбайт ЭСППЗУ: 100000 циклов запись/стирание
- Встроенное статическое ОЗУ емкостью 4 кбайт
- Опциональная возможность адресации внешней памяти размером до 64 кбайт
- Программируемая защита кода программы
- Интерфейс SPI для внутрисистемного программирования
- Интерфейс JTAG (совместимость со стандартом IEEE 1149.1)
- Граничное сканирование в соответствии со стандартом JTAG
- Обширная поддержка функций встроенной отладки
- Программирование флэш-памяти, ЭСППЗУ, бит конфигурации и защиты через интерфейс JTAG

## МК

Отличительные особенности периферийных устройств

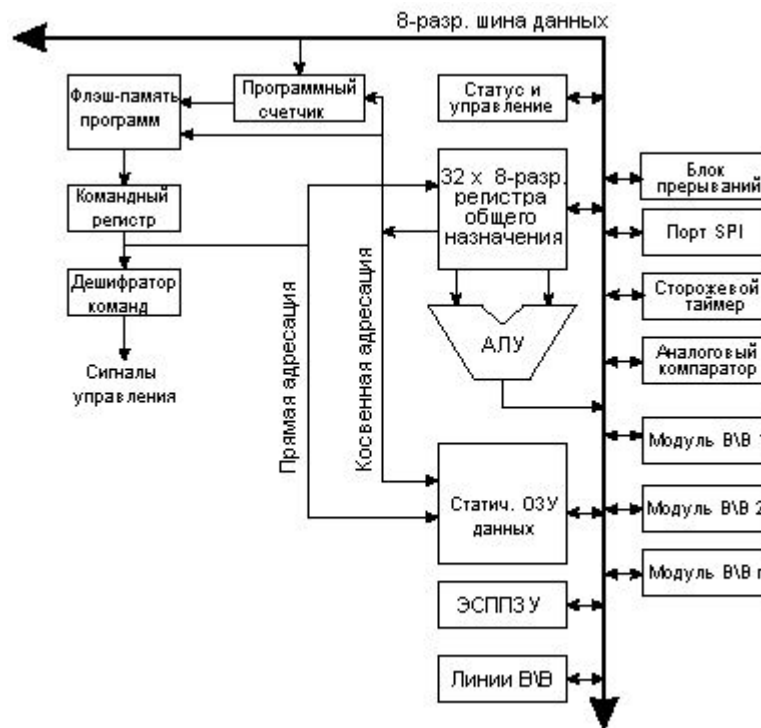
- Два 8-разр. таймера-счетчика с отдельными предделителями и режимами сравнения
- Два расширенных 16-разр. таймера-счетчика с отдельными предделителями, режимами сравнения и режимами захвата
- Счетчик реального времени с отдельным генератором
- Два 8-разр. каналов ШИМ
- 6 каналов ШИМ с программируемым разрешением от 2 до 16 разрядов
- Модулятор выходов сравнения
- 8 мультиплексированных каналов 10-разрядного аналогово-цифрового преобразования

- 8 несимметричных каналов
- 7 дифференциальных каналов
- 2 дифференциальных канала с выборочным усилением из 1x, 10x и 200x
- – Двухпроводной последовательный интерфейс, ориентированный на передачу данных в байтном формате
- – Два канала программируемых последовательных УСАПП
- – Последовательный интерфейс SPI с поддержкой режимов ведущий/подчиненный
- – Программируемый сторожевой таймер со встроенным генератором
- – Встроенный аналоговый компаратор
- Специальные возможности микроконтроллера
- – Сброс при подаче питания и программируемая схема сброса при снижении напряжения питания
- – Встроенный калиброванный RC-генератор
- – Внешние и внутренние источники прерываний
- – Шесть режимов снижения энергопотребления: холостой ход (Idle), уменьшение шумов АЦП, экономичный (Power-save), выключение (Power-down), дежурный (Standby) и расширенный дежурный (Extended Standby)
- – Программный выбор тактовой частоты
- – Конфигурационный бит для перевода в режим совместимости с ATmega103
- – Общее выключение подтягивающих резисторов на всех линиях портов ввода-вывода
- Ввод-вывод и корпуса
- – 53 – программируемые линии ввода-вывода
- – 64-выв. корпус TQFP
- Рабочие напряжения
- – 2.7 - 5.5В для ATmega128L
- – 4.5 - 5.5В для ATmega128
- Градации по быстродействию
- – 0 - 8 МГц для ATmega128L
- – 0 - 16 МГц для ATmega128



# Краткий обзор архитектуры

В целях достижения максимальной производительности и параллелизма у AVR-микроконтроллеров используется Гарвардская архитектура с отдельными памятью и шинами программ и данных. Команды в памяти программ выполняются с одноуровневой конвейеризацией. В процессе выполнения одной инструкции следующая предварительно считывается из памяти программ. Данная концепция позволяет выполнять одну инструкцию за один машинный цикл. Память программ представляет собой внутрисистемно программируемую флэш-память.



# **АЛУ – арифметико-логическое устройство**

**Высокопроизводительное АЛУ AVR-микроконтроллеров работает в непосредственной связи со всеми 32 универсальными рабочими регистрами.**

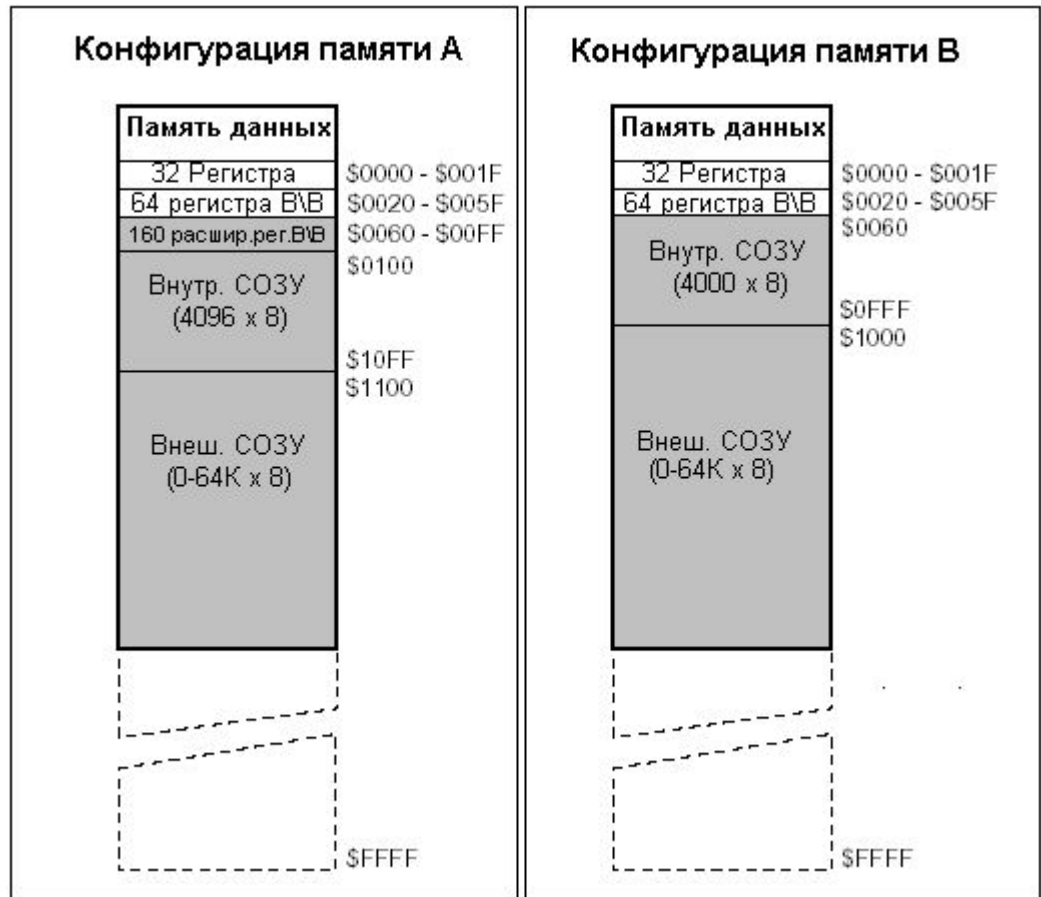
**АЛУ позволяет выполнить за один машинный цикл операцию между двумя регистрами или между регистром и константой.**

**Операции АЛУ могут быть классифицированы на три группы: арифметические, логические и битовые. Кроме того, архитектурой ATmega128 поддерживаются операции умножения со знаком и без знака и дробным форматом.**

# Статическое ОЗУ памяти данных

АТmega128 поддерживает две различные конфигурации статического ОЗУ памяти данных ).

Конфигурация Встроенное статическое ОЗУ памяти данных  
Внешнее статическое ОЗУ памяти данных  
Нормальный режим 4096 до 64 кбайт  
Режим совместимости с АТmega103 4000 до 64



## Адресация

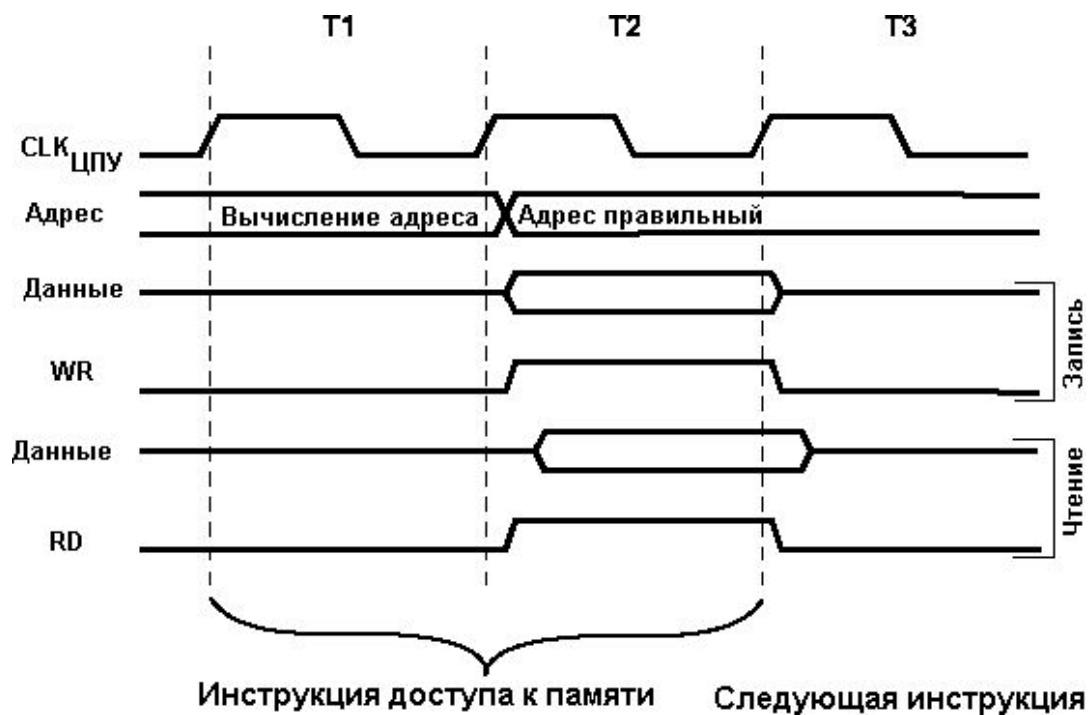
Реализовано пять различных способов адресации для охвата всей памяти: прямая, косвенная со смещением, косвенная, косвенная с предварительным декрементом и косвенная с последующим инкрементом. Регистры R26...R31 из файла регистров используются как регистры-указатели для косвенной адресации.

Прямая адресация позволяет адресоваться ко всей памяти данных.

Косвенная адресация со смещением позволяет адресовать 63 ячейки, начиная с адреса указанного в регистрах Y или Z.

- 32 рабочих регистров общего назначения, 64 регистра ввода-вывода и 4096 байт внутреннего статического ОЗУ данных в ATmega128 доступны с помощью всех этих режимов адресации. Файл регистров описывается в разделе “Файл регистров общего назначения”.

# Временная диаграмма доступа к памяти

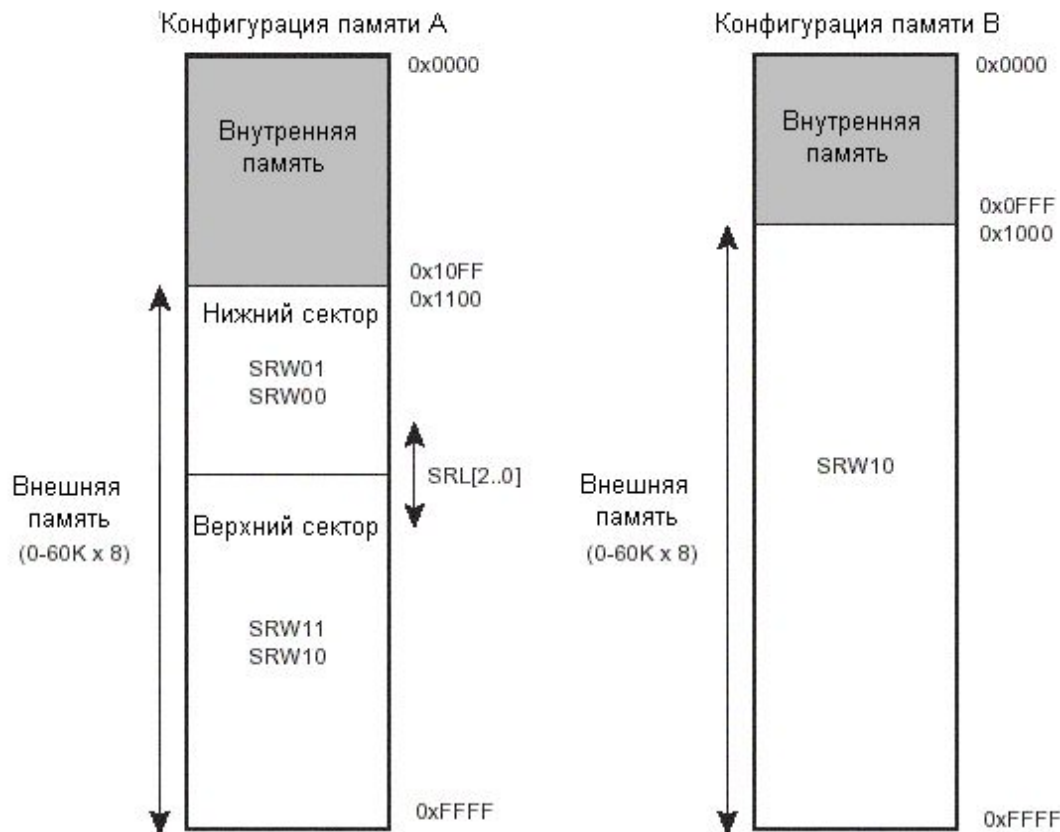


# Интерфейс внешней памяти

Характеристики интерфейса внешней памяти позволяет его использовать не только для подключения к внешнему статическому ОЗУ или флэш-памяти, но и в качестве интерфейса с внешними периферийными устройствами, например, ЖК-дисплеи, АЦП и ЦАП. Его основными отличительными особенностями являются:

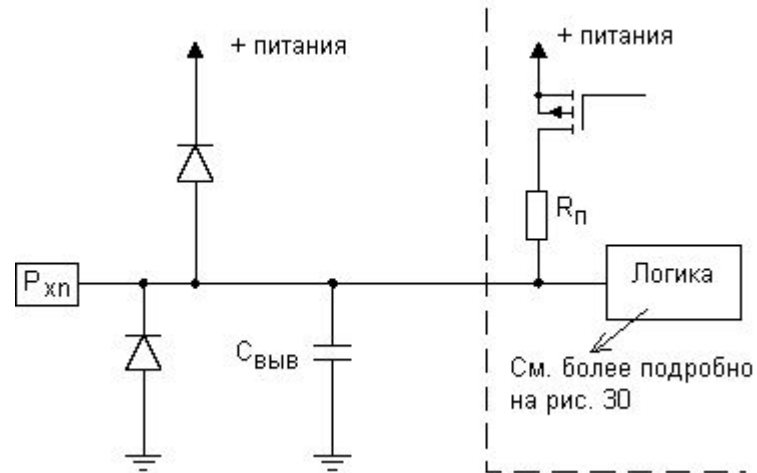
Возможность задания четырех различных по длительности состояний ожидания, в т.ч. без состояния ожидания.

Возможность установки различных состояний ожидания для разных секторов внешней памяти (размер сектора конфигурируется).



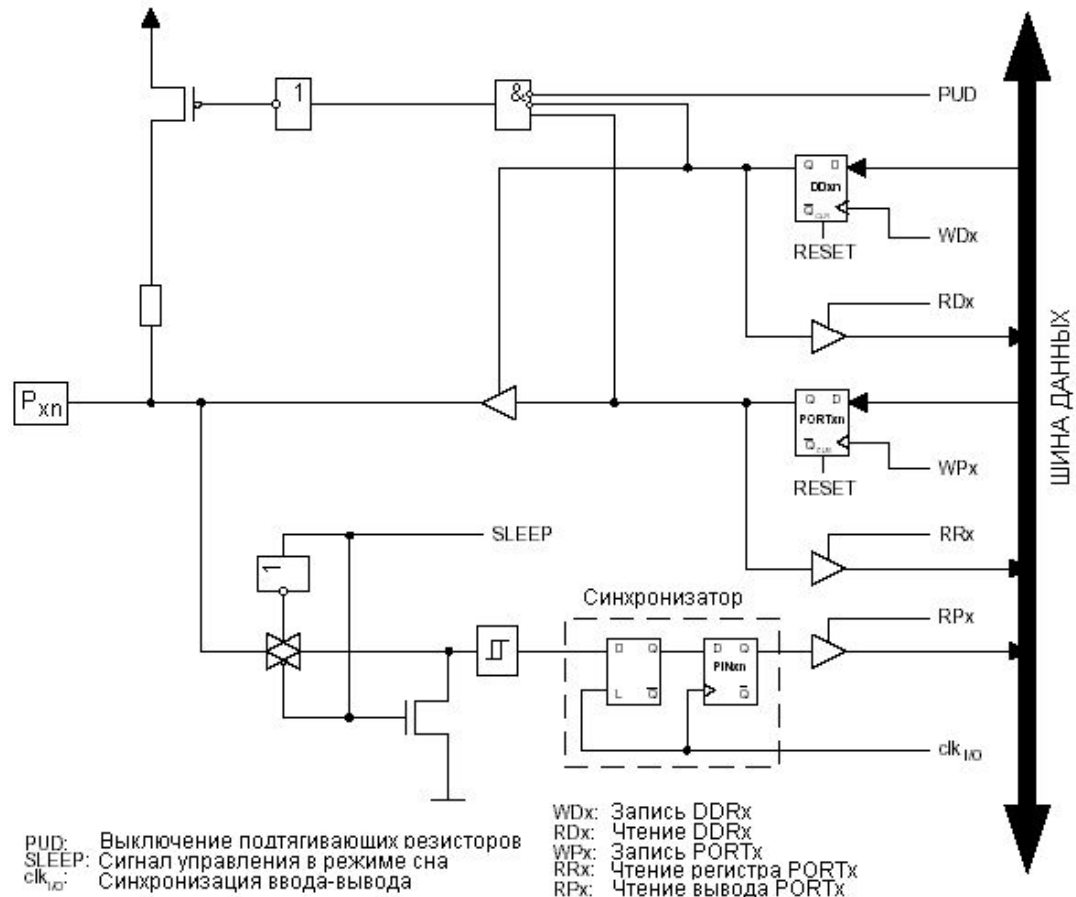
## Порты ввода-вывода

Все порты ввода-вывода (ПВВ) AVR-микроконтроллеров работают по принципу чтение-модификация-запись при использовании их в качестве портов универсального ввода-вывода. Это означает, что изменение направления ввода-вывода одной линии порта командами SBI и CBI будет происходить без ложных изменений направления ввода-вывода других линий порта. Данное распространяется также и на изменение логического уровня (если линия порта настроена на вывод) или на включение/отключение подтягивающих резисторов (если линия настроена на ввод). Каждый выходной буфер имеет симметричную характеристику управления с высоким втекающим и вытекающим выходными токами



# Организация универсального цифрового ввода-вывода

Для каждого порта ввода-вывода в памяти ввода-вывода зарезервировано три ячейки: одна под регистр данных – PORTx, другая под регистр направления данных – DDRx и третья под состояние входов порта – PINx. Ячейка, хранящая состояние на входах портов, доступна только для чтения, а регистры данных и направления данных имеют двунаправленный доступ. Кроме того, установка бита выключения подтягивающих резисторов PUD регистра SFIOR отключает функцию подтягивания на всех выводах всех портов.





# Лекция 5

Функциональные узлы МК.  
Устройства аналогового ввода –  
вывода.

# Аналогово-цифровой преобразователь

## Отличительные особенности:

10-разрядное разрешение

Интегральная нелинейность 0.5 мл. разр.

Абсолютная погрешность  $\pm 2$  мл. разр.

Время преобразования 65 - 260 мкс.

Частота преобразования до 15 тыс. преобр. в сек. при максимальном разрешении

8 мультимплексированных однополярных входов

7 дифференциальных входных каналов

2 дифференциальных входных канала с опциональным усилением на 10 и 200

Представление результата с левосторонним или правосторонним выравниванием в 16-разр. слове

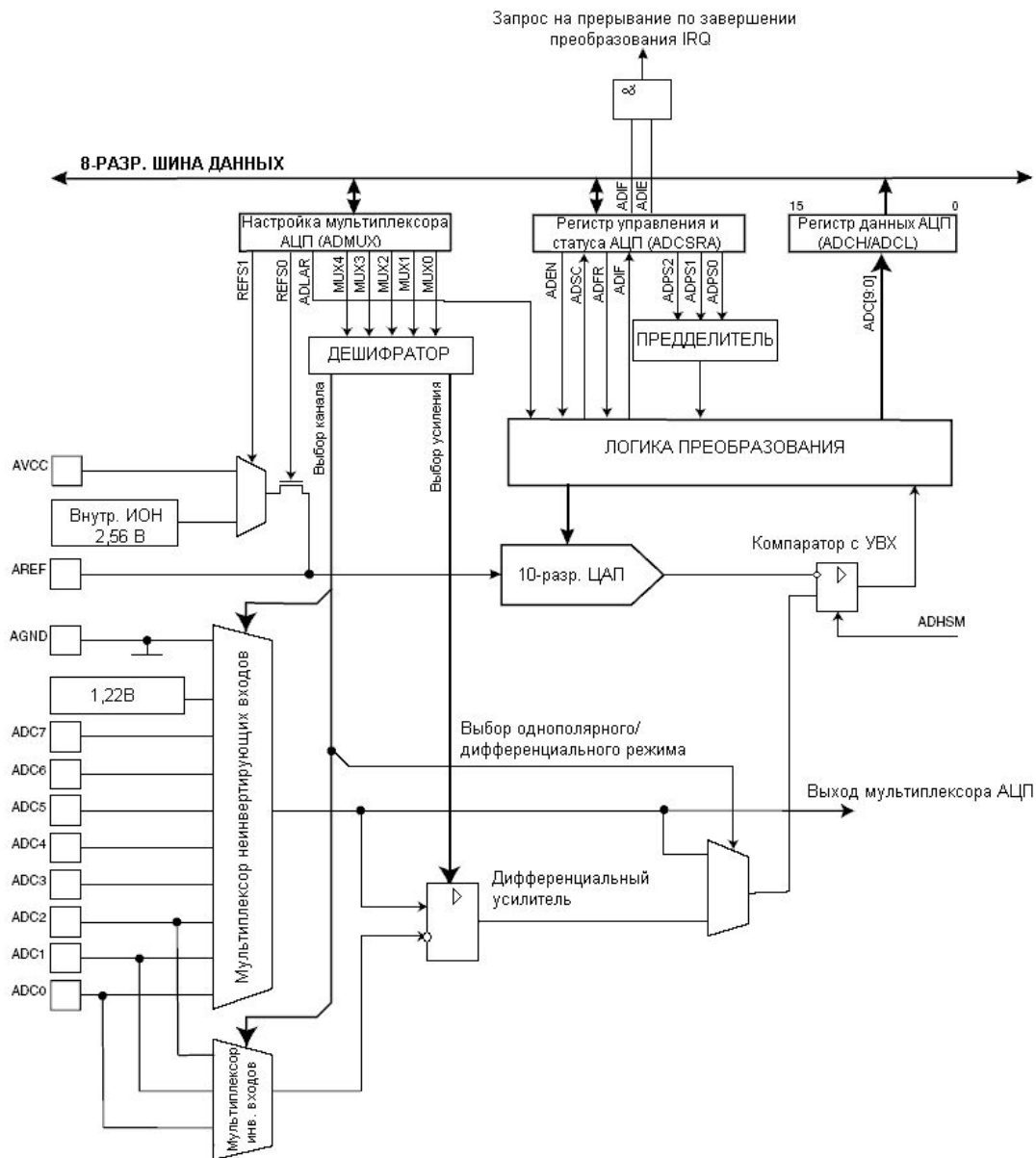
Диапазон входного напряжения АЦП 0...VCC

Выборочный внутренний ИОН на 2.56 В

Режимы одиночного преобразования и автоматического перезапуска

Прерывание по завершении преобразования АЦП

Механизм подавления шумов в режиме сна



## АЦП

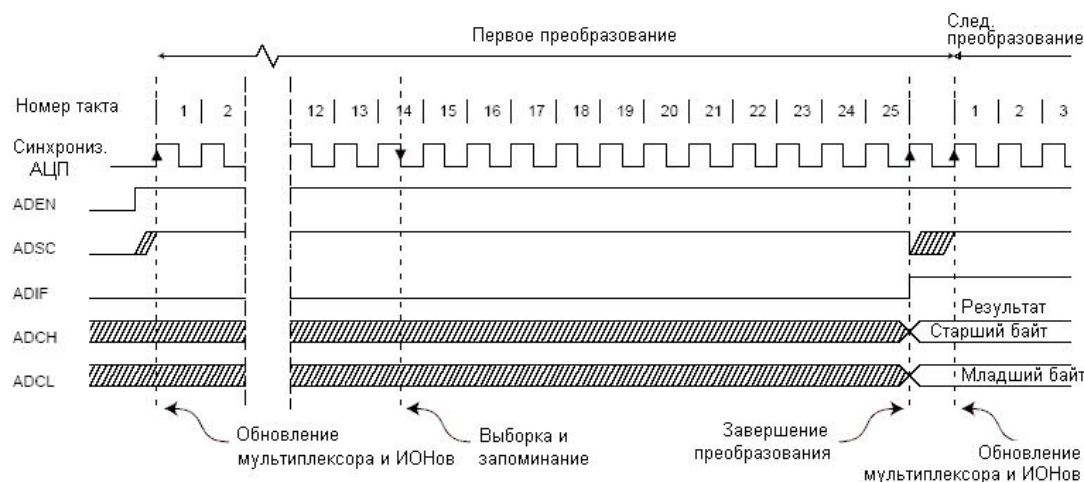
АТmega128 содержит 10-разр. АЦП последовательного приближения. АЦП связан с 8-канальным аналоговым мультиплексором, 8 однополярных входов которого связаны с линиями порта F. Общий входных сигналов должен иметь потенциал 0В (т.е. связан с GND). АЦП также поддерживает ввод 16 дифференциальных напряжений. Два дифференциальных входа (ADC1, ADC0 и ADC3, ADC2) содержат каскад со ступенчатым программируемым усилением: 0 дБ (1x), 20 дБ (10x), или 46 дБ (200x). Семь дифференциальных аналоговых каналов используют общий инвертирующий вход (ADC1), а все остальные входы АЦП выполняют функцию неинвертирующих входов. Если выбрано усиление 1x или 10x, то можно ожидать 8-разр. разрешение, а если 200x, то 7-разрядное. АЦП содержит УВХ (устройство выборки-хранения), которое поддерживает на постоянном уровне напряжение на входе АЦП во время преобразования

- АЦП имеет отдельный вывод питания AVCC (аналоговое питание). AVCC не должен отличаться более чем на  $\pm 0.3$  В от VCC.
- В качестве внутреннего опорного напряжения может выступать напряжение от внутреннего ИОНа на 2.56В или напряжение AVCC. Если требуется использование внешнего ИОН, то он должен быть подключен к выводу AREF с подключением к этому выводу блокировочного конденсатора для улучшения шумовых характеристик.

# Принцип действия АЦП

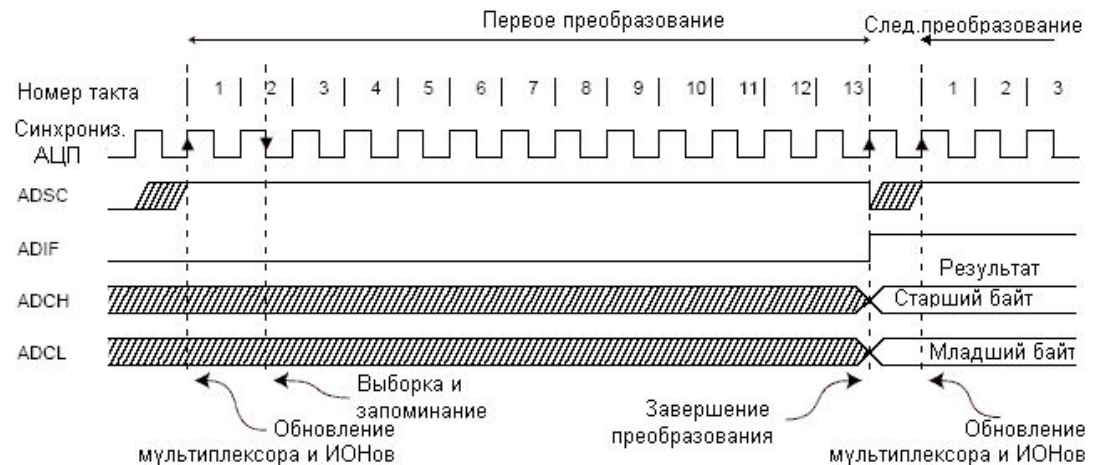
АЦП преобразовывает входное аналоговое напряжение в 10-разр. код методом последовательных приближений. Минимальное значение соответствует уровню GND, а максимальное уровню AREF минус 1 мл. разр. К выводу AREF опционально может быть подключено напряжение AVCC или внутренний ИОН на 1.22В путем записи соответствующих значений в биты REFSn в регистр ADMUX. Несмотря на то, что ИОН на 2.56В находится внутри микроконтроллера, к его выходу может быть подключен блокировочный конденсатор для снижения чувствительности к шумам, т.к. он связан с выводом AREF.

Канал аналогового ввода и каскад дифференциального усиления выбираются путем записи бит MUX в регистр ADMUX. В качестве однополярного аналогового входа АЦП может быть выбран один из входов ADC0...ADC7, а также GND и выход фиксированного источника опорного напряжения 1,22 В. В режиме дифференциального ввода предусмотрена возможность выбора инвертирующих и неинвертирующих входов к дифференциальному усилителю.



# Временная диаграмма работы АЦП в режиме одиночного преобразования

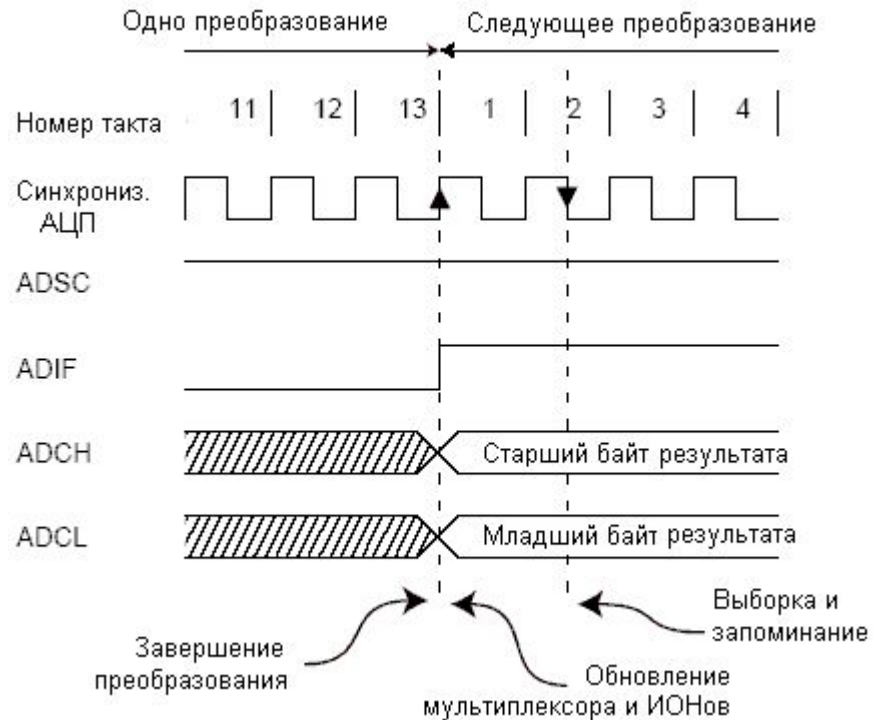
Если выбран дифференциальный режим аналогового ввода, то дифференциальный усилитель будет усиливать разность напряжений между выбранной парой входов на заданный коэффициент усиления. Усиленное таким образом значение поступает на аналоговый вход АЦП. Если выбирается однополярный режим аналогового ввода, то каскад усиления пропускается. Работа АЦП разрешается путем установки бита ADEN в ADCSRA. Выбор опорного источника и канала преобразования не возможно выполнить до установки ADEN. Если ADEN = 0, то АЦП не потребляет ток, поэтому, при переводе в экономичные режимы сна рекомендуется предварительно отключить АЦП.



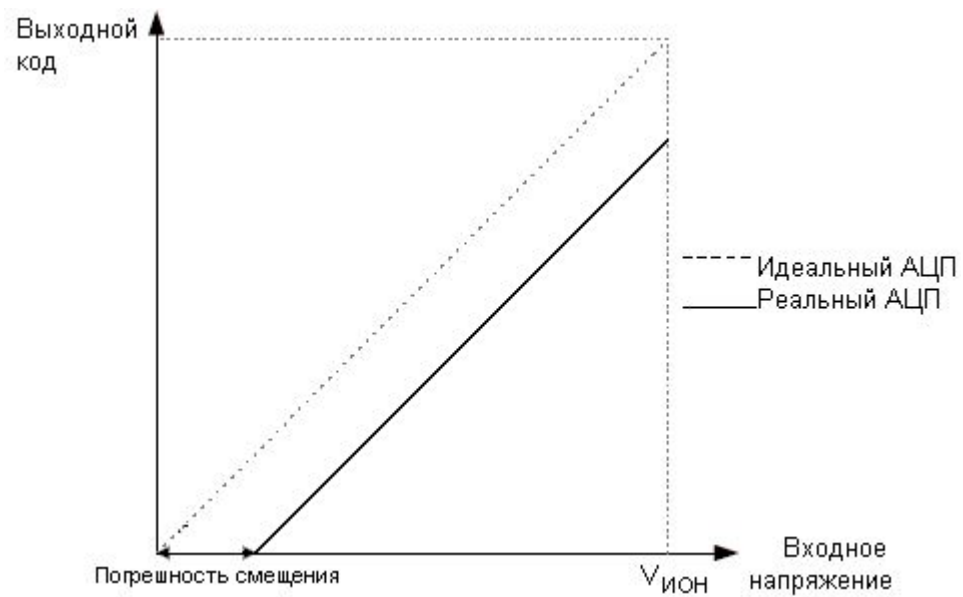
# Временная диаграмма работы АЦП в режиме автоматического перезапуска

АЦП генерирует 10-разрядный результат, который помещается в пару регистров данных АЦП ADCH и ADCL. По умолчанию результат преобразования размещается в младших 10-ти разрядах 16-разр. слова (выравнивание справа), но может быть опционально размещен в старших 10-ти разрядах (выравнивание слева) путем установки бита ADLAR в регистре ADMUX.

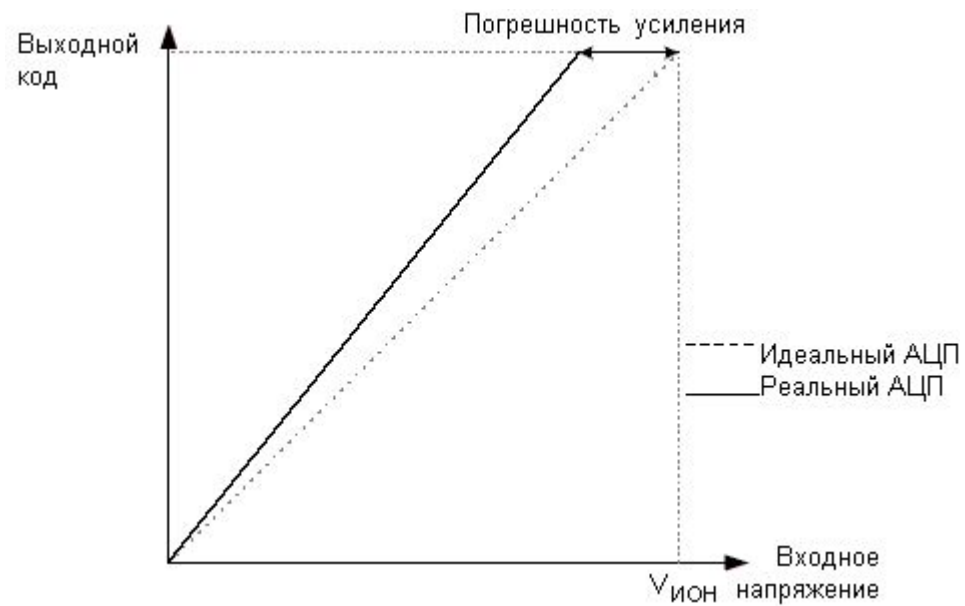
Практическая полезность представления результата с выравниванием слева существует, когда достаточно 8-разрядное разрешение, т.к. в этом случае необходимо считать только регистр ADCH. В другом же случае необходимо первым считать содержимое регистра ADCL, а затем ADCH, чем гарантируется, что оба байта являются результатом одного и того же преобразования. Как только выполнено чтение ADCL блокируется доступ к регистрам данных со стороны АЦП. Это означает, что если считан ADCL и преобразование завершается перед чтением регистра ADCH, то ни один из регистров не может модифицироваться и результат преобразования теряется. После чтения ADCH доступ к регистрам ADCH и ADCL со стороны АЦП снова разрешается. АЦП генерирует собственный запрос на прерывание по завершении преобразования. Если между чтением регистров ADCH и ADCL запрещен доступ к данным для АЦП, то прерывание возникнет, даже если результат преобразования будет потерян.



# Погрешность смещения

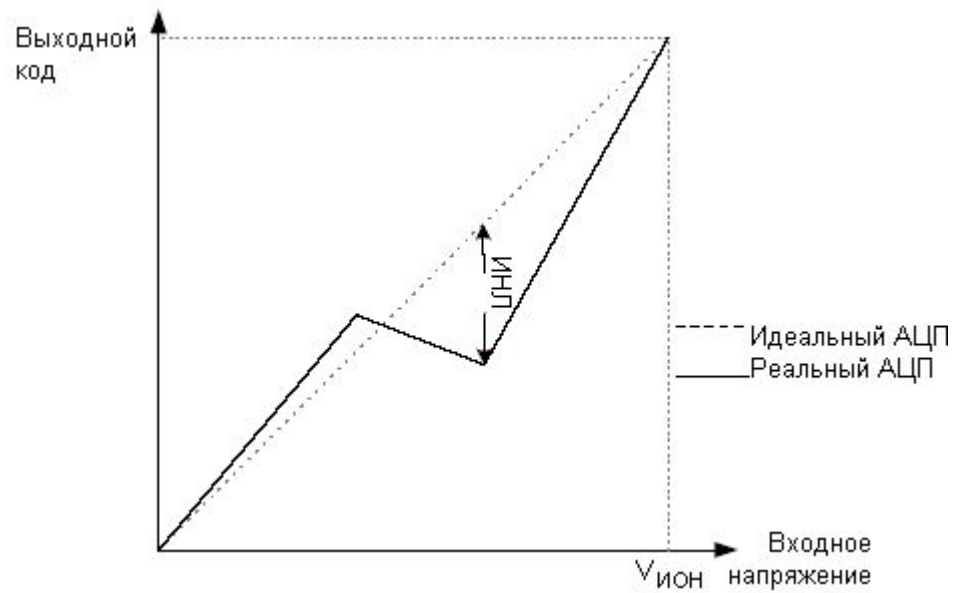


# Погрешность усиления

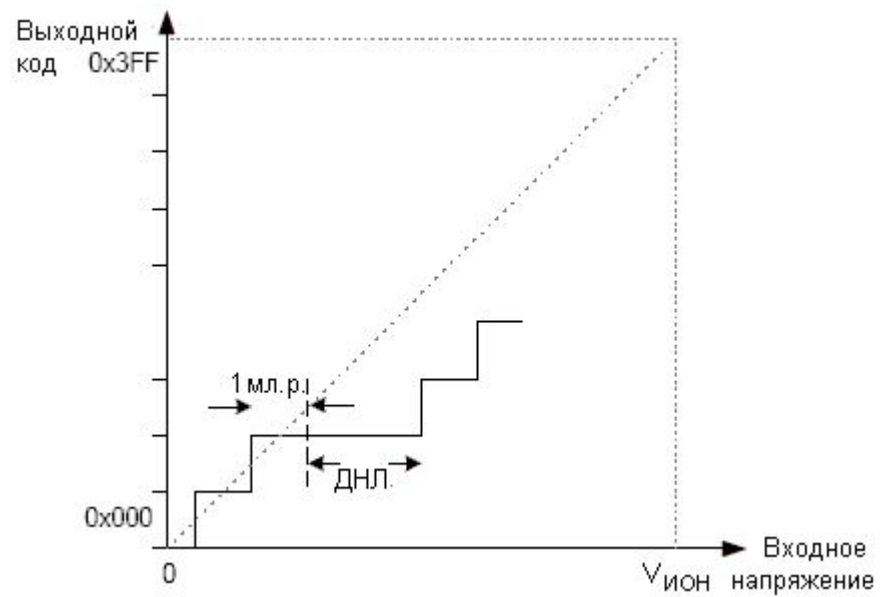




# Интегральная нелинейность (ИНЛ)



# Дифференциальная нелинейность (ДНЛ)



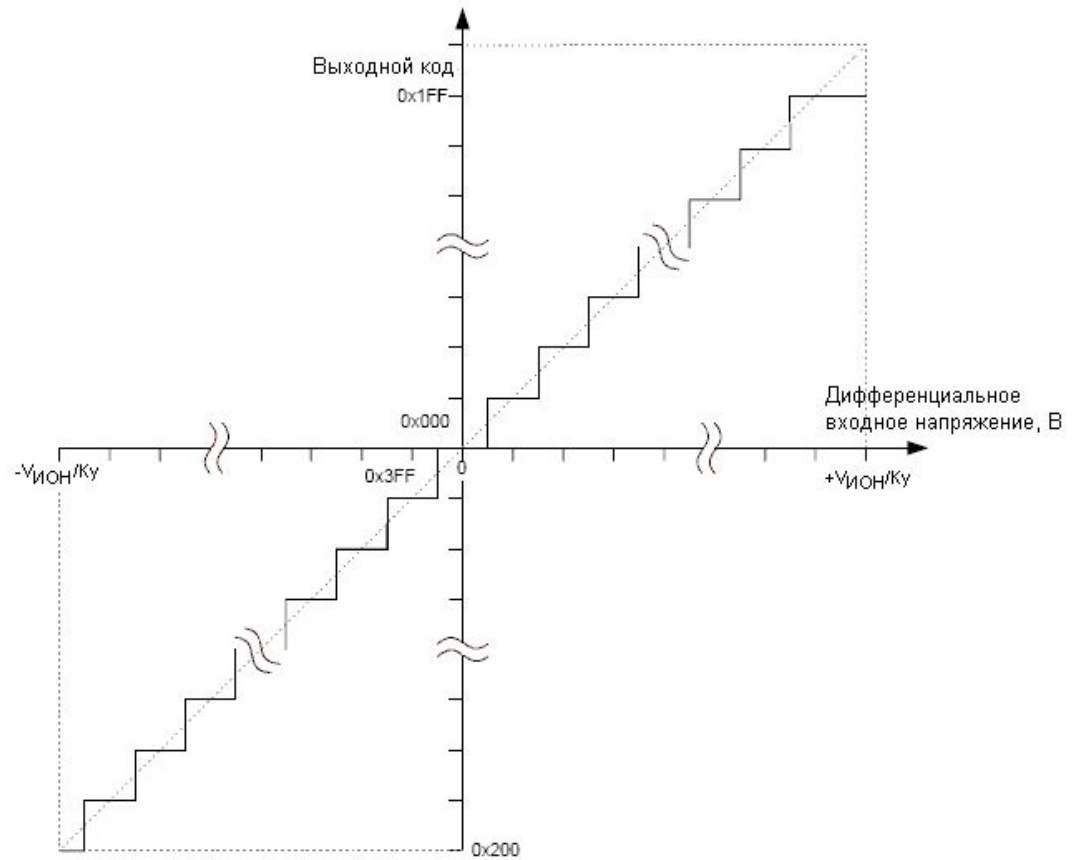
## Определение погрешности

Для однополярного и дифференциального преобразования

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot GAIN \cdot 512}{V_{REF}}$$

# Функция преобразования АЦП при измерении дифференциального сигнала



## Внешние прерывания

Внешние прерывания осуществляются через выходы INT7:0. Обратите внимание, что после разрешения внешние прерывания будут генерироваться, даже если линии INT7:0 настроены как выходы. Данная особенность может использоваться для программной генерации прерывания. Внешние прерывания могут генерироваться по подающему или нарастающему фронту, а также по низкому лог. уровню. Одна из этих установок задается в регистрах управления внешними прерываниями EICRA (INT3:0) и EICRB (INT7:4). Если внешнее прерывание разрешено и настроено на срабатывание при низком уровне, то прерывание будет инициироваться постоянно пока на выводе будет оставаться низкий уровень

# 16-разрядные таймеры-счетчики 1 и 3

## Основные отличительные особенности:

16-разрядные счетчики (в т.ч. возможность организации 16-разр. ШИМ)

Три отдельных блока сравнения  
Двойная буферизация регистров порога сравнения (OCR)

Один блок захвата

Подавитель шума на входе блока захвата

Режим сброса таймера при совпадении с порогом сравнения (автоматическая перезагрузка)

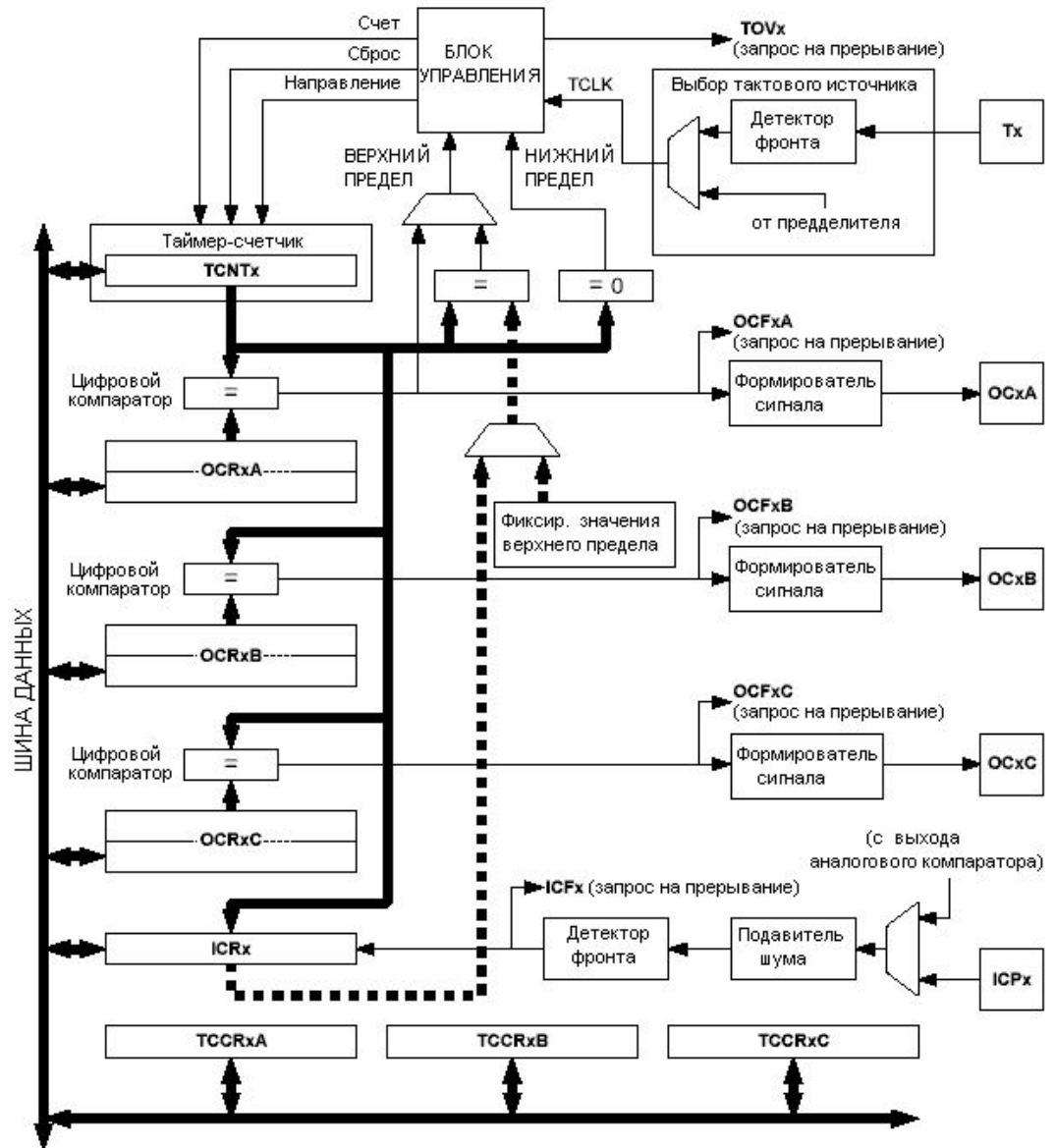
Широтно-импульсная модуляция без генераций ложных импульсов при записи нового порога сравнения в OCR (двойная буферизация) и фазовая коррекция

Переменный период ШИМ

Частотный генератор

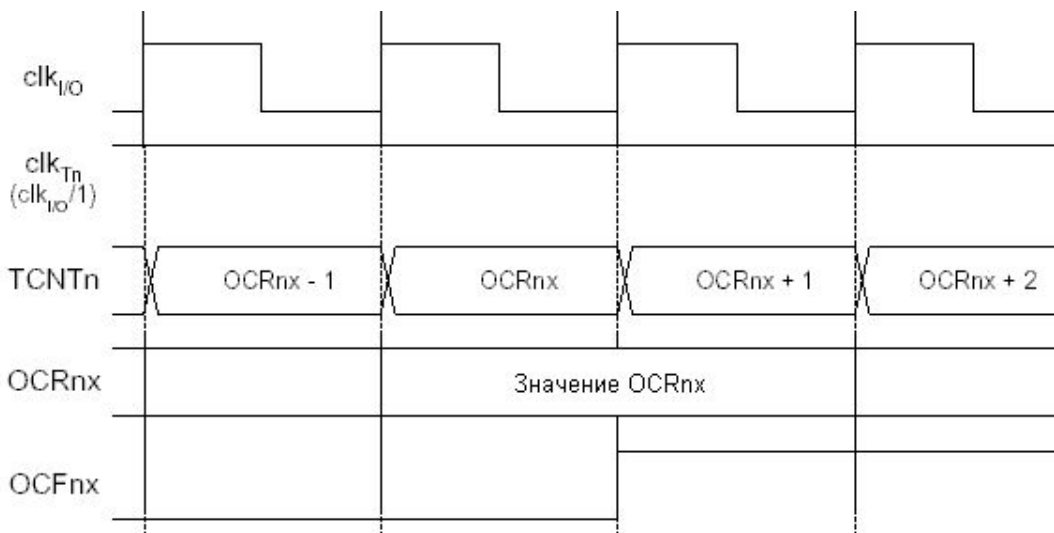
Счетчик внешних событий

10 самостоятельных источников прерываний (TOV1, OCF1A, OCF1B, OCF1C, ICF1, TOV3, OCF3A, OCF3B, OCF3C и ICF3)

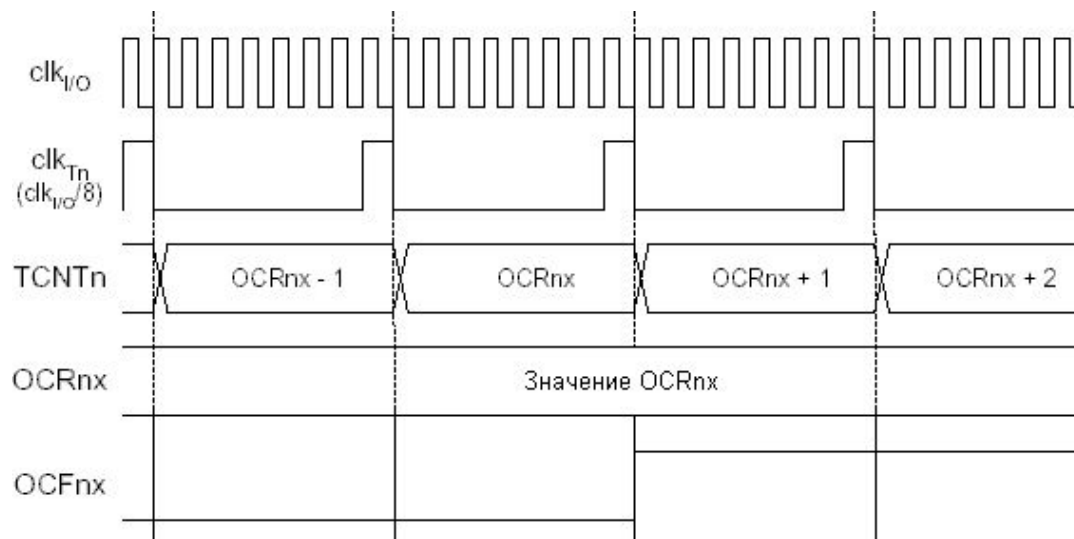


# Регистры

Регистр таймера-счетчика (TCNTn), регистры порогов сравнения (OCRnA/B/C), а также регистр захвата (ICRn) являются 16-разрядными регистрами. В связи с этим, во время доступа к этим регистрам должна быть соблюдена специальная процедура. Регистры управления таймером (TCCRnA/B/C) являются 8-разр. регистрами, поэтому, доступ к ним со стороны ЦПУ не связан с какими-либо ограничениями. Все сигналы запросов на прерывание представлены в регистре флагов прерываний таймеров (TIFR) и регистре флагов расширенных прерываний (ETIFR). Все прерывания индивидуально маскируются регистром маски прерываний таймеров (TIMSK) и регистром маски расширенных прерываний (ETIMSK). Регистры (E)TIFR и (E)TIMSK не представлены на функциональной схеме, т.к. они совместно используются другими таймерами микроконтроллера.



# Временная диаграмма таймера-счетчика с предделением на 8





# Лекция 6

ПРОЕКТИРОВАНИЕ  
МИКРОПРОЦЕССОРНЫХ  
СИСТЕМ.  
УРОВНИ ПРЕДСТАВЛЕНИЯ  
МИКРОПРОЦЕССОРНОЙ  
СИСТЕМЫ

**МИКРОПРОЦЕССОРНЫХ  
СИСТЕМ  
УРОВНИ ПРЕДСТАВЛЕНИЯ  
МИКРОПРОЦЕССОРНОЙ  
СИСТЕМЫ**

- Микропроцессорная система может быть описана, например, на одном из следующих уровней абстрактного представления:
  - 1) "черный ящик";
  - 2) структурный;
  - 3) программный;
  - 4) логический;
  - 5) схемный .

## Отладка

Дефект не может быть обнаружен до тех пор, пока не будут созданы условия для возникновения из-за него неисправности, результат которой должен быть, в свою очередь, передан на выход испытуемого объекта, для того чтобы сделать неисправность наблюдаемой. Метод испытаний должен позволить генерировать тесты, ставящие испытуемый объект в условия, при которых моделируемые неисправности проявляли бы себя в виде обнаруживаемых ошибок. Если испытуемый объект предназначен для эксплуатации, то при обнаружении ошибки необходимо произвести локализацию неисправности с целью ее устранения путем ремонта или усовершенствования испытуемого объекта.

- О правильности функционирования микропроцессорной системы на уровне "черного ящика" с полностью неизвестной внутренней структурой можно говорить лишь тогда, когда произведены ее испытания, в ходе которых реализованы все возможные комбинации входных воздействий, и в каждом случае проверена корректность ответных реакций. Однако исчерпывающее тестирование имеет практический смысл лишь для простейших элементов систем.

## Диагностика неисправности

Диагностика неисправности - процесс определения причины появления ошибки по результатам тестирования.

Отладка - процесс обнаружения ошибок и определение источников их появления по результатам тестирования при проектировании микропроцессорных систем.

Средствами отладки являются приборы, комплексы и программы

- Точность, с которой тот или иной тест локализует неисправности, называется его разрешающей способностью. Требуемая разрешающая способность определяется конкретными целями испытаний. Например, при испытаниях аппаратуры в процессе эксплуатации для ее ремонта часто необходимо установить, в каком сменном блоке изделия имеется неисправность. В заводских условиях желательно осуществлять диагностику неисправности вплоть до уровня наименьшего заменяемого элемента, чтобы минимизировать стоимость ремонта. В лабораторных условиях в процессе отладки опытного образца необходимо определять природу неисправности (физического или нефизического происхождения). В случае возникновения и проявления дефекта требуется локализовать место неисправности с точностью до заменяемого элемента, а при проявлении субъективной неисправности - с точностью до уровня представления (программного, схемного, логического и т. д.), на котором была внесена неисправность, и места.

## **Функции средств отладки**

- Средства отладки должны:
  - 1) управлять поведением системы или/и ее модели на различных уровнях абстрактного представления;
  - 2) собирать информацию о поведении системы или/и ее модели, обрабатывать и представлять на различных уровнях абстракции;
  - 3) преобразовывать системы, придавать им свойства контролепригодности;
  - 4) моделировать поведение внешней среды проектируемой системы.

## Этапы проектирования микропроцессорных систем

Микропроцессорные системы по своей сложности, требованиям и функциям могут значительно отличаться надежностными параметрами, объемом программных средств, быть однопроцессорными и многопроцессорными, построенными на одном типе микропроцессорного набора или нескольких, и т.д. В связи с этим процесс проектирования может видоизменяться в зависимости от требований, предъявляемых к системам. Например, процесс проектирования МПС, отличающихся одна от другой содержанием ПЗУ, будет состоять из разработки программ и изготовления ПЗУ.

- При проектировании многопроцессорных микропроцессорных систем, содержащих несколько типов микропроцессорных наборов, необходимо решать вопросы организации памяти, взаимодействия с процессорами, организации обмена между устройствами системы и внешней средой, согласования функционирования устройств, имеющих различную скорость работы, и т. д. Ниже приведена примерная последовательность этапов, типичных для создания микропроцессорной системы:
  1. Формализация требований к системе.
  2. Разработка структуры и архитектуры системы.
  3. Разработка и изготовление аппаратных средств и программного обеспечения системы.
  4. Комплексная отладка и приемосдаточные испытания.

## Технология разработки микропроцессорных контроллеров

Технология проектирования контроллеров на основе микропроцессоров и микроконтроллеров полностью соответствует концепции неразрывности процесса проектирования и отладки аппаратной и программной составляющих, принятой во всей микропроцессорной технике. Единый процесс проектирования микропроцессорной системы и ее отладки в англоязычной литературе обозначается словом development, мы будем употреблять термин разработка. Важной особенностью применения контроллеров является работа в реальном масштабе времени, т.е. гарантированная реакция на внешние события в течение определенного интервала времени. Очевидно, что решение задачи комплексной разработки аппаратуры и программного обеспечения в реальном масштабе времени при произвольной структуре и схемотехнике контроллера является весьма сложной, дорогостоящей и долговременной работой.

- В качестве основного метода разработки микропроцессорных систем фирмой Intel в 70-х годах был предложен метод внутрисхемной эмуляции. Основой этого метода является моделирование разрабатываемой системы с использованием средств специализированного инструментального компьютера - схемного эмулятора. В соответствии с первоначальной идеей схемный эмулятор, представляющий из себя по сути конструктор, должен был иметь все аппаратные средства, которые могли понадобиться для реализации целевой системы, плюс средства управления отладкой. Вначале контроллер конфигурировался из аппаратуры эмулятора и разрабатывалась управляющая программа, далее на пустую макетную плату устанавливалась розетка целевого микропроцессора и схемы ближайшего обрамления, эмулятор своей эмуляционной вилкой включался в розетку вместо микропроцессора и начинался поэтапный перенос аппаратных средств с соответствующими программными фрагментами из эмулятора на плату контроллера. Чтобы обеспечить такой перенос, адресное пространство памяти картируется с образованием сегментов, которые физически могут находиться как в эмуляторе, так и на целевой плате.

## Технология МК

Основным преимуществом метода внутрисхемной эмуляции является упорядоченность процесса проектирования, который может быть разделен на ряд отдельных этапов. На каждом этапе все ошибки локализованы во вновь создаваемом аппаратном и программном обеспечении, фундаментом являются заведомо работоспособные средства эмулятора. Основным недостатком такого глобального подхода к эмуляции является дороговизна инструментальных средств. Это усугублялось в первое время тем, что до появления персональных ЭВМ фирма Intel пошла по пути создания отладочных комплексов типа Intellec в виде собственной специализированной мини-ЭВМ с оригинальной операционной системой ISIS.

- В настоящее время средства моделирования в схемных эмуляторах, которые мы называем имитирующим процессором, в большинстве случаев замещают только целевую микропроцессорную БИС и память. Разработка остальных частей контроллера, который в настоящее время часто представляет собой сложную систему управления, ложится на разработчика.



## Технология МК

Особенностью микропроцессорных контроллеров является то, что они сами интегрируются в некоторый объект (embedded controllers). Это предполагает, что перед разработчиком микропроцессорной системы такого рода стоит задача полного цикла проектирования, начиная от разработки алгоритма функционирования и заканчивая комплексными испытаниями в составе изделия, а возможно и сопровождением при производстве. Т

- Требования начинают цикл проектирования микропроцессорного контроллера. Особенностью именно микропроцессорных контроллеров является то, что возможности их программирования подвигают заказчика заложить максимально широкие функции управления, чтобы иметь возможность использовать контроллер для управления целой гаммой аналогичных приборов. Критерием выбора должна служить экономическая целесообразность любого увеличения объема аппаратных средств, что определяется в результате исследования рынка приборов данного типа, и максимальное улучшение показателя цена/функциональные\_возможности. На этом этапе явно или неявно формулируются требования к типу используемого микропроцессора или микроконтроллера.
- Этап разработки алгоритма управления является наиболее ответственным, поскольку ошибки этого этапа обнаруживаются при испытаниях законченного изделия и приводят к дорогостоящей переработке всей системы управления. Прорабатывается несколько вариантов алгоритма, обеспечивающих выполнение технических требований с использованием наработанных ранее функционально-топологических модулей.

## Технология МК

Основные варианты отличаются соотношением объема программного обеспечения и аппаратуры.

- Критерием выбора является максимальное увеличение программы и уменьшение аппаратуры при обеспечении заданных показателей быстродействия и надежности в полном диапазоне эксплуатационных воздействий. Часто определяющим требованием является возможность размещения кода управляющей программы во внутренней памяти микроконтроллера, что позволяет обеспечить ее защиту. На этом этапе окончательно определяется тип микропроцессорной БИС и важнейших схем обрамления (flash-памяти, ПЛИС, программируемых интерфейсов, АЦП и т.п.).

## Технология МК

Содержание этапов разработки исходного текста программы, трансляции и отладки логических связей на модели существенно зависит от используемых системных средств. В настоящее время ресурсы 8-разрядных микроконтроллеров достаточны для поддержки программирования на языках высокого уровня. Это позволяет использовать все преимущества структурного программирования, разрабатывать программное обеспечение как проект с использованием отдельно транслируемых модулей. Одновременно продолжают широко использоваться языки уровня ассемблера, особенно при необходимости обеспечить контролируемые интервалы времени.

- На этапе разработки структуры микропроцессорного контроллера окончательно определяется состав имеющихся и подлежащих разработке аппаратных модулей, протоколы обмена между модулями, типы разъемов. Поскольку контроллер встраивается в изделие, выполняется предварительная проработка конструкции плат. В части программного обеспечения определяется состав и связи программных модулей, язык программирования. На этом же этапе производится выбор средств проектирования и отладки.

## Квазипараллельные процессы в микропроцессорных системах управления

Задачи управления в микропроцессорных системах решаются как программно, так и аппаратно. Программную часть реализации функций управления осуществляет модуль центрального процессора, аппаратную - специализированные интерфейсные модули. Эти модули: последовательный порт, контроллер прямого доступа к памяти и другие, - выполняют свои операции преобразования данных после получения команд и данных от процессора, работают параллельно с ним. Ввод и вывод данных через буферы шины

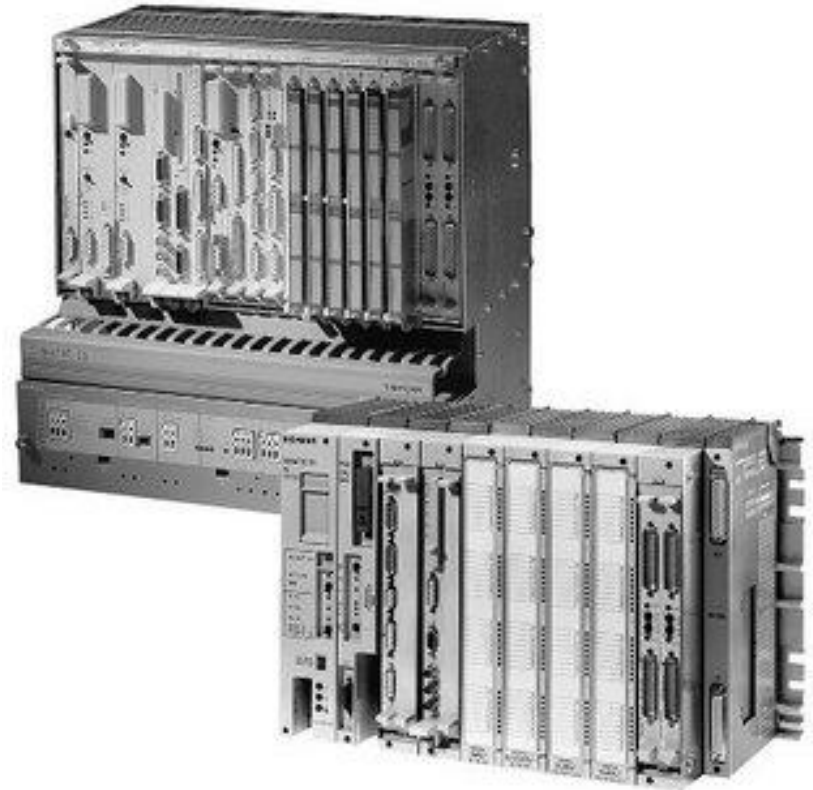
- Часть типовых операций управления, необходимых для связи с реальными объектами, реализованы в традиционной системе аппаратно в виде специализированных интерфейсных БИС. Эти операции не являются тривиальными, поскольку реализуются сложными самостоятельно функционирующими цифровыми автоматами, преобразующими данные. Такие автоматы, как последовательный порт, тактируются некоторым синхросигналом и проходят в процессе работы ряд состояний. Они самостоятельно воспринимают внешние события (сигналы) и обрабатывают их, привязка к сетке процессора производится по прерываниям. То же касается и формирования выходных сигналов управления. При отсутствии интерфейсной БИС с нужной операцией прежде приходилось создавать требуемый автомат из схем малой степени интеграции.

# Лекция7

Персональные компьютеры и  
программируемые логические  
контроллеры в системах  
управления

# История развития средств автоматизации

- С развитием средств автоматизации возникают новые требования к способам хранения, отображения и передачи информации. Современные ПТК имеют развитую сетевую инфраструктуру, которая дает возможность передавать большие объемы данных о состоянии объекта автоматизации между отдельными уровнями автоматизации.



- Первыми средствами автоматизации, которые исполняли логические операции, связанные с управлением были релейные схемы. Эти схемы использовали нормально замкнутые или нормально разомкнутые контакты реле, которые используя параллельное или последовательное подключение, образовывали логическую схему управления. С развитием средств автоматизации появилась необходимость управления не только дискретными сигналами, но и использовать аналоговые датчики и регуляторы, которые исполняли функции плавного регулирования, и поддержания режимных параметров. Недостатком таких средств автоматизации была сложность переконфигурирования, и такие средства автоматизации занимали достаточно много места в щитах автоматизации. Кроме того, такие средства автоматизации не давали возможности производить сложный математический анализ процессов, которые происходили в прошлом. Это связано с тем, что для хранения информации о ходе процесса использовались обычные регистраторы, которые не имели интерфейсов для цифровой передачи данных.

# ПК или ПЛК

- Персональные компьютеры класса PC всегда служили для построения высших уровней в иерархии систем управления машинами. Они использовались для сложных вычислений, мониторинга, измерений, создания коммуникационной инфраструктуры или в качестве интерфейсов оператора программируемых контроллеров.
- Компьютеры обычно устанавливались в местах, не требующих защиты от воздействий среды, потому что они не могли работать в таких сложных условиях, как программируемые контроллеры.



- Со временем персональные компьютеры получили возможность реализации управляющих функций в реальном времени, по-прежнему обладая высоким уровнем гибкости. Компьютеры PC со временем становились все более устойчивыми системами управления, являясь альтернативой для ПЛК, даже в тяжелых условиях среды. Интеграция функций ПЛК в технологии персональных компьютеров проявилась даже в аппаратно-программной архитектуре. На рынке появились карты, реализующие функции ПЛК (с разъемом PCI), контроллеры внешних устройств, которыми компьютеры уже могли управлять, а также интересные программные решения, предоставляющие компьютерам функции программируемых контроллеров. Самым важным изменением с этой точки зрения было создание ядра операционной системы реального времени. Это было необходимо, чтобы осуществить реализацию алгоритмов автоматической регуляции, в которых необходимо сохранять постоянное время дискретизации (сохранение детерминизма в вычислительных процессах в рамках алгоритма).

- С другой стороны, на рынке появились программируемые контроллеры со встроенными функциями, характерными для персональных компьютеров, поддерживающие мышь и компьютерную клавиатуру.
- Чем же сейчас следует руководствоваться, принимая решение о выборе типа системы управления – основанной на программируемых контроллерах или на компьютерной технологии?
- Безусловно, здесь поможет анализ, а затем сравнение нескольких ключевых факторов с точки зрения проектируемого приложения:
  - – метод выполнения разрабатываемой управляющей программы,
  - – степень устойчивости аппаратуры к тяжелым условиям среды,
  - – диапазон интеграции аппаратных компонентов,
  - – уровень интеграции связанных функций,
  - – метод программирования задач контроллера,
  - – стоимости всего инвестиционного проекта.

- Метод работы системы управления очень важен. А также то, каким образом выполняются отдельные задачи управления. Стандартные программируемые контроллеры имеют встроенные (с урезанными лишними функциями) операционные системы реального времени, а также специально созданные процессоры, благодаря чему обеспечивается высокий уровень надежности проектируемой системы управления. Программируемые контроллеры выполняют только задачи управления, поэтому для их задач не нужно запуска под операционной системой таких инструментов, как антивирусное программное обеспечение или обновление программных компонентов системы (известные по операционным системам общего назначения).

- Персональные компьютеры с запущенным ядром реального времени или операционной системой реального времени могут обеспечить тот же уровень надежности, что и программируемые контроллеры. Пользователи, имевшие опыт работы на домашних персональных компьютерах, опасались, что они будут зависать и прерывать работу (так называемый «синий экран смерти»). Следует учитывать, что зависание (в случае программируемых контроллеров проявляется переходом в сервисный режим) может произойти в любой операционной системе, также в программируемых контроллерах, если конфигурация приоритетов и времени выполнения задач управления не будут заданы правильно. В промышленности программное обеспечение, запускаемое на компьютерах PC, создано для применения в управлении производственными процессами, поэтому опасность «зависания» сведена к минимуму. Даже если это произойдет, ядро реального времени защищено, и будет правильно работать без перерыва.

- Выполнение задач управления в реальном времени означает, что их выполнение гарантировано через определенные промежутки времени, именуемое временем время
- Устойчивость к воздействиям среды
- Под устойчивостью контроллера понимается возможность его работы в тяжелых и часто меняющихся условиях. Типичные программируемые контроллеры не имеют движущихся частей, отсюда их долговечность в промышленных условиях. С другой стороны, стандартные персональные компьютеры имеют много движущихся частей, такие как вентиляторы или жесткие диски. Поэтому сложно ожидать от них долговечности, особенно там, где наблюдается высокий уровень механических колебаний. У промышленных компьютеров жесткие диски изготовлены по технологии SSD (Solid State Drive), имеют безвентиляторное охлаждение (например, с помощью радиаторов), могут также устанавливаться в специальных корпусах.

- Благодаря всему этому персональный компьютер становится более прочным, его устойчивость к внешним факторам приближается к программируемым контроллерам. К сожалению, обеспечение такого же уровня устойчивости к воздействию промышленной среды в случае компьютерных технологий приводит к удорожанию всей системы управления.
- Вопрос сервисного обслуживания

- С точки зрения пользователя очень важно, чтобы стоимость сервисного обслуживания системы управления была как можно ниже в течение всего «жизненного цикла». Обслуживание должно быть как можно проще и занимать мало времени. В случае программируемых контроллеров замена как внешних устройств, так и самих контроллеров происходит легко, минимизируя таким образом время простоя машины или участка производственной линии. В компьютерах все чаще применяется технология горячей замены (hot-swap), позволяющая менять компоненты системы управления без необходимости отключения питания. Ограничивают применение этой технологии внешние устройства, как правило подключенных к компьютеру с помощью интерфейса USB. Если аппаратная архитектура промышленного компьютера является модульной (наиболее частое решение), тогда время замены компонентов примерно такое же, как в случае программируемых контроллеров.

- Дополнительным пунктом является минимизация трудозатрат на выполнение сервисного обслуживания и запас нужного количества запасных частей по месту работы машины или производственной линии. Важным аспектом также является длительный временной горизонт доступности запасных частей.
- В некоторых отраслях промышленности необходимо использовать в сервисном обслуживании подход типа «сору-ехаст» (нельзя заменить компонент другим с подобными функциями – заменяемый компонент должен быть идентичным). В случае программируемых контроллеров это просто осуществить, поскольку ни оборудование, ни встроенное программное обеспечение (firmware) часто не меняются. С поиском идентичной запасной части в случае компьютеров (через год или два после покупки) могут возникнуть сложности.



# Интеграция аппаратных компонентов

- Сейчас инженеры обращают внимание на диапазон выбора опций в рамках видов аппаратных компонентов систем управления: памяти, периферийного оборудования, также интерфейса пользователя (оператора). Как программируемые контроллеры, так и компьютеры (в том числе промышленные), могут управлять множеством устройств с помощью промышленных коммуникационных сетей. К наиболее распространенным протоколам относятся: SERCOS, Profibus, DeviceNet и CANbus, а также их сетевые аналоги: SERCOS III, Profinet, Ethernet/IP, EtherCAT, Ethernet Powerlink. Как программируемые контроллеры, так и компьютеры предлагают различные возможности в области интерфейсов связи, однако при этом контроллеры в большинстве случаев имеют как минимум один встроенный интерфейс, а для компьютеров (в том числе промышленных) необходимы дополнительные карты, а также программируемые контроллеры для обслуживания сетей. Компьютеры, помимо интерфейсов связи, имеют другие возможности подключения внешних устройств, такие как USB, последовательный и параллельный порт, FireWire, беспроводное соединение TCP/IP. Благодаря этому пользователи могут решать задачи, обычно недоступные для типичных программируемых контроллеров. Примером может быть сложная обработка изображений с высоким разрешением, анализ, сравнение с эталонными изображениями, наконец, архивирование результатов расчетов. Компьютеры отлично подходят для этой цели, в основном в связи с требованием значительного объема памяти и вычислительной мощности процессора. Возможности программируемых контроллеров в этой области значительно ограничены.

- Интерфейс оператора является обязательным элементом большинства приложений для систем управления. Компьютеры имеют интерфейс исходя из принципа своей работы. Интерфейсы оператора для программируемых контроллеров могут состоять из: переключателей, панелей оператора, сигнализирующих элементов, таких как диоды, лампочки, звуковая сигнализация или даже панельных компьютеров. Из этого видно, что хотя программируемые контроллеры могут легко обмениваться данными с другими устройствами в промышленных сетях, им необходимы персональные компьютеры для выполнения задач, требующих значительного объема памяти, высокой вычислительной мощности и обмена данными с типичными периферийными устройствами.

# Безопасность приложений

- Ранее вопрос безопасности системы файлов, а тем самым приложения и данных процесса, хранящихся в системе управления, игнорировался, теперь же он все чаще поднимается. В рамках проблемы безопасности приложения для управления мы выделяем:
  - а) защиту от несанкционированного доступа (а тем самым потенциально опасного) извне приложения (например, вирусные атаки),
  - б) ограничение прав доступа к функции приложения для управления для разных пользователей (ограниченные возможности записи, чтения, отображения файлов в системе).
  - 
  -
- Традиционно (хотя сейчас, по правде говоря, в зависимости от данного производителя) программируемые контроллеры менее подвержены несанкционированному доступу извне. Благодаря использованию специальных версий операционных систем известно лишь несколько случаев вирусных атак на системы управления, построенные на программируемых контроллерах. К сожалению, это не означает, что программируемым контроллерам не угрожают вирусы.

- Отсутствует антивирусное программное обеспечение для программируемых контроллеров, которое бы их защищало. Хотя персональные компьютеры больше подвержены атакам вирусов, благодаря стандартизации в области операционных систем они помимо антивирусного программного обеспечения позволяют гибко настраивать как область, так и способ защиты.
- Программируемые контроллеры и персональные компьютеры, хотя и различными способами, позволяют задать уровни доступа к системным файлам и данным о процессе.
- Функциональная безопасность систем автоматики, а также вопросы программирования
- В зависимости от условий работы системы управления функциональная безопасность может являться ключевым аспектом выбора технологии. В случае машин имеет место постоянное взаимодействие пользователя и системы управления (что может породить потенциальную опасность).

- Программируемые контроллеры давно используются в автоматизации машин. Помимо обмена данными с исполнительными устройствами и/или датчиками (концевыми выключателями, световыми завесами безопасности) они обеспечивали функциональность контуров безопасности и избыточности.
- Интегрированная функциональная безопасность доступна на немногочисленных платформах систем управления, основанных на компьютерной технологии.
- Так повелось, что функциональные возможности устройств в рамках проектируемой системы управления зависят от программы, которая использует эти устройства для реализации задач управления. Обеспечение оптимального качества функционирования системы управления несомненно поддерживается программной средой, а также числом доступных языков программирования.

- Главным различием между программируемыми контроллерами и компьютерами РС является способ выполнения программы управления. В простых программируемых контроллерах сочетается постоянное выполнение задач (основанное на механизме так называемого сканирования) с выполнением задач в зависимости от происходящих событий в системе управления. В компьютерах РС задачи выполняются в основном в зависимости от возникающих событий. В случае механизма сканирования задачи не могут быть синхронизированы системно и должны ожидать выполнения, пока процессор контроллера завершит выполнение функций с более высоким приоритетом.

- Современные системы управления, как основанные на ПЛК, так и на РС, часто оснащаются операционными системами реального времени. Благодаря этому, помимо механизмов сканирования и выполнения заданий в зависимости от событий, в распоряжении пользователей имеется мощный инструмент – простая синхронизация множества выполняемых одновременно задач управления. Можно также задать приоритет реализуемых задач, в том числе собственных алгоритмов автоматического регулирования, а также распределения вычислений в случае мультипроцессорных систем или многоядерных процессоров.

- Различия в способе выполнения задач требуют разного подхода к созданию приложений управления. Пользователи часто ограничиваются только одним подходом к программированию. Программы создаются в разных языках, в зависимости от выбранной системной платформы. Для программируемых контроллеров действует стандарт IEC 61131-3, в которой указано несколько языков программирования (релейно-контактные схемы, список инструкций, структурированный текст, последовательностные функциональные диаграммы, функциональные блочные диаграммы). Производители предлагают пользователям упрощенные версии языков программирования, созданные для их решения.



- Для персональных компьютеров типичными языками программирования являются C/C++/.net.
- 
- Примечание редакции: Компьютерные системы управления, в которых функции ПЛК выполняет работающее под Windows программное обеспечение – ядро операционной системы реального времени – обычно программируются с использованием языков, характерных для программируемых контроллеров.

# Стоимость решения

- Выбор платформы системы управления должен зависеть от разрабатываемого приложения. В зависимости от того, что важнее пользователю, по-разному будет формироваться стоимость решения. Стоимость зависит в основном от следующих факторов:
  - – производительности (вычислительной мощности) системы управления,
  - – возможности расширения системы в процессе использования,
  - – условий среды, в которых будет работать система,
  - – планируемого времени внедрения приложения.

- Если система управления требует сложных вычислений, обслуживания крупной сети исполнительных устройств и/или датчиков, обработки большого объема данных, решения, основанные на компьютерах, будут более экономичными. Начальная стоимость компьютера в системах управления обычно выше, однако это решение значительно большую производительность, а стоимость растет только тогда, когда действительно будет не хватать (хотя часто это маловероятно) вычислительной мощности. Простые версии программируемых контроллеров намного дешевле компьютеров, но могут стать дороже компьютерных решений по мере роста производительности.

- Возможности расширения системы управления зависят от того, можно ли добавлять периферийное оборудование, увеличивать вычислительную мощность или реализовать функции хранения данных. Если не нужны дополнительные функциональные возможности, тогда необходимость расширения (например, в будущем) низкая. В этой ситуации решение с использованием компьютера будет дороже, чем программируемого контроллера, однако его стоимость по сравнению с ПЛК возрастет незначительно, если окажется, что расширение необходимо.
- Стоимость контроллера в связи с его оптимизацией обычно ниже в случае менее требовательных приложений, где возможность расширения не так важна, как прочие аспекты выбора. К сожалению, стоимость приложений резко растет по мере увеличения подсоединенных внешних устройств.

- Программируемые контроллеры с самого начала проектировались для работы в сложных промышленных условиях. В случае компьютеров их адаптация к работе в производственной среде обходится значительно дороже, чем в случае программируемых контроллеров.
- Время разработки управляющего программного обеспечения (программы, описывающей задачи управления) может существенно повлиять на стоимость всей системы. Стоимость управляющего приложения часто скрыта в цене всей системы, когда она запускается впервые. Интегрируя впервые систему управления на данной аппаратной платформе, пользователь может выбрать два пути: обучить своих сотрудников или нанять фирму, специализирующуюся в подобных решениях. Недостаток первого подхода – более длительное время внедрения приложения; недостаток второго – более высокая стоимость приложения.

- Многие инженеры знают языки программирования программируемых контроллеров, описанные в стандарте IEC 61131-3. В этих языках имеется множество встроенных функций, облегчающих реализацию систем управления. Тем самым программируемые контроллеры глобально могут быть более дешевым решением, чем системы управления, основанные на компьютерной технологии, особенно для начинающих и малоопытных пользователей. Время реализации и развитие приложения (а также его стоимость) уменьшается вместе с ростом опыта и знанием пользователей.
- В ситуации, когда приложение настолько большое, что требует обширных познаний и высокого уровня подготовки инженеров-программистов, стоимость решений, основанных на программируемых контроллерах и компьютерах класса PC, становится сопоставимой.

- В настоящее время на рынке появилось много инструментов, благодаря которым научные знания из области теории управления многомерными динамическими объектами можно просто и эффективно реализовать в промышленных системах управления. Несмотря на все еще высокую стоимость такого типа программных решений системы управления могут в соответствии с правилами так называемого быстрого прототипирования получить уникальные функции, существенно повышающие качество всей системы. В этом случае стоимость такого решения может быстро окупиться. Это особенно важно при производстве нового технологического оборудования, например станки с числовым управлением для обработки металлов резанием или машин для тепловой обработки.

# Лекция 8

РС или ПЛК: Сравнение опций  
управления



# РС или ПЛК: Сравнение опций управления

- Проблема выбора между программируемым логическим контроллером (ПЛК) и промышленным компьютером (РС) – одна из наиболее часто возникающих в наше время при модернизации существующих или внедрении новых систем управления в промышленности. Эти технологии обладают различными пользовательскими свойствами: принцип действия/обслуживания, устойчивость к промышленным условиям, легкость выполнения сервисного обслуживания, уровень интеграции оборудования и программного обеспечения, функциональная безопасность, обслуживание безопасных в управлении функций, метод программирования, наконец, стоимость внедрения или последующей эксплуатации.

- Одним из ключевых решений, принимаемых обычно на начальной стадии проектирования любого оборудования, является выбор типа системы управления. В настоящее время на рынке автоматике несомненно доминируют программируемые логические контроллеры (ПЛК). Они появились во второй половине семидесятых годов и получили признание инженеров, неоднократно подтверждая свою высокую позицию в области внедрения систем управления. Многие годы именно программируемые контроллеры являлись основой новых технологий управления машинами. В этот период, независимо от того, шла ли речь о простом управлении в автопромышленности или масштабных распределенных системах в управлении целыми заводами, именно программируемые контроллеры всегда были на первом месте. С начала девяностых персональные компьютеры класса PC с успехом внедрялись в системах управления. Это происходило в основном из-за растущей вычислительной мощности процессоров, а также резкого снижения цены на аппаратные компоненты.

- Количество решений для систем управления с использованием компьютерных технологий постоянно растет. Тем самым стирается некогда четкая граница между двумя решениями. Благодаря персональным компьютерам класса РС время внедрения приложения значительно сократилось. Еще десять лет назад выбор опции аппаратной платформы системы управления не вызывал затруднений, то сейчас он уже не так очевиден, в каждом случае нужен индивидуальный подход.

- Программируемые контроллеры первоначально являлись более удобной, гибкой и надежной альтернативой для шкафов управления с контакторами и реле (системами переключения). Контроллеры использовались для выполнения избранных задач управления, а язык их программирования (в основном релейно-контактные схемы), его структура, был разработан так, чтобы моделировать сеть переключающих схем, которую они должны были заменить. Кроме того, главным преимуществом программируемых контроллеров является их устойчивость к тяжелым условиям среды, электромагнитному излучению (EMI), пыли и вибрации.

- Со временем обычные программируемые контроллеры эволюционировали в направлении более гибких решений с точки зрения функциональности. Современные программируемые контроллеры имеют следующие возможности:
  - – реализация функции управления электроприводами,
  - – реализация функции цифровых регуляторов постоянных процессов (в том числе ПИД),
  - – интеграция обслуживания функциональной безопасности,
  - – передача данных с использованием множества протоколов, в том числе детерминистических,
  - – интеграция информационных технологий, таких как услуги серверов страниц WWW, серверов файлов FTP, или стандартизации, описанной в рамках спецификации серверов OPC.
-

- Любая машина, способная автоматически выполнять некоторые операции, имеет в своем составе *управляющий контроллер* - модуль, обеспечивающий логику работы устройства.
- Программируемый логический контроллер (далее ПЛК) (англ. Programmable Logic Controller, PLC) или просто программируемый контроллер - электронная составляющая промышленного контроллера, специализированного (компьютеризированного) устройства, используемого для автоматизации технологических процессов.
- Также ПЛК можно определить как - программно управляемый *дискретный автомат*, имеющий некоторое множество входов, подключенных посредством датчиков к объекту управления, и множество выходов, подключенных к исполнительным устройствам.

- Несомненно, появление первых ПЛК сделало прорыв в сфере автоматизации и послужило революционным скачком по степени внедрения информационных технологий на уровень автоматизации технологических процессов.
- На заре своего появления ПЛК имели только *бинарные входы*, т. е. входы, значения сигналов на которых способны принимать только два состояния — логического нуля и логической единицы. Так, наличие тока (или напряжения) в цепи входа считается обычно логической единицей. Отсутствие тока (напряжения) означает логический «0». Датчиками, формирующими такой сигнал, являются кнопки ручного управления, концевые датчики, датчики движения, контактные термометры и многие другие. *Бинарный выход* также имеет два состояния: включен и выключен. Сфера применения бинарных выходов: электромагнитные реле, силовые пускатели, электромагнитные клапаны, световые сигнализаторы и т. д.

- В современных ПЛК широко используются *аналоговые* входы и выходы. Аналоговый или *непрерывный* сигнал отражает уровень напряжения или тока, соответствующий некоторой физической величине в каждый момент времени. Этот уровень может относиться к температуре, давлению, весу, положению, скорости, частоте и т.д. В сфере применения ПЛК бинарные входы и выходы называют обычно *дискретными*.
- ПЛК предназначен для работы в режиме реального времени в условиях промышленной среды и должен быть доступен для программирования неспециалистом в области информатики. В системах реального времени помимо правильности решения определяющую роль играет *время реакции*. Логически верное решение, полученное с задержкой более допустимой, не является приемлемым.
- К негативным факторам, определяющим промышленную среду, относятся: температура и влажность, удары и вибрация, коррозионно-активная газовая среда, минеральная и металлическая пыль, электромагнитные помехи. Перечисленные факторы, весьма характерные для производственных условий, обуславливают жесткие требования, определяющие схемотехнические решения, элементную и конструктивную базу ПЛК. В процессе серийного производства ПЛК обязательным является технический прогон готовых изделий, включающий климатические, вибрационные и другие испытания.



## Физическое представление ПЛК.

- Физически, типичный ПЛК представляет собой блок, имеющий определенный набор выходов и входов, для подключения датчиков и исполнительных механизмов. Логика управления описывается программно на основе микрокомпьютерного ядра. Абсолютно одинаковые ПЛК могут выполнять совершенно разные функции. Причем для изменения алгоритма работы не требуется каких-либо переделок аппаратной части. Аппаратная реализация входов и выходов ПЛК ориентирована на сопряжение с унифицированными приборами и мало подвержена изменениям.
- Аппаратно ПЛК является вычислительной машиной. Поэтому архитектура его процессорного ядра практически не отличается от архитектуры компьютера. Отличия заключены в составе периферийного оборудования, отсутствуют видеоплата, средства ручного ввода и дисковая подсистема. Вместо них ПЛК имеет блоки входов и выходов (Рис. 1.1.).

# Программируемый логический контроллер Шнайдер Электрик Modicon M340.



- Для классификации огромного разнообразия существующих в настоящее время контроллеров рассмотрим их существенные различия.
- Основным показателем ПЛК является число каналов ввода-вывода. По этому признаку ПЛК делятся на следующие группы:
  - - нано-ПЛК (менее 16 каналов);
  - - микро-ПЛК (более 16, до 100 каналов);
  - - средние (более 100, до 500 каналов);
  - - большие (более 500 каналов).

# По расположению модулей ввода-вывода ПЛК бывают:

- - моноблочными, в которых устройство ввода-вывода не может быть удаленно из контроллера или заменено на другое. Конструктивно контроллер представляет собой единое целое с устройствами ввода-вывода (например, одноплатный контроллер). Моноблочный контроллер может иметь, например, 16 каналов дискретного ввода и 8 каналов релейного вывода;
- - модульные, состоящие из общей корзины (шасси), в которой располагаются модуль центрального процессора и сменные модули ввода-вывода. Состав модулей выбирается пользователем в зависимости от решаемой задачи. Типовое число слотов для сменных модулей — от 8 до 32;
- - распределенные (с удаленными модулями ввода-вывода), в которых модули ввода-вывода выполнены в отдельных корпусах, соединяются с модулем контроллера по сети (обычно на основе интерфейса RS-485) и могут быть расположены на расстоянии до 1,2 км от процессорного модуля.

- Часто перечисленные конструктивные типы контроллеров комбинируются, например моноблочный контроллер может иметь несколько съемных плат; моноблочный и модульный контроллеры могут быть дополнены удаленными модулями ввода-вывода, чтобы увеличить общее количество каналов.
- Многие контроллеры имеют набор сменных процессорных плат разной производительности. Это позволяет расширить круг потенциальных пользователей системы без изменения ее конструктива.
- По конструктивному исполнению и способу крепления контроллеры делятся на:
  - - панельные (для монтажа на панель или дверцу шкафа);
  - - для монтажа на DIN-рейку внутри шкафа;
  - - для крепления на стене;
  - - стоечные — для монтажа в стойке;
  - - бескорпусные (обычно одноплатные) для применения в специализированных конструктивах производителей оборудования (ОЕМ — Original Equipment Manufacturer).

- По области применения контроллеры делятся на следующие типы:
- - универсальные общепромышленные;
- - для управления роботами;
- - для управления позиционированием и перемещением;
- - коммуникационные;
- - ПИД- контроллеры;
- - специализированные.

- По способу программирования контроллеры бывают:
- - программируемые с лицевой панели контроллера;
- - программируемые переносным программатором;
- - программируемые с помощью дисплея, мыши и клавиатуры;
- - программируемые с помощью персонального компьютера.
- Контроллеры могут программироваться на языке МЭК 61131-3, используются также языки C, C#, Visual Basic.

- Контроллеры могут содержать в своем составе модули ввода-вывода или не содержать их. Примерами контроллеров без модулей ввода-вывода являются коммуникационные контроллеры, которые выполняют функцию межсетевых шлюзов, или контроллеры, получающие данные от контроллеров нижнего уровня иерархии АСУ ТП.
- Характерным для современных контроллеров является использование многопроцессорных решений. В этом случае модули ввода-вывода имеют собственные микропроцессоры, выполняющие необходимую предварительную обработку данных.
- Модуль центрального процессора имеет выделенную скоростную магистраль данных для работы с памятью и отдельную магистраль (сеть) для общения с модулями ввода-вывода. Еще одним вариантом построения ПЛК является *мезонинная технология*. Все силовые цепи, устройства защиты контроллера выполняются на несущей плате. Процессорное ядро контроллера, включающее систему исполнения, выполнено на отдельной сменной (мезонинной) плате. В результате появляется возможность составлять несколько комбинаций процессорного ядра и разных силовых плат без необходимости корректировки программного обеспечения. При необходимости процессор можно заменить даже в готовой системе.



- Типичный ПЛК представляет собой блок, имеющий определенный набор выходов и входов, для подключения датчиков и исполнительных механизмов. Логика управления описывается программно на основе микрокомпьютерного ядра. Абсолютно одинаковые ПЛК могут выполнять совершенно разные функции. Причем для изменения алгоритма работы не требуется каких-либо переделок аппаратной части. Аппаратная реализация входов и выходов ПЛК ориентирована на сопряжение с унифицированными приборами и мало подвержена изменениям.

# Режимы работы ПЛК.

- Разрабатывая системы управления, важной характеристикой, которую должен учитывать конструктор, это возможность системы работать в реальном времени.
- Режим реального времени это режим обработки данных, при котором обеспечивается взаимодействие вычислительной системы с внешними по отношению к ней процессами в темпе, соизмеримом со скоростью протекания этих процессов.
- Принято различать системы жесткого и мягкого реального времени. В системах *жесткого реального времени* существует выраженный временной порог. При его превышении наступают необратимые катастрофические последствия. В системах *мягкого реального времени* характеристики системы ухудшаются с увеличением времени управляющей реакции. Система может работать плохо или еще хуже, но ничего катастрофического при этом не происходит.
- Классический подход для задач жесткого реального времени требует построения *событийно управляемой системы*. Для каждого события в системе устанавливается четко определенное время реакции и определенный приоритет. Практическая реализация таких систем сложна и всегда требует тщательной проработки и моделирования.

- Для ПЛК существенное значение имеет не только быстродействие самой системы, но и время проектирования, внедрения и возможной оперативной переналадки. Абсолютное большинство ПЛК работают по методу *периодического опроса* входных данных (сканирования). ПЛК опрашивает входы, выполняет пользовательскую программу и устанавливает необходимые значения выходов.

- Специфика применения ПЛК обуславливает необходимость одновременного решения нескольких задач. Прикладная программа может быть реализована в виде множества логически независимых задач, которые должны работать одновременно. На самом деле ПЛК имеет обычно один процессор и выполняет несколько задач псевдопараллельно, последовательными порциями. Время реакции на событие оказывается зависящим от числа одновременно обрабатываемых событий. Рассчитать минимальное и максимальное значения времени реакции, можно, но добавление новых задач или увеличение объема программы приведет к увеличению времени реакции. Такая модель более подходит для систем мягкого реального времени. Современные ПЛК имеют типовое значение времени рабочего цикла, измеряемое единицами миллисекунд и менее. Поскольку время реакции большинства исполнительных устройств значительно выше, с реальными ограничениями возможности использования ПЛК по времени приходится сталкиваться редко.

- В некоторых случаях ограничением служит не время реакции на событие, а обязательность его фиксации, например работа с датчиками, формирующими импульсы малой длительности. Это ограничение преодолевается специальной конструкцией входов.
- Так, счетный вход позволяет фиксировать и подсчитывать импульсы с периодом во много раз меньшим времени рабочего цикла ПЛК. Специализированные интеллектуальные модули в составе ПЛК позволяют автономно обрабатывать заданные функции, например модули управления сервоприводом.

# *Интеграция ПЛК в систему управления предприятием.*

- Контроллеры традиционно работают в нижнем звене *автоматизированных систем управления предприятием (АСУ)* — систем, непосредственно связанных с технологией производства (ТП). ПЛК обычно являются первым шагом при построении систем АСУ. Это объясняется тем, что необходимость автоматизации отдельного механизма или установки всегда наиболее очевидна. Она дает быстрый экономический эффект, улучшает качество производства, позволяет избежать физически тяжелой и рутинной работы. Контроллеры по определению созданы именно для такой работы.

- Далеко не всегда удастся создать полностью автоматическую систему. Часто общее руководство со стороны квалифицированного человека — диспетчера необходимо. В отличие от автоматических систем управления такие системы называют *автоматизированными*. Еще 10 — 15 лет назад диспетчерский пульт управления представлял собой табло с множеством кнопок и световых индикаторов. В настоящее время подобные пульта применяются только в очень простых случаях, когда можно обойтись несколькими кнопками и индикаторами. В более «серьезных» системах применяются ПК. Появился целый класс программного обеспечения реализующего интерфейс *человек—машина* (ММІ). Это так называемые системы сбора данных и оперативного диспетчерского управления (SupervisoryControlAndDataAcquisitionSystem — SCADA). Современные SCADA-системы выполняются с обязательным применением средств мультимедиа. Помимо живого отображения процесса производства, хорошие диспетчерские системы позволяют накапливать полученные данные, проводят их хранение и анализ, определяют критические ситуации и производят оповещение персонала по каналам телефонной и радиосети, позволяют создавать сценарии управления, формируют данные для анализа экономических характеристик производства.

- Создание систем диспетчерского управления является отдельным видом бизнеса. Разделение производства ПЛК, средств программирования и диспетчерских систем привело к появлению *стандартных протоколов обмена данными*. Наибольшую известность получила технология OPC (OLE for Process Control), базирующаяся на механизме DCOM Microsoft Windows. Механизм динамического обмена данными (DDE) применяется пока еще достаточно широко, несмотря на то, что требованиям систем реального времени не удовлетворяет.



- Все это «многоэтажное» объяснение призвано подчеркнуть еще одно немаловажное преимущество ПЛК — средства системной интеграции являются составной частью базового программного обеспечения современного ПЛК.
- Второй часто возникающей задачей является интеграция нескольких ПЛК с целью синхронизации их работы. Здесь появляются сети, обладающие рядом специфических требований. В целом это требования, аналогичные требованиям к ПЛК: режим реального времени, надежность в условиях промышленной среды, ремонтпригодность, простота программирования. Такой класс сетей получил название *промышленных сетей* (fieldbus). Существует масса фирменных реализаций и достаточно много стандартов таких сетей (Bitbus, Modbus, Profibus, CANopen, DeviceNet), позволяющих интегрировать аппаратуру различных фирм, но ни один из них нельзя признать доминирующим. Благодаря продуктивному развитию средств сетевой интеграции появилась возможность создания *распределенных систем управления*. В 80-х гг. XX в. доминировали ПЛК с числом входов-выходов несколько сотен. В настоящее время большим спросом пользуются микро-ПЛК с количеством входов-выходов до 64. В распределенных системах каждый ПЛК решает локальную задачу. Задача синхронизации управления выполняется компьютерами среднего звена АСУ. Распределенные системы выигрывают по надежности, гибкости монтажа и простоте обслуживания.

# Лекция 9

Стандарты МЭК.  
Языки программирования  
ПЛК

# Стандарт МЭК.

- *Предпосылки создания.*
- При производстве совместимой продукции (подчиненной требованиям открытого стандарта) фирма производитель может сконцентрироваться на развитии наиболее удачных своих решений. Не опасаясь потерять заказчика, производитель может отказаться от невыгодных для себя изделий или частей работы. Кроме того, благодаря совместимости появляется возможность внедрять свои передовые изделия даже в полностью захваченных областях рынка. Тем самым расширяется и сам рынок.

# *Создание МЭК.*

- Международная электротехническая комиссия (МЭК, англ. International Electrotechnical Commission, IEC)— международная некоммерческая организация по стандартизации в области электрических, электронных и смежных технологий. Некоторые из стандартов МЭК разрабатываются совместно с Международной организацией по стандартизации (ISO).
- МЭК составлена из представителей национальных служб стандартов. МЭК была основана в 1906 году и в настоящее время в её состав входят более 76 стран.

- В 1979 году в рамках Международной Электротехнической Комиссии (МЭК) была создана специальная группа технических экспертов по проблемам ПЛК, включая аппаратные средства, монтаж, тестирование, документацию и связь. Первый вариант стандарта был опубликован в 1982 году. Ввиду сложности получившегося документа было решено разбить его на несколько частей. В настоящее время стандарт включает следующие части.
- Часть 1. Общая информация.
- Часть 2. Требования к оборудованию и тестам.
- Часть 3. Языки программирования.
- Часть 4, Руководства пользователя.
- Часть 5. Спецификация сообщений.
- Часть 6. Промышленные сети.
- Часть 7. Программирование с нечеткой логикой.
- Часть 8. Руководящие принципы применения и реализации языков ПЛК,
- Первоначально стандарт имел номер 1131, с 1997 года МЭК перешел на 5-цифровые обозначения. Теперь правильное наименование международной версии стандарта — МЭК 61131.

# *Простота программирования.*

- Инженер, спроектировавший машину, должен иметь возможность самостоятельно написать программу управления. Никто лучше его не знает, как должна работать данная машина.
- Многие решения, требовавшие раньше аппаратной поддержки, реализуются сегодня программно. Соответственно, требования к качеству программного обеспечения очень высоки. Поэтому сложную программу должны писать специалисты. Но для ответственных проектов очень важно, чтобы программа алгоритма была понятна техническому персоналу, осуществляющему настройку, сопровождение и ремонт оборудования. Они не обязаны изучать программу досконально, но понимать, что происходит, безусловно, должны.

- Очень часто технологи описывают процесс примерно так: «слегка перемешать, подогреть и довести до готовности». С аппаратными средствами здесь фактически все понятно, а вот с алгоритмом управления значительно сложнее. Для более детальных обсуждений технологии необходим некий общий язык, документальный и наглядный.
- Современные системы программирования контроллеров позволяют выполнить несколько распечаток программы с комментариями на разных языках -русском, немецком и т. д. Очевидно, это уже не маркетинговый ход разработчиков, а реальное требование современного бизнеса. Неэффективно реализованную программу можно заставить работать быстрее увеличением быстродействия процессора. Доходчивость представления достичь гораздо сложнее. Программу, в которой невозможно разобраться, придется рано или поздно выбросить.

- Внедрение стандарта дало фундамент для создания единой школы подготовки специалистов. Человек, прошедший обучение по программе, включающей стандарт МЭК 61131, сможет работать с ПЛК любой фирмы. В то же время, если он имел ранее опыт работы с любыми ПЛК, его навыки окажутся полезными и существенно упростят изучение новых возможностей.
- Программист не ограничен применением заданных в стандарте типов данных и операций. Стандарт допускает возможность создания пользовательских типов данных и функциональных блоков. Функции и функциональные блоки великолепно реализуют инкапсуляцию деталей реализации. Созданные пользователем библиотеки абсолютно равноправны стандартным. Новые оригинальные аппаратные решения изготовителей ПЛК могут быть поддержаны собственными библиотеками. Причем при создании внешних библиотек можно использовать любые инструменты, вплоть от ассемблера до C++.



# Языки МЭК.

- Стандарт МЭК 61131-3 устанавливает пять языков программирования ПЛК, три графических и два текстовых: LD, FBD, SFC, IL, ST. Рассмотрим каждый из них подробнее.
- *Язык IL.*
- Язык IL (InstructionList, Список Команд) представляет собой ассемблероподобный язык, достаточно несложный по замыслу авторов стандарта, для его практического применения в задачах промышленной автоматизации пользователем, не имеющим, с одной стороны, профессиональной подготовки в области программирования, с другой стороны, являющимся специалистом в той или иной области производства.
- Ввиду своей не наглядности, IL практически не используется для программирования комплексных алгоритмов автоматизированного управления, но часто применяется для кодирования отдельных функциональных блоков, из которых впоследствии складываются схемы FBD или SFC. При этом IL позволяет достичь высокой оптимальности кода: программные блоки, написанные на IL, имеют высокую скорость исполнения и наименее требовательны к ресурсам контроллера.

- Язык IL имеет все недостатки, которые присущи другим низкоуровневым языкам программирования: сложность и высокую трудоемкость программирования, трудность модификации написанных на нем программ, малую степень «видимого» соответствия исходного текста программы и решаемой задачи.

# Язык ST.

- Язык ST (Structured Text, Структурированный Текст) представляет собой язык высокого уровня, имеющий черты языков Pascal и Basic. Данный язык имеет те же недостатки, что и IL, однако они выражены в меньшей степени.
- С помощью ST можно легко реализовывать арифметические и логические операции (в том числе, побитовые), безусловные и условные переходы, циклические вычисления; возможно использование как библиотечных, так и пользовательских функций. Язык также интерпретирует более 16 типов данных.
- Язык ST может быть освоен технологом за короткий срок, однако текстовая форма представления программ служит сдерживающим фактором при разработке сложных систем, так как не дает наглядного представления ни о структуре программы, ни о происходящих в ней процессах.

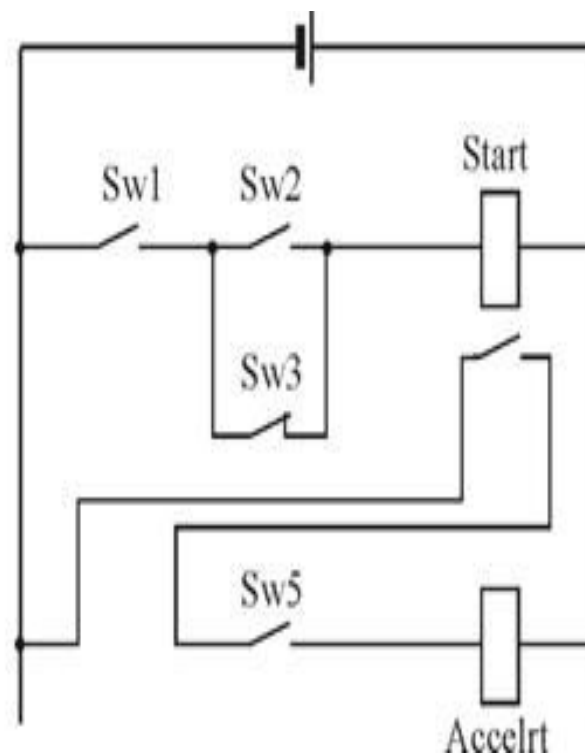
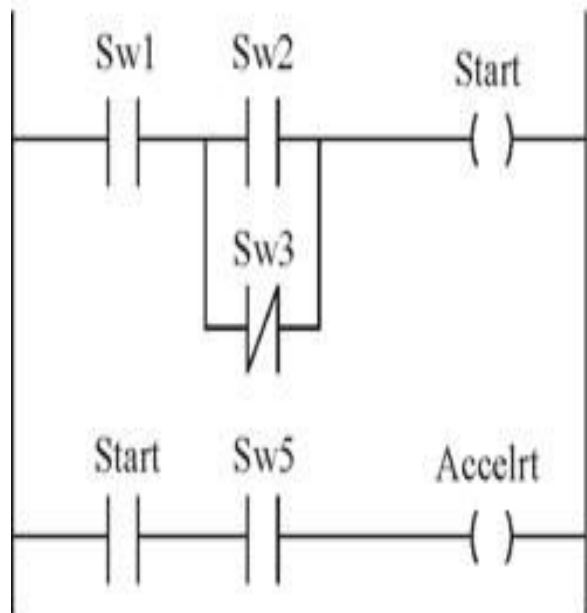
# *Пример листинга программы на языке ST.*

- IF Voltage>220 THEN
- Current:=Current - 10; (\*Если V>220 В, то уменьшить ток на 10\*)
- ELSE
- Current:=50; Speed:= ON;(\*Установить ток 50А и включить мотор\*)
- END\_IF;

# Язык LD.

- Графический язык релейной логики впервые появился в виде электрических схем, которые состояли из контактов и обмоток электромагнитных реле (рис. 2.1). Такие схемы использовались в автоматике конвейеров для сборки автомобилей до эры микропроцессоров. Язык релейной логики был интуитивно понятен людям, слегка знакомым с электротехникой и поэтому оказался наиболее распространенным в промышленной автоматике. Обслуживающий персонал легко находил отказ в оборудовании, прослеживая путь сигнала по релейной диаграмме.
- Однако язык LD проблематично использовать для реализации сложных алгоритмов, поскольку он не поддерживает подпрограммы, функции, инкапсуляцию и другие средства структурирования программ с целью повышения качества программирования. Эти недостатки затрудняют многократное использование программных компонентов, что делает программу длинной и сложной для обслуживания.

Пример программы на языке LD (слева) и ее эквивалент в виде электрической цепи с реле и выключателями (справа).



# Язык FBD.

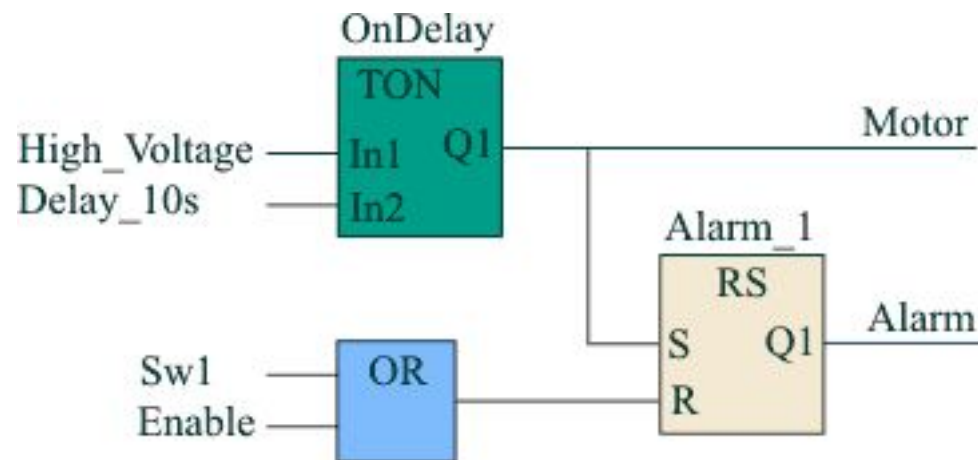
- FBD является графическим языком и наиболее удобен для программирования процессов прохождения сигналов через функциональные блоки. Язык FBD удобен для схемотехников, которые легко могут составить электрическую схему системы управления на "жесткой логике", но не имеют опыта программирования.
- Функциональные блоки представляют собой фрагменты программ, написанных на IL, SFC или других языках, которые могут быть многократно использованы в разных частях программы и которым соответствует графическое изображение, принятое при разработке функциональных схем электронных устройств ( рис. 2.2).
- Язык FBD может быть использован для программирования функций, функциональных блоков и программ, а также для описания шагов и переходов в языке SFC. Функциональные блоки инкапсулируют данные и методы, чем напоминают объектно-ориентированные языки программирования, но не поддерживают наследование и полиморфизм.

- Типичным применением языка FBD является описание "жесткой логики" и замкнутых контуров систем управления. Язык функциональных блоков является удобным также для создания и пополнения библиотеки типовых функциональных блоков, которую можно многократно использовать при программировании задач промышленной автоматизации. К типовым блокам относятся блок таймера, ПИД-регулятора, блок секвенсора, триггера, генератора импульсов, фильтра, и т. п.





# Пример программы на языке FBD.



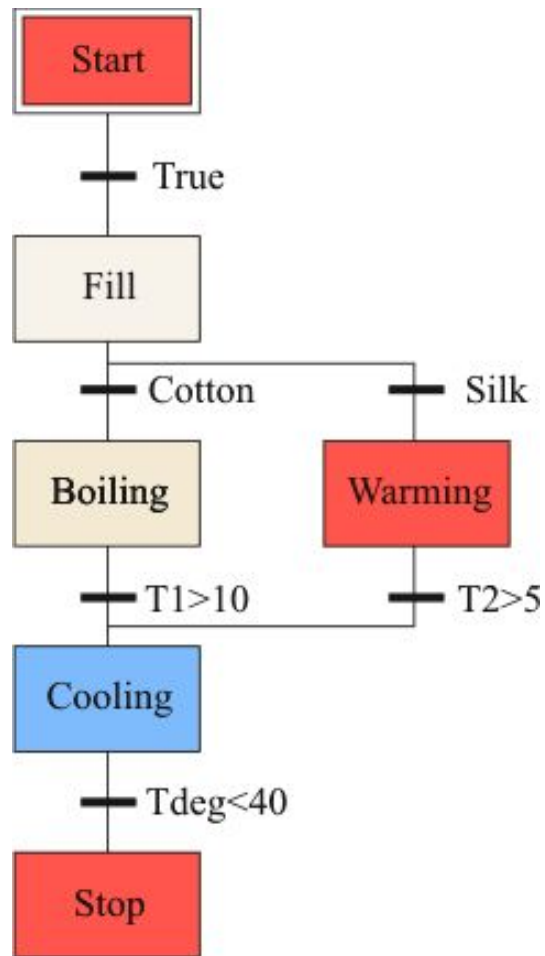
# Язык SFC.

- SFC называют языком программирования, хотя по сути это не язык, а вспомогательное средство для структурирования программ. Он предназначен специально для программирования последовательности выполнения действий системой управления, когда эти действия должны быть выполнены в заданные моменты времени или при наступлении некоторых событий. В его основе лежит представление системы управления с помощью понятий состояний и переходов между ними.

- Язык SFC предназначен для описания системы управления на самом верхнем уровне абстракции, например, в терминах "Старт", "Наполнение автоклава", "Выполнение этапа №1", "Выполнение этапа №2", "Выгрузка из автоклава". Язык SFC может быть использован также для программирования отдельных функциональных блоков, если алгоритм их работы естественным образом описывается с помощью понятий состояний и переходов. Например, алгоритм автоматического соединения модема с коммутируемой линией описывается состояниями "Включение", "Обнаружение тона", "Набор номер", "Идентификация сигнала" и переходами "Если длинный - то ждать 20 сек", "Если короткий - перейти в состояние "Набор Номера"" и т.д.

- На рис. 2.3 показан фрагмент программы на языке SFC. Программа состоит из шагов и условий переходов. Шаги показываются на схеме прямоугольниками, условия переходов - жирной перечеркивающей линией. Программа выполняется сверху вниз. Начальный шаг на схеме показывается в виде двойного прямоугольника. Условия переходов записываются рядом с их обозначениями. Каждый шаг программы может представлять собой реализацию сложного алгоритма, написанного на одном из МЭК-языков.

# Пример программы на языке SFC.

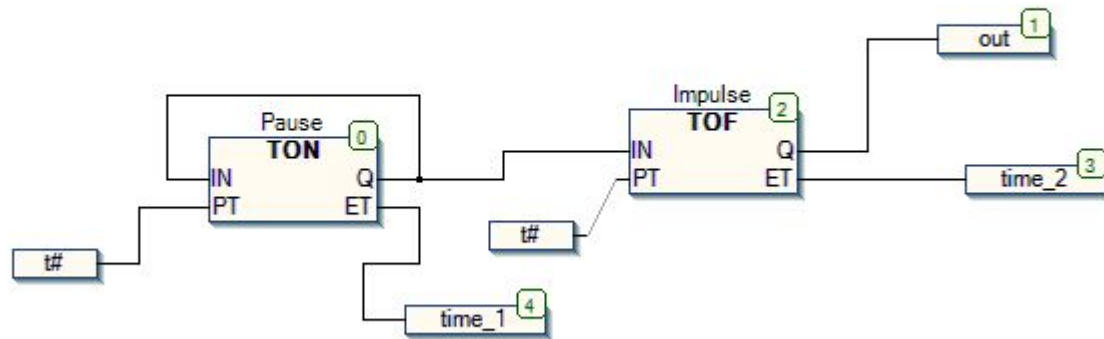


# Программное обеспечение для программирования ПЛК.

- Существует множество комплексов для разработки программного обеспечения. Фирмы, выпускающие промышленные контроллеры, стараются также предоставить свои среды для программирования. Наибольшей известностью в мире пользуются следующие комплексы.
- *CoDeSys*
- CoDeSys — инструментальный программный комплекс промышленной автоматизации. Производится и распространяется компанией 3S-Smart SoftwareSolutionsGmbH (Кемптен, Германия). Название CoDeSys является акронимом от ControllerDevelopmentSystem. Версия 1.0 была выпущена в 1994 году. С ноября 2012 изменено написание на CODESYS.
- Основой комплекса CODESYS является среда разработки прикладных программ для программируемых логических контроллеров (ПЛК). Она распространяется бесплатно и может быть без ограничений установлена на нескольких рабочих местах.

- В CODESYS для программирования доступны все пять определяемых стандартом МЭК 61131-3 языков:
- В дополнение к FBD поддержан язык программирования CFC (ContinuousFunctionChart) с произвольным размещением блоков и расстановкой порядка их выполнения.
- Пример реализации генератора импульсов на языке CFC на рис.2.4.

# Генератор импульсов.





- В CODESYS реализован ряд других расширений спецификации стандарта IEC 61131-3. Самым существенным из них является поддержка Объектно-ориентированного программирования (ООП).
- Встроенные компиляторы CODESYS генерируют машинный код (двоичный код), который загружается в контроллер. Поддержаны основные 16-и и 32-х разрядные процессоры: Infineon C166, TriCore, 80x86, ARM (архитектура), Power PC, SH, MIPS (архитектура), AnalogDevicesBlackfin, TI C2000/28x и другие.
- При подключении к контроллеру, среда программирования CODESYS переходит в режим отладки. В нем доступен мониторинг/изменение/фиксация значений переменных, точки останова, контроль потока выполнения, горячее обновление кода, графическая трассировка в реальном времени и другие отладочные инструменты.

- CODESYS версии V3 построен на базе так называемой платформы автоматизации: CODESYS AutomationPlatform. Она позволяет изготовителям оборудования развивать комплекс путем подключения собственных плагинов.
- Расширенная профессиональная версия среды разработки носит название CODESYS ProfessionalDeveloperEdition. Она включает поддержку UML диаграмм классов и состояний, подключение системы контроля версий Subversion, статический анализатор и профилировщик кода. Распространяется по лицензии.

- Инструмент CODESYS ApplicationComposer позволяет перейти от программирования практических приложений к их быстрому составлению. Пользователь составляет собственную базу объектов, соответствующих определенным приборам, механическим узлам машины и т.п. Каждый объект включает программную реализацию и визуальное представление. Законченное приложение составляется из необходимых объектов, конфигурируется и автоматически генерируется программа на языках МЭК 61131-3.

# *TwinCAT.*

- Пакет программного обеспечения Beckhoff TwinCAT превращает любой совместимый ПК в мульти задачную PLC- систему управления в режиме реального времени, с широкими возможностями ЧПУ. TwinCAT является единой программной средой для все контроллеров Beckhoff: от высокопроизводительных промышленных ПК, до встроенных ПЛК. TwinCAT заменяет привычные ПЛК и ЧПУ контроллеры предоставляя:
  - - открытое, ПК совместимое аппаратное обеспечение
  - - интеграция программного обеспечения ПЛК (IEC 61131-3) и ЧПУ в среду Windows NT/2000/XP NT/XP Embedded, CE
  - - среды программирование и runtime могут устанавливаться на одном ПК или отдельно
  - - сопряжение со всеми наиболее популярными промышленными шинами и интерфейсами ПК
  - - сопряжение данных с пользовательскими интерфейсами и другими программами с использованием открытых стандартов Microsoft (OPC, ODX, DLL и т.д.)

- TwinCAT состоит из run-time системы управления в режиме реального времени, среды программирования, диагностики и конфигурирования системы. Все приложения Windows, например, программы визуализации пакета Office, могут обращаться к данным TwinCAT с помощью интерфейсов Microsoft или выдавать команды.
- Созданное как целостное программное решение для ПЛК, TwinCAT PLC позволяет создавать на одном ПК до четырех виртуальных «PLC процессора», каждый из которых может обслуживать до четырех пользовательских задач. Программы для ПЛК создаются на одном или нескольких языках стандарта IEC 61131-3:

- IL (Язык линейных инструкций),
- LD (Релейные диаграммы),
- FBD/CFC (Функциональные блочные диаграммы),
- SFC (Последовательные функциональные схемы)
- ST (Структурированный текст).
- TwinCAT PLC работает под ОС Windows NT/2000/XP, включая в себя среду программной разработки и систему выполнения задач, не требуя дополнительных программных инструментов. Для операционных систем Windows CE и встроенных операционных систем программируемых контроллеров серий VX и VC доступен только TwinCAT PLC в режиме run-time. Изменение программ производится посредством взаимодействия через шинную коммуникацию с системой выполнения программ.

- Программирование осуществляется:
- - локально
- - через TCP/IP
- - через промышленные шины (VХxxxx и ВСxxxx)..
- NC (numerical control) - это числовое программное управление (ЧПУ), ПО для ЧПУ состоит из:
- - задач позиционирования (установка вычисленных значений и контроль положения)
- -интегрирования ПЛК с ЧПУ интерфейсом
- - программы проверки правильности функционирования
- - управления осями через промышленные шины.

- TwinCAT NC I (интерполяция) - ЧПУ система для линейно-круговой интерполированной траектории перемещения группы осей, каждая из которых содержит два или три привода.
- Система включает интерпретатор, программу формирования заданных значений и контроллер позиционирования. Выполняемые функции ПЛК интегрированы в систему для обеспечения связи осей с промышленной шиной.
- TwinCAT CNC - ПО компьютерного ЧПУ (CNC) для решения сложнейших задач.
- TwinCAT CNC предлагает все функции CNC в виде программного решения на базе ПК, охватывает все области классического управления траекторией CNC, вплоть до систем High-end, отвечающих комплексным требованиям к движению и кинематике. Это решение для классического CNC приложения с поддержкой до 32 интерполированных движущихся осей и управления шпинделями в одном CNC канале или распределенных вплоть до 10 каналов. В сочетании со всесторонней и мощной CNC функциональностью и возможной высокоскоростной пятиосевой машинной обработкой, TwinCAT CNC может полностью обеспечить любое CNC приложение в различных технологиях обработки на станках. В отличие от классических линейной, круговой и спиральной интерполяций, TwinCAT CNC имеет дополнительно удобную сплайн интерполяцию, основанную по выбору на би-сплайн интерполяции или на методе Акимы, для уменьшения числа NC блоков.



# Интерфейс TwinCat System Manager и PLC Control.

