

**ГУСС СВЯТОСЛАВ ВЛАДИМИРОВИЧ**

# **МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ**

**ВВЕДЕНИЕ В ДИСЦИПЛИНУ**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

ОМСК – 2013



С.В. ГУСС

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

ВВЕДЕНИЕ В ДИСЦИПЛИНУ

Учебно-методическое пособие для студентов, обучающихся по специальностям

«Вычислительные машины, комплексы, системы и сети»,

«Информатика и вычислительная техника».

УДК 004.318

Гусс С.В. Микропроцессорные системы. Введение в дисциплину.

В учебном пособии представлен теоретический материал по курсу «Микропроцессорные системы» для студентов, обучающихся по специальностям 230101.65 «Вычислительные машины, комплексы, системы и сети», 230100.62 «Информатика и вычислительная техника».

## СОДЕРЖАНИЕ

<b>ТЕМА 1. ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ .....</b>	<b>8</b>
<b>Лекция 1 .....</b>	<b>8</b>
<b>1.1. Термины и определения .....</b>	<b>8</b>
Ключевые определения .....	8
Электронные цифровые системы.....	10
Особенности традиционных цифровых систем .....	12
<b>1.2. Микропроцессор и микропроцессорные системы .....</b>	<b>12</b>
Микропроцессор .....	12
Управление работой микропроцессора .....	13
Структура микропроцессора .....	14
Требования к разработчику систем .....	15
<b>Вопросы .....</b>	<b>15</b>
<b>Лекция 2 .....</b>	<b>17</b>
<b>1.3. Коммуникация элементов микропроцессорной системы.....</b>	<b>17</b>
Классическая структура связей.....	17
Шинная структура связей.....	17
Структура микропроцессорной системы.....	18
Обмен информацией.....	19
Виды обмена информацией.....	20
<b>1.4. Разновидности микропроцессорных систем .....</b>	<b>21</b>
Архитектура с общей шиной для команд и данных .....	21
Архитектура с отдельными шинами команд и данных.....	22
Типы микропроцессорных систем .....	22
<b>Вопросы .....</b>	<b>24</b>
<b>ТЕМА 2. ЦИКЛЫ ОБМЕНА ИНФОРМАЦИЕЙ .....</b>	<b>25</b>
<b>Лекция 3 .....</b>	<b>25</b>
<b>2.1. Циклы обмена информацией .....</b>	<b>25</b>

Циклы обмена .....	25
Шины магистрали .....	26
Обмен данными .....	27
<b>2.2. Циклы программного обмена .....</b>	<b>28</b>
Обмен по мультиплексированной асинхронной магистрали .....	28
Обмен по синхронной не мультиплексированной магистрали .....	31
<b>Вопросы .....</b>	<b>33</b>
<b>Лекция 4 .....</b>	<b>34</b>
<b>2.3. Циклы обмена по прерываниям.....</b>	<b>34</b>
Типы прерываний .....	34
Обмен по векторным прерываниям .....	34
Обмен по радиальным прерываниям .....	36
<b>2.4. Циклы обмена в режиме прямого доступа к памяти .....</b>	<b>36</b>
Обмен в режиме прямого доступа к памяти.....	36
<b>Вопросы .....</b>	<b>39</b>
<b>Промежуточный контроль .....</b>	<b>39</b>
<b>ТЕМА 3. ФУНКЦИИ УСТРОЙСТВ МАГИСТРАЛИ .....</b>	<b>40</b>
<b>Лекция 5 .....</b>	<b>40</b>
<b>3.1. Устройство процессора .....</b>	<b>40</b>
Процессор.....	40
Схема управления выборкой команд.....	41
Арифметико-логическое устройство .....	41
Регистры .....	41
Управление.....	42
<b>3.2. Память.....</b>	<b>43</b>
Устройство памяти .....	43
<b>Вопросы .....</b>	<b>48</b>

<b>ТЕМА 4. АДРЕСАЦИЯ ОПЕРАНДОВ И РЕГИСТРЫ ПРОЦЕССОРА .....</b>	<b>49</b>
<b>Лекция 6 (часть 1) .....</b>	<b>49</b>
<b>4.1. Адресация операндов .....</b>	<b>49</b>
Адресация.....	49
Методы адресации .....	50
<b>4.2. Сегментирование памяти .....</b>	<b>51</b>
Сегментирование в Intel 8086.....	51
Метод сегментирования в Intel 80286.....	53
Адресация памяти и сегментирование в Intel 80386.....	53
<b>4.3. Регистры процессора.....</b>	<b>54</b>
Регистры процессора Intel 8086 .....	54
<b>Вопросы .....</b>	<b>55</b>
<b>ТЕМА 5. СИСТЕМА КОМАНД ПРОЦЕССОРА .....</b>	<b>56</b>
<b>Лекция 6 (часть 2) .....</b>	<b>56</b>
<b>5.1. Команды процессора .....</b>	<b>56</b>
Команды пересылки данных .....	56
Арифметические команды .....	56
Логические команды.....	57
Команды переходов .....	57
<b>Вопросы .....</b>	<b>58</b>
<b>ТЕМА 6. ОРГАНИЗАЦИЯ МИКРОКОНТРОЛЛЕРОВ.....</b>	<b>59</b>
<b>Лекция 7 .....</b>	<b>59</b>
<b>6.1. Устройство микроконтроллера .....</b>	<b>59</b>
Классы микроконтроллеров .....	59
Модульная организация микроконтроллера .....	59
<b>6.2. Система команд микроконтроллера.....</b>	<b>61</b>
Система команд .....	61
Память программ .....	62

Память данных .....	62
<b>6.3. Дополнительные вопросы организации микроконтроллеров.....</b>	<b>63</b>
Регистры микроконтроллера .....	63
Стек микроконтроллера.....	63
Внешняя память .....	63
Порты ввода/вывода .....	64
Модуль прерываний микроконтроллера .....	65
Режимы работы микроконтроллера .....	65
<b>Вопросы .....</b>	<b>65</b>
<b>Лекция 8 .....</b>	<b>67</b>
<b>Промежуточный контроль .....</b>	<b>67</b>
<b>ТЕМА 7. ПРОЕКТИРОВАНИЕ УСТРОЙСТВ.....</b>	<b>69</b>
<b>Лекция 9 .....</b>	<b>69</b>
<b>7.1. Задачи проектирования микропроцессорной системы.....</b>	<b>70</b>
Технико-экономический анализ и обоснование.....	70
Использование микропроцессорной техники .....	70
<b>7.2. Классификация микропроцессорных устройств .....</b>	<b>72</b>
<b>Вопросы .....</b>	<b>76</b>
<b>Лекция 10 .....</b>	<b>77</b>
<b>7.3. Этапы разработки и модернизации .....</b>	<b>77</b>
Системный анализ .....	77
Выбор типов используемых средств.....	80
<b>Вопросы .....</b>	<b>85</b>
<b>Лекция 11 .....</b>	<b>85</b>
Разработка аппаратных средств .....	89
Разработка программного обеспечения .....	89
Совместная отладка .....	92
<b>Вопросы .....</b>	<b>92</b>

<b>ВОПРОСЫ ДЛЯ ПРОВЕРКИ ПО ДИСЦИПЛИНЕ .....</b>	<b>93</b>
<b>СПИСОК ЛИТЕРАТУРЫ .....</b>	<b>96</b>



# ТЕМА 1. ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

## Лекция 1

### 1.1. Термины и определения

#### Ключевые определения

Определение 1. *Электронная система* (в рамках изучаемой дисциплины «Микропроцессорные системы») – система, производящая обработку информации. Такая система может быть представлена электронным *узлом* (как часть устройства), *блоком* (в виде совокупности взаимосвязанных элементов и узлов устройства, выполняющих определённую функцию), *прибором* (как самостоятельное устройство) или *комплексом* (как совокупность устройств, образующих единое целое).

Определение 2. *Задача* – требование к электронной системе в виде набора функций (внешняя сторона); основная единица работы системы, требующая выделения ресурсов (внутренняя сторона).

Определение 3. *Быстродействие* – показатель скорости выполнения электронной системой своей задачи (внешняя сторона); количество элементарных операций (сложение, сдвиг, ...), выполняемых в единицу времени (внутренняя сторона).

*Номинальное быстродействие* – предельное быстродействие, соответствующее определённому режиму использования устройства.

Определение 4. *Гибкость* – способность электронной системы подстраиваться под определённые задачи (в этом случае говорят об *эксплуатационной гибкости*).

В традиционном понимании, *гибкость системы* – свойство системы, состоящее в возможности её совершенствования, расширения и придания ей новых качеств.

Определение 5. *Избыточность* – степень соответствия возможностей электронной системы решаемой задаче. Когда говорят об избыточности, подразумевают дополнительные средства (программные или аппаратные), вводимые в систему для повышения её надежности и защищённости.

*Алгоритмическая избыточность* – дополнительные средства, включенные в алгоритм решения задачи, удаление которых не влияет на результат. Такая избыточность используется для повышения достоверности результатов.

*Функциональная избыточность* – дополнительные функции и компоненты системы, наличие которых не является принципиально необходимым для реализации возлагаемых на неё задач.

Предположим, что у нас есть электронная система, которая способна предоставить определённый функционал своему пользователю (табл. 1.1). Назовём функции системы – возможностями (№1, №2, №3).

Пусть возможность №1 будет реализована с низким уровнем качества, возможность №2 будет реализована с высоким уровнем качества, а возможность №3 со средним уровнем качества. Также допустим, что нашей системе необходимо решать задачи, в которых требуется, чтобы все возможности системы были реализованы со средним уровнем качества. В таком случае, наша система не будет обладать достаточным уровнем избыточности, поскольку одна из возможностей реализована на низком уровне. С другой стороны, если задача требует низкого уровня реализации возможностей системы, тогда наша система будет избыточной и в случае необходимости она будет способна выполнять задачи, требующие высокого уровня качества возможности №2 или, например, среднего уровня возможности №3.

Таблица 1.1

## Избыточность электронной системы (иллюстрация к примеру)

Уровень качества Функции	Низкий	Средний	Высокий
Возможность №1	+		
Возможность №2			+
Возможность №3		+	

Определение 6. *Интерфейс* – правила обмена информацией между устройствами, участвующими в обмене.

*Внешний интерфейс* – средства и правила взаимодействия подсистемы с внешними объектами (пользователем, вычислительной сетью).

*Внутренний интерфейс* – интерфейс с внутренним компонентом системы (или между внутренними компонентами системы).

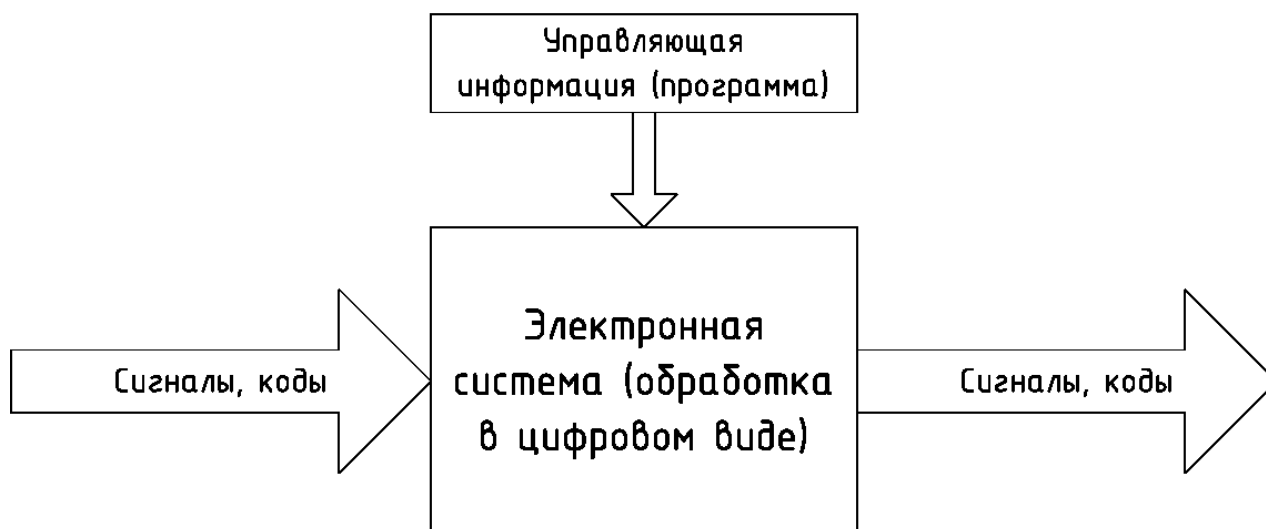
Определение 7. *Микропроцессорная система* – частный случай цифровой электронной системы, предназначенной для обработки входных сигналов и выдачи выходных, это вычислительная или управляющая система, в структуру которой входит *микропроцессор*.

### Электронные цифровые системы

На рис. 1.1 представлена *схема электронной микропроцессорной системы*, осуществляющей обработку информации.

На вход системы могут поступать *аналоговые сигналы, цифровые сигналы*, а также *цифровые коды* в виде набора сигналов. На выходе системы могут быть обнаружены сигналы той же природы, только обработанные в соответствии с

алгоритмом программы, выступающей в качестве *управляющей информации* для представленной электронной системы.



**Рисунок 1.1.** Электронная микропроцессорная система

В случаях, когда система имеет дело с аналоговыми сигналами, используются специальные устройства, такие как *аналогово-цифровой преобразователь* (АЦП) и *цифро-аналоговый преобразователь* (ЦАП) на входе и выходе электронной системы соответственно.

Определение 8. **Традиционная цифровая система** – электронная цифровая система, алгоритмы обработки и хранения информации в которой «жестко» связаны со *схемотехникой* системы.

---

Примечание:

*В данном случае если, например, нужна операция суммирования, необходимо добавить в систему такой элемент, как сумматор (на аппаратном уровне). А это в свою очередь приведёт к новому производственному циклу, включающему проектирование, изготовление и отладку (подробно о процессе производства будет говориться в теме 7).*

---

Традиционная электронная система по своей природе является *специализированной* системой.

## Особенности традиционных цифровых систем

### Преимущества специализированной системы:

1. **Отсутствие аппаратной избыточности.** Это ведёт к тому, что система может работать в полную мощность.
2. **Максимально высокое быстродействие.** Быстродействие зависит от состава системы, т.е. от её логических элементов и схемы путей прохождения информации (в специализированной системе схемы путей обычно оптимизированы).

### Недостатки специализированной системы:

1. **Отсутствие гибкости.** Переход с одного алгоритма на другой ведёт к изменению аппаратуры.

«Частность» микропроцессорной системы заключается в возможности изменения алгоритма работы, не изменяя аппаратной части. Средством универсальности является управляющая программа. Возможность смены управляющей программы обеспечивается аппаратной избыточностью. Аппаратная избыточность приводит к увеличению стоимости и потребляемой мощности, а также снижению надёжности и быстродействия.

Тенденция сегодняшнего дня – вытеснение традиционных систем на жёсткой логике, вызванное развитием программируемых (микропроцессорных) систем.

## 1.2. Микропроцессор и микропроцессорные системы

### Микропроцессор

Определение 9. **Микропроцессор** – ключевой элемент микропроцессорной системы. Заменяет «жёсткую» логику. Это большая *интегральная схема* (БИС) (содержащая сотни, тысячи, а то и миллионы элементов и компонентов в одном кристалле полупроводника), выполняющая функции *центрального процессора* (т.е. устройства, предназначенного для интерпретации программ,

выполняющего основные функции по обработке данных и управлению работой частей системы).

*Встроенный микропроцессор* – микропроцессор, встроенный в устройство и используемый в системе управления этим устройством.

*Коммуникационный микропроцессор* – специализированная система, выполняющая функции управления сетевыми протоколами в сетевой телеобработке. Обеспечивает повышение производительности сетей.

*Секционный микропроцессор* – микропроцессор, полученный на основе соединения одноступенчатых 2-, 4- и 8-разрядных микропроцессорных интегральных схем, каждая из которых имеет арифметико-логическое устройство и несколько регистров.

Работа микропроцессора заключается в выполнении:

1. Арифметических функций.
2. Логических функций.
3. Временного хранения кодов.
4. Пересылке кодов между узлами системы.

#### **Управление работой микропроцессора**

Определение 10. Программа – набор цифровых кодов, расшифровав которые, процессор узнаёт, что ему надо делать. Программа представляет собой данные, предназначенные для управления конкретными компонентами системы обработки данных в целях реализации определённого алгоритма.

Определение 11. Система команд – набор команд, выполняемых процессором. Это полный набор команд, допустимых в машинном языке данного типа вычислительного устройства.

### Характеристики системы команд:

1. **Специализация**, выражается кругом решаемых задач. Может быть *узкой* или *широкой*.
2. **Преимущество**. Выражается *быстродействием*, *гибкостью* и *удобством использования*.
3. **Размер**. Выражается *количеством* занимаемых *разрядов* ячеек памяти.
4. **Время выполнения** отдельных команд. Как правило, выражается в *единицах времени*.

Таким образом, ведя разговор о системе команд конкретного микропроцессора, можно обсуждать ее *специализацию*, т.е. на что она оптимизирована лучше всего (например, на обработку сигналов, передающих звуковую или визуальную информацию, в так называемых *процессорах цифровой обработки сигналов*).

Говоря о *преимуществах* системы команд, весьма кстати будет вопрос об удобстве её использования, поскольку это напрямую влияет на производительность труда разработчиков программной части микропроцессорной системы.

Такие характеристики, как *размер* системы команд (имеется в виду размер памяти – количество ячеек, занимаемых каждой командой) и *время выполнения* отдельных команд, особый интерес будут представлять для проектировщиков сложных встраиваемых систем реального времени.

### **Структура микропроцессора**

На рис. 1.2 представлена *обобщённая структура микропроцессора*.

#### Структурные составляющие процессора:

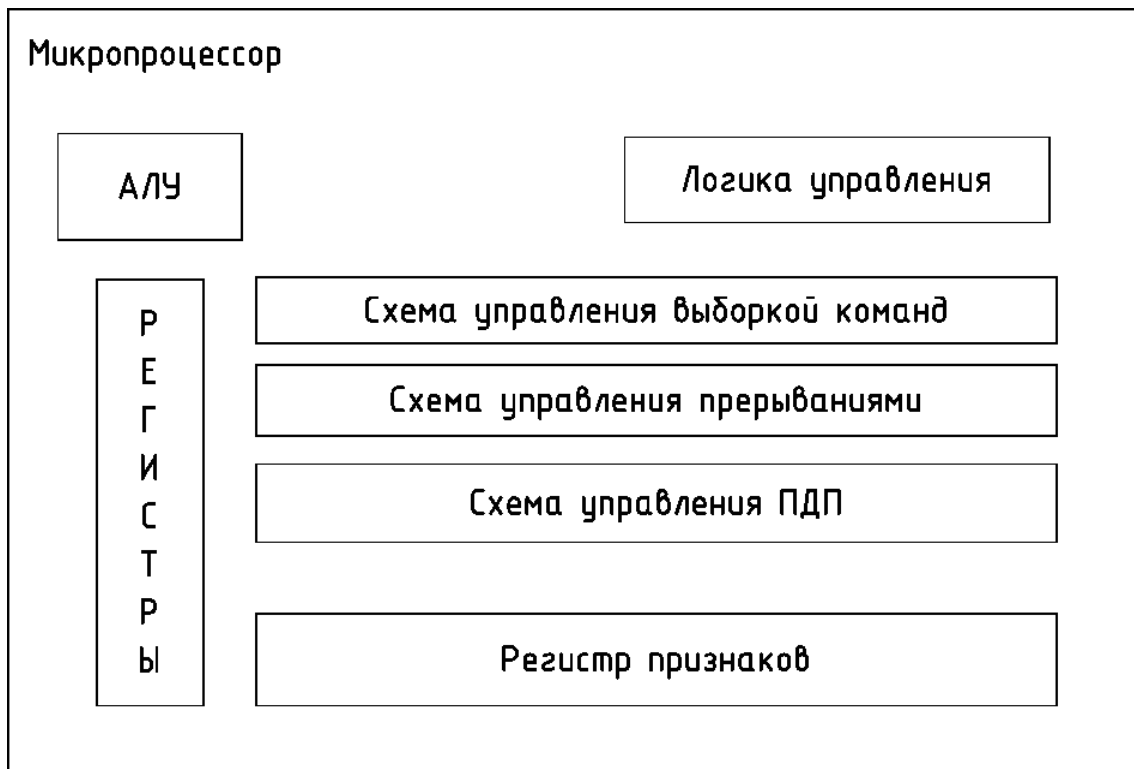
1. Арифметико-логическое устройство.
2. Мультиплексоры.
3. Буферы.

#### 4. Регистры.

#### Требования к разработчику систем

Требования к разработчику микропроцессорных систем:

1. **Знание процессора:** система команд, доступные режимы работы, правила обмена информацией с внешними устройствами (его интерфейс).



**Рисунок 1.2.** Структура процессора

#### Вопросы

1. Что такое электронная система?
2. Как (в какой форме или в каком виде) может быть представлена электронная система?
3. Какие отличия (или связи) между узлом, блоком, прибором и комплексом?
4. Дайте определение понятию «задача».
5. Что такое быстродействие системы?
6. Что такое номинальное быстродействие электронной системы?
7. Дайте определения понятиям «гибкость» и «эксплуатационная гибкость».



8. Какие виды избыточности систем вы знаете?
9. Какая разница между внутренним и внешним интерфейсом?
10. Что является неотъемлемой частью микропроцессорной системы?
11. Чем отличается цифровой сигнал от цифрового кода?
12. Зачем нужны аналого-цифровой и цифро-аналоговый преобразователи?
13. Что понимают под традиционной цифровой системой?
14. Как правильно понять фразу «алгоритм обработки информации жёстко связан со схмотехникой системы»? Можете ли вы привести пример такой связи (или устройства)?
15. Что такое специализированная система? В чём её преимущества и недостатки?
16. Что является средством универсальности микропроцессорной системы?
17. Что такое микропроцессор? Зачем он нужен?
18. Что такое большая интегральная схема? Почему она называется большой (несмотря на то, что её размер может быть с монету)?
19. Что такое центральный процессор?
20. Дайте определения понятиям «встроенный микропроцессор», «коммуникационный микропроцессор», «секционный микропроцессор».
21. В чём заключается работа микропроцессора?
22. Что такое программа? В чём разница между алгоритмом, программой и исходным кодом программы?
23. Что такое система команд? Как можно охарактеризовать систему команд. Приведите примеры применения характеристик (например, в рамках дискуссии между инженерами-проектировщиками микропроцессорных систем).
24. Какие вы можете выделить структурные составляющие микропроцессора?
25. Какие требования предъявляют к разработчику микропроцессорных систем? Как вы думаете, этих требований достаточно?

## Лекция 2

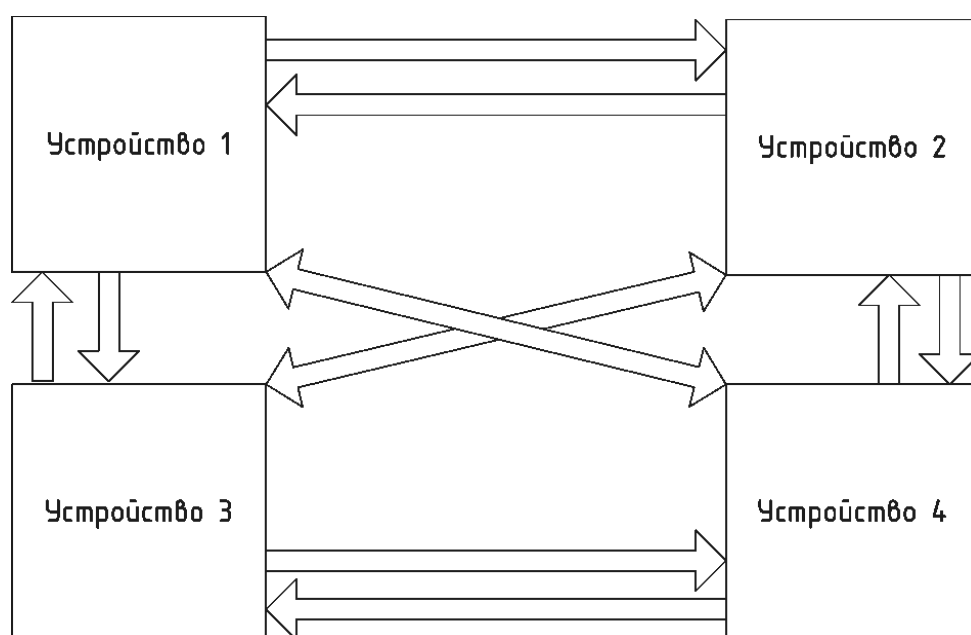
### 1.3. Коммуникация элементов микропроцессорной системы

#### Классическая структура связей

Характеристики классической структуры связей:

1. Множество линий связи.
2. Различные правила обмена информацией.

На рис. 1.3 представлена *классическая структура связей* между устройствами.



**Рисунок 1.3.** Классическая структура связей

#### Шинная структура связей

Характеристики шинной структуры связей:

1. Линии связи (у устройств) одни, время связи разное. Такая передача носит название *мультиплексированной передачи*. Результат – снижение быстродействия.
2. *Двунаправленная передача*.

3. **Одни протоколы обмена информацией.** Таким образом, обеспечивается унификация шинных устройств (т.е. устройств, соединённых единой магистралью).

Недостатки:

1. **Пониженная надёжность.** Неисправность одного устройства может привести к выходу из строя всей системы.

На рис. 1.4 представлена *шинная структура связей*.

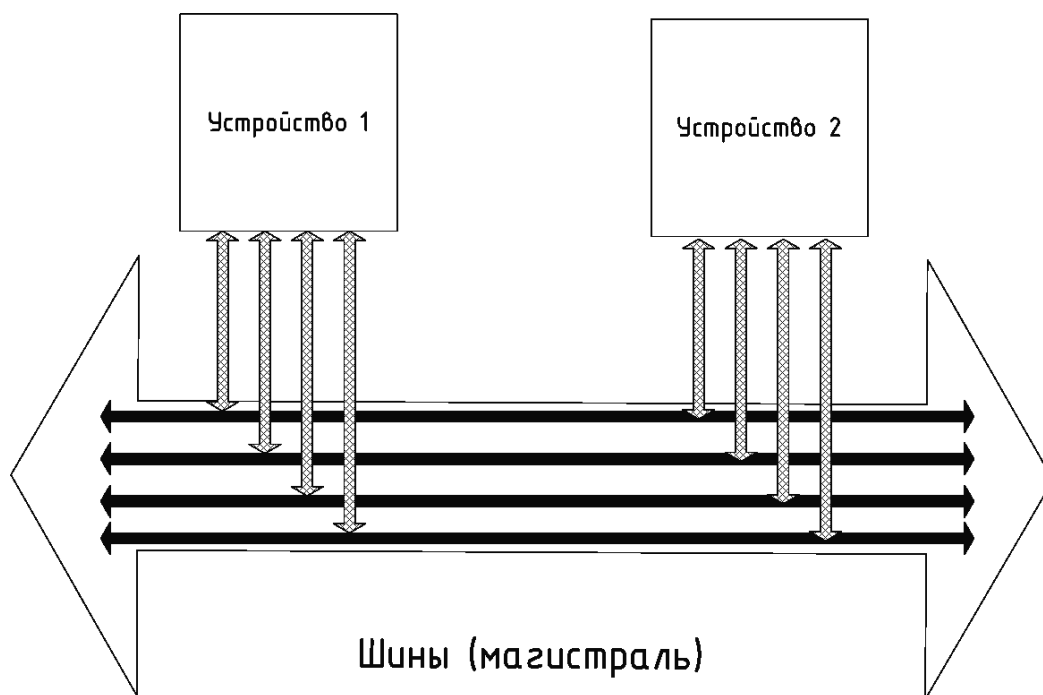


Рисунок 1.4. Шинная структура связей

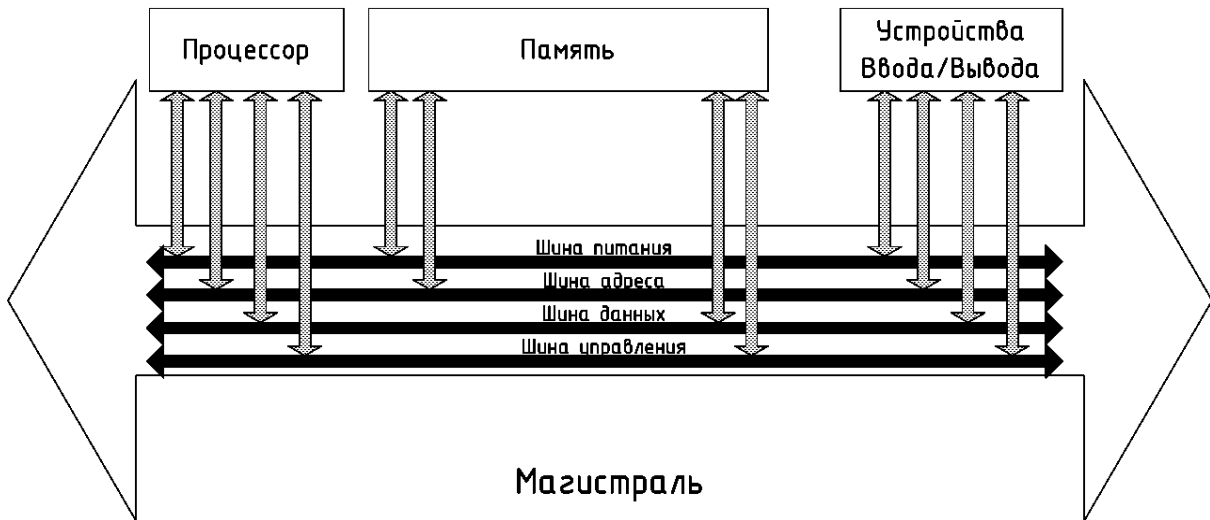
**Структура микропроцессорной системы**

Характеристики микропроцессорной системы:

1. Каждое устройство в микропроцессорной системе, каждая ячейка памяти имеют адрес. Адрес выставляется на *шину адреса*.
2. Коды данных выставляются на *шину данных*.
3. Сигналы *шины управления*: *стробы* (они говорят о том, что код выставлен на шину), сигналы *подтверждения* (часто свидетельствуют о том, что код

получен устройством, участвующим в обмене), сброс устройств (рестарт), тактирование устройств (синхронизация работы).

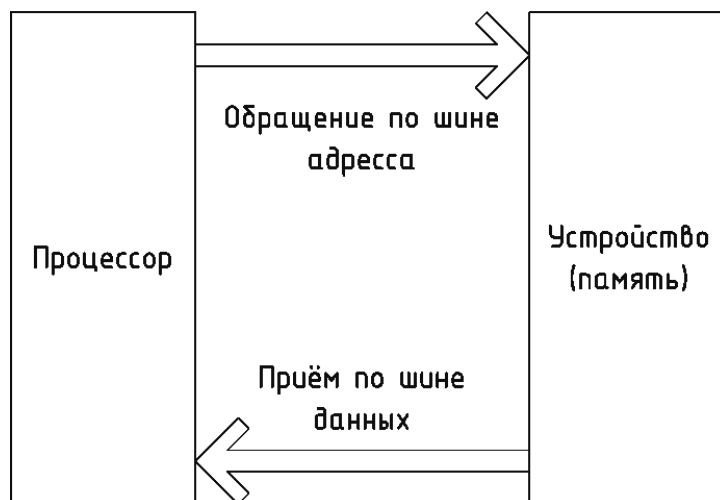
На рис. 1.5 представлена структура микропроцессорной системы.



**Рисунок 1.5.** Структура микропроцессорной системы

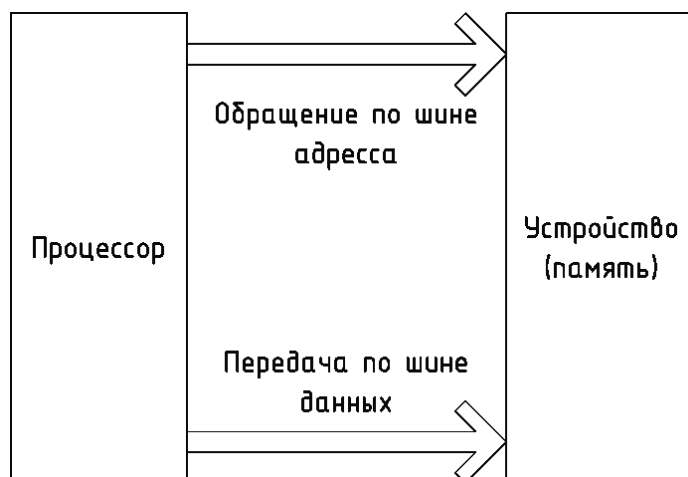
### Обмен информацией

Ввод входного кода в микропроцессорную систему (рис. 1.6):



**Рисунок 1.6.** Ввод входного кода в микропроцессорную систему

## Вывод выходного кода из микропроцессорной системы (рис. 1.7)



**Рисунок 1.7.** Вывод выходного кода из микропроцессорной системы

### **Виды обмена информацией**

#### **1. Программный обмен информацией.**

##### Характеристики:

- Процессор инициирует все операции обмена информацией.
- На внешние события, не связанные с программой, процессор не реагирует.
- Все сигналы магистрали контролируются процессором.

#### **2. Обмен по прерываниям.** Применяется в тех случаях, когда требуется реакция процессора на приход внешнего сигнала (от клавиатуры, сетевого адаптера, принтера).

##### Способы организации реакции на внешнее событие:

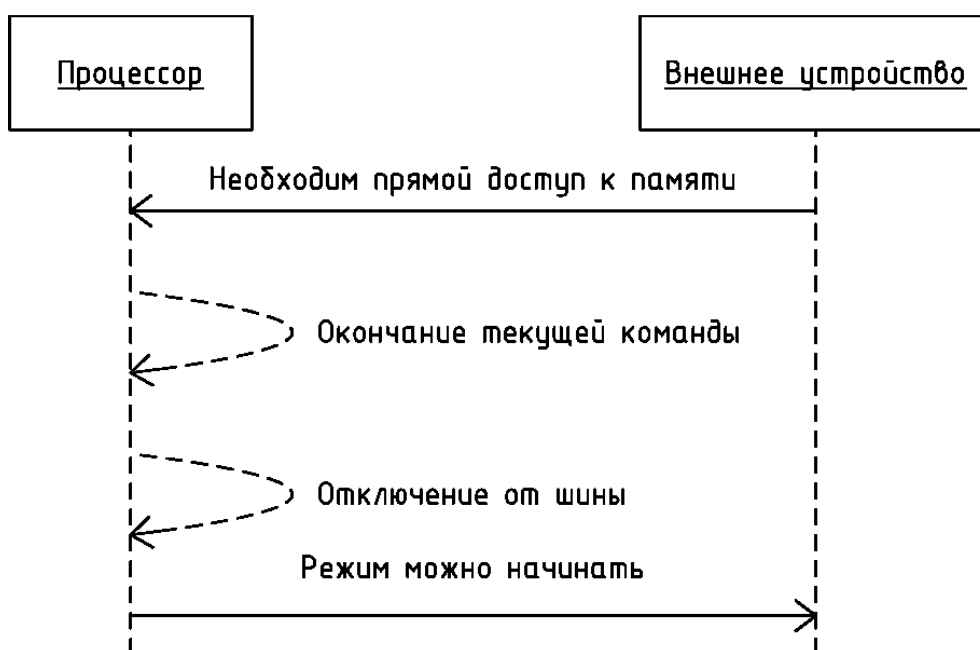
- *Метод опроса флага* (процессор считывает специальную информацию из устройства, которому требуется прерывание процессора; процессор может и не обрабатывать такие «маскируемые» прерывания).
- *Прерывание работы процессора* (в данном случае процессор обязан приостановить свою работу над текущей задачей и переключиться на ту, которую требует прервавшее его устройство; обычно это так называемые «немаскируемые» прерывания, требующие неотлагательной обработки).

- *Прямой доступ к памяти* (как способ организации реакции на внешнее событие) (процессор отключается от системной магистрали и предоставляет право прервавшему его работу устройству самостоятельно выполнить требуемые операции работы с памятью).

### 3. *Прямой доступ к памяти* (как вид обмена информацией) (рис. 1.8).

#### Характеристики:

- Обмен по шине идёт без участия центрального процессора системы.
- Необходим *контроллер прямого доступа* к памяти либо в самой микропроцессорной системе, либо в устройстве ввода/вывода.



**Рисунок 1.8.** Прямой доступ к памяти

## 1.4. Разновидности микропроцессорных систем

### Архитектура с общей шиной для команд и данных

Такая архитектура носит названия – *принстонская* или *архитектура фон Неймана*. В данном случае единая шина позволяет гибко распределять объём между кодами команд и кодами данных (рис 1.9).



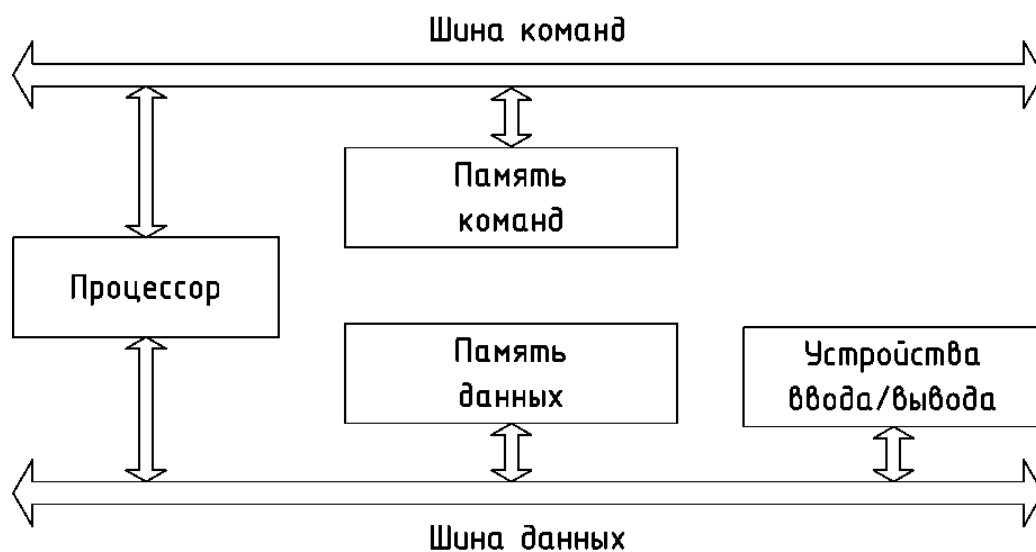
**Рисунок 1.9.** Архитектура фон Неймана (принстонская)

### Архитектура с отдельными шинами команд и данных

Такая архитектура называется *гарвардской* (рис. 1.10).

Характеристики:

1. Программа размещается в памяти команд, данные в памяти данных.
2. Обмен по шинам может идти параллельно во времени.



**Рисунок 1.10.** Гарвардская архитектура

### Типы микропроцессорных систем

Далее перечисляются типы микропроцессорных систем:

1. **Микроконтроллеры.** Узлы системы, выполненные в виде одной микросхемы. Микроконтроллер представляет собой специализированный

микропроцессор для использования в контроллерах внешних устройств или приборов.

**2. Контроллеры.** Микропроцессорные системы, выполненные в виде отдельных модулей.

*Контроллер в вычислительной машине* – специализированный процессор (в отличие от центрального), предназначенный для управления внешними устройствами – накопителями, дисплеями, принтерами. Наличие такого контроллера освобождает центральный процессор от выполнения этих функций.

*Контроллер устройства* – устройство сопряжения вычислительной машины с внешним устройством и управления обменом между ними. По сравнению с адаптером, такой контроллер выполняет более сложные функции.

**3. Микрокомпьютеры.** Микропроцессорные системы с развитыми средствами сопряжения с внешними устройствами. Микрокомпьютер – вычислительная машина малых размеров, созданная на базе микропроцессора. Выделяют встроенные и персональные, настольные и портативные, профессиональные и бытовые.

**4. Компьютеры.** Универсальные микропроцессорные системы. Под компьютером понимают комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

Стоит сделать несколько замечаний и привести ряд примеров касательно вышеприведённой классификации.

В вычислительных центрах, предоставляющих «высокопроизводительные» услуги, стоят *компьютеры*, объединённые для совместной работы (часто называемые *комплексами*). Так называемые «персональные» компьютеры тоже относятся к классу *компьютеров*.



К *микрокомпьютерам* можно отнести карманные мобильные устройства, устройства воспроизведения и записи цифровой информации, предназначенные для прослушивания и просмотра аудиовизуального содержимого. Сегодня микрокомпьютеры могут быть обнаружены даже в холодильниках (такие холодильники можно программировать по своему усмотрению, выходить в интернет и просматривать телепередачи, а также общаться с друзьями и единомышленниками в специализированных сетевых сообществах).

Как в компьютерах, так и в микрокомпьютерах могут присутствовать *контроллеры* и *микроконтроллеры*. Причём микроконтроллеры могут работать в рамках контроллеров. Контроллеры могут быть независимыми устройствами и самостоятельно (почти самостоятельно) управлять внешними устройствами и собирать информацию с объектов. Более подробно о типах контроллеров – в теме 7.

## **Вопросы**

1. Какая основная идея классической структуры связей?
2. Перечислите характеристики классической структуры связей.
3. Какая основная идея шинной структуры связей?
4. Перечислите характеристики шинной структуры связей. В чём её недостатки (и каковы тогда преимущества)?
5. Что такое мультиплексированная передача данных? Каковы её достоинства и недостатки?
6. Что такое протокол обмена информацией?
7. Каковы характеристики микропроцессорной системы?
8. Какое отличие между шиной и магистралью. Какие бывают шины (и какие бывают магистрали, привести примеры)?
9. Какие вы можете назвать типы сигналов шины управления (согласны ли вы с такой классификацией)?
10. Как происходит ввод входного кода в микропроцессорную систему?

11. Как происходит вывод выходного кода из микропроцессорной системы?
12. Перечислите виды обмена информацией, опишите их составляющие и характеристики.
13. Какие вы знаете способы организации реакции на внешнее событие (внешнее относительно микропроцессорной системы)?
14. Что такое принстонская архитектура (есть ли ещё названия у этой архитектуры)?
15. Что такое гарвардская архитектура?
16. Какие вы знаете типы микропроцессорных систем?
17. Дайте характеристику следующим системам: микроконтроллер, контроллер, микрокомпьютер, компьютер. Где они используются?

## **ТЕМА 2. ЦИКЛЫ ОБМЕНА ИНФОРМАЦИЕЙ**

### **Лекция 3**

#### **2.1. Циклы обмена информацией**

##### **Циклы обмена**

Ещё одно требование к разработчику микропроцессорных систем – знать *принципы организации обмена информацией по шинам*. Эти знания намного важнее, чем знания особенностей конкретных микропроцессоров, т.к. жизненный цикл магистрали дольше, чем у процессоров.

Обмен информацией происходит в циклах обмена.

Определение 1. *Цикл обмена информацией* – временной интервал, в течение которого происходит выполнение одной элементарной операции обмена по шине (например, пересылка кода данных).

### Типы циклов:

1. **Цикл записи** (вывода). Процессор записывает (выводит) информацию.
2. **Цикл чтения** (ввода). Процессор читает (вводит) информацию.
3. **Цикл чтение-модификация-запись** (ввод-пауза-вывод). Процессор читает информацию из памяти или устройства ввода/вывода, преобразует её и записывает по тому же адресу.
4. **Цикл прямого доступа к памяти.**
5. **Цикл запроса и предоставления прерывания.**

Длительность цикла включает в себя несколько периодов сигнала тактовой частоты (в самом лучшем случае за один такт выполняется один элементарный цикл, а за счёт особой организации микропроцессора можно добиться выполнения команды за один такт).

$$Ч_{ц} \leq Ч_{тп},$$

где  $Ч$  – частота,  $ц$  – цикл,  $тп$  – тактовый процессор.

### **Шины магистралей**

#### Шина данных

Количество линий связи – *количество разрядов.*

Шина данных определяет:

1. **Скорость** системы.
2. **Эффективность** информационного обмена (качество передачи).
3. Максимально возможное **количество команд** (например, если у нас всего три линии в шине, то максимальное количество распознаваемых команд – 8, если не использовать никаких «ухищрений», например передачу команды за несколько циклов).

Шина данных *двунаправленная*. Она также определяет *разрядность всей магистрали*. В таком случае говорят не о внутренней (разрядность регистров и

арифметико-логического устройства), а о внешней (по отношению к процессору) разрядности. Внутренняя и внешняя разрядности могут не совпадать.

### **Шина адреса**

Шина адреса определяет:

**1. Допустимый *объём прямоадресуемой памяти.***

$2^N$  – *количество прямых адресов*, обеспечиваемых шиной адреса, N – количество разрядов шины.

Шина адреса может быть *однонаправленной* и *двунаправленной*. В первом случае магистралью управляет только процессор. Во втором случае предусмотрен режим *прямого доступа к памяти*.

Часто применяется *мультиплексирование* шин адреса и данных. В таком случае, циклы передачи адреса и данных происходят по одной шине, но в разное время. Мультиплексирование осуществляется за счёт специальных сигналов, передающихся по шине управления.

### **Шина управления**

Управляющие сигналы определяют *тип текущего цикла*, фиксируют *моменты стадий цикла*, согласуют работу центрального процессора с работой памяти или устройств ввода/вывода, обслуживают запросы на предоставление прерываний, а также запросы на прямой доступ к памяти.

Линии шины управления могут быть *однонаправленными* или *двунаправленными*.

### **Обмен данными**

Типы обмена данными:

**1. Синхронный обмен.** Процессор заканчивает обмен через *установленный временной интервал*.

**2. Асинхронный обмен.** Процессор заканчивает обмен, когда внешнее устройство *подтверждает* выполнение операции *специальным сигналом*.

В зависимости от типа обмена данными магистрали шин могут быть *синхронными* или *асинхронными*.

## 2.2. Циклы программного обмена

### Обмен по мультиплексированной асинхронной магистрали

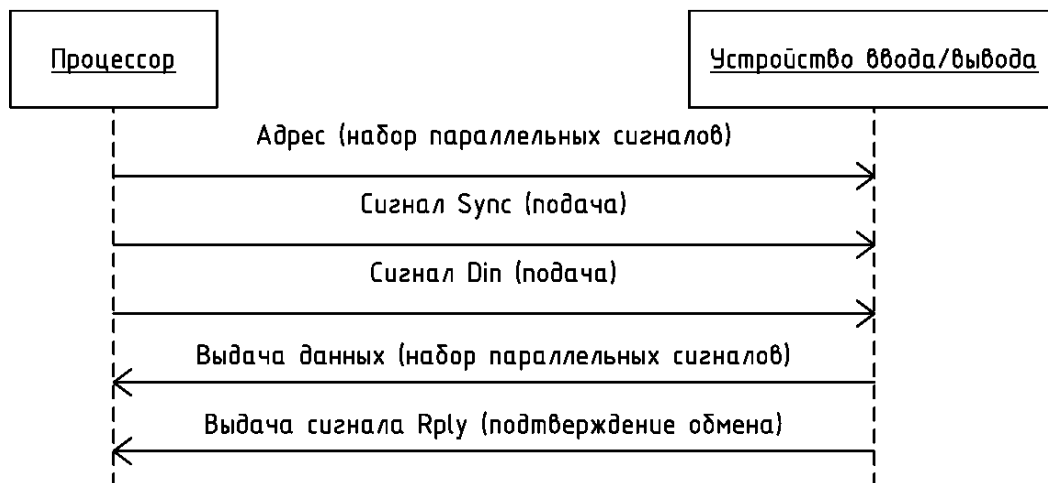
Рассмотрим пример обмена по *мультиплексированной асинхронной* магистрали **Q-bus** от фирмы DEC. Эта магистраль применялась в компьютерах и промышленных контроллерах и является хорошим примером для демонстрации организации асинхронного обмена по мультиплексированной (объединённой) шине адреса/данных.

#### Цикл чтения (рис. 2.1)

*Фаза адреса:* 1 – 4 шаг. *Фаза данных:* 5 – 6 шаг.

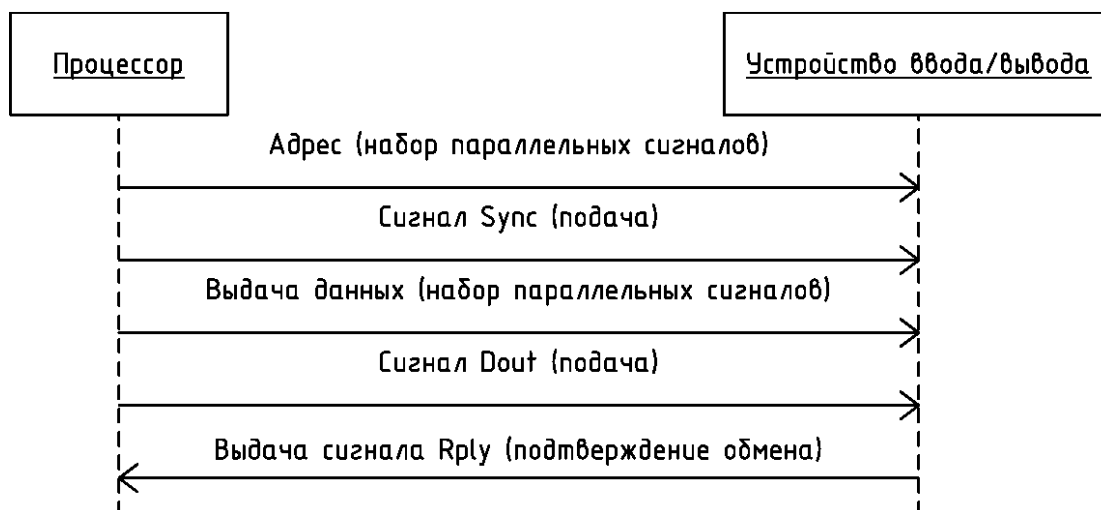
#### Шаги:

1. Процессор выставляет код адреса на шину данных/адреса.
2. Для стробирования используется сигнал **Sync**, выставляемый процессором на шину управления.
3. Получив код адреса, распознав его, устройство ввода/вывода или память готовится к обмену данными.
4. Через некоторое время после сигнала **Sync** процессор снимает адрес и начинает фазу данных.
5. Процессор выставляет сигнал строба чтения данных **Din**.
6. Устройство выставляет код данных, одновременно подтверждая выполнение операции сигналом подтверждения обмена **Rply** (сигнал подтверждения говорит о том, что устройство готово к следующему циклу обмена).



**Рисунок 2.1.** Циклы программного обмена по мультиплексированной асинхронной магистрали. Цикл чтения данных

**Цикл записи** (рис. 2.2)



**Рисунок 2.2.** Циклы программного обмена по мультиплексированной асинхронной магистрали. Цикл записи

Шаги фазы данных (фаза адреса такая же, как и в предыдущем случае):

1. Процессор выставляет на шину адреса/данных код данных, сопровождает это сигналом строба записи **Dout**.
2. Устройство принимает данные от процессора, формирует сигнал подтверждения **Rply**.
3. Процессор, получив сигнал **Rply**, заканчивает цикл обмена.

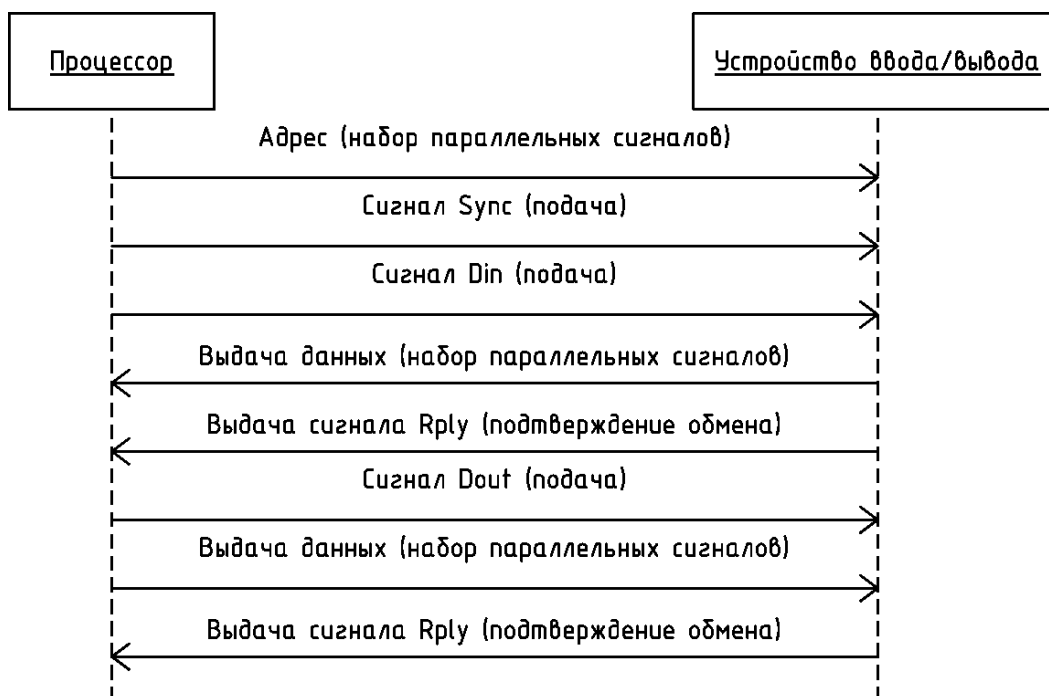
4. Процессор снимает сигнал **Dout**.
5. Устройство снимает сигнал **Rply**.

Таким образом, адрес передаётся *синхронно*, а данные передаются *асинхронно*.

### Цикл чтение-модификация-запись (рис. 2.3)

Фаза адреса такая же, как и в предыдущем случае. Моменты фазы данных:

1. Процессор производит сначала чтение, затем запись по одному и тому же адресу.
2. Для чтения используется строб **Din**, для записи используется строб **Dout**.
3. На сигнал процессора **Din** устройство выдаёт данные, на сигнал **Dout** устройство принимает данные по шине адреса/данных.
4. Устройство подтверждает выполнение каждой операции сигналом подтверждения **Rply**.
5. Сигнал **Sync** вырабатывается процессором в начале цикла и держится до окончания всего цикла.



**Рисунок 2.3.** Цикл программного обмена по мультиплексированной асинхронной магистрали. Цикл чтение-модификация-запись

## Обмен по синхронной не мультиплексированной магистрали

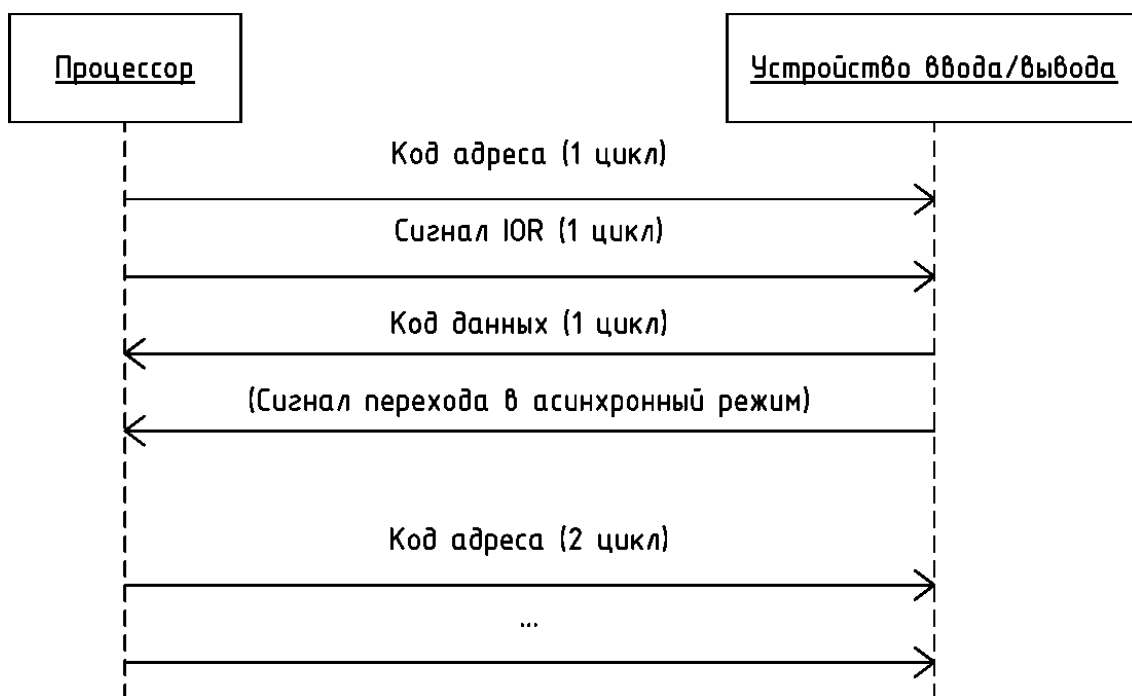
Рассмотрим другой пример. В данном случае обмен будет происходить по *синхронной не мультиплексированной* магистрали **ISA** от фирмы IBM. Магистраль эта широко используется в персональных компьютерах.

Циклы чтения (рис. 2.4) и записи (рис. 2.5) одинаковы для фазы адреса:

1. Процессор выставляет код адреса на шину адреса. Сигнал на шине остаётся до конца цикла.
2. Фаза адреса заканчивается (сигнал на шине адреса всё ещё остаётся) с началом строба **IOR** (для чтения) или **IOW** (для записи).
3. Устройство принимает адрес, распознаёт его, готовится к обмену.

Фаза данных (цикл чтения):

1. Процессор выставляет сигнал **IOR** (готовность к чтению).
2. Устройство выдаёт на шину данных код данных.
3. Через определённое время сигнал **IOR** снимается процессором, чуть позже снимается код адреса с шины адреса.



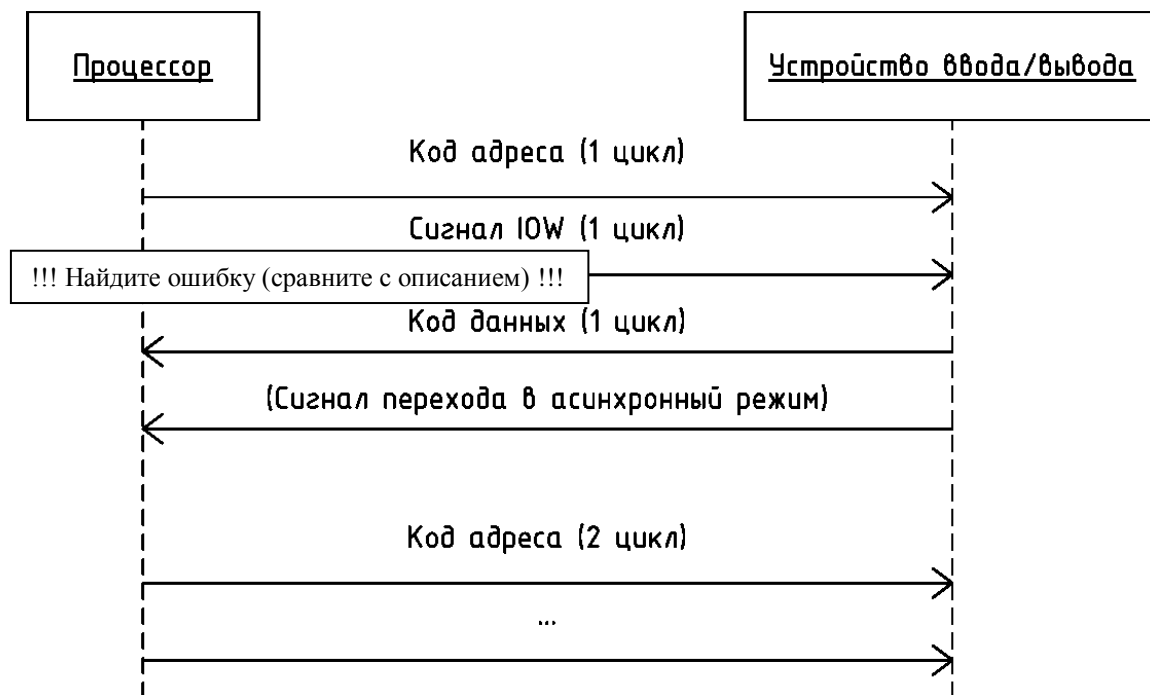
**Рисунок 2.4.** Цикл программного обмена по синхронной не мультиплексированной магистрали. Цикл чтения



Шина **ISA** допускает *асинхронный обмен*. В этом случае применяется *сигнал готовности канала*. Если устройство не успевает за процессором, оно может сообщить процессору о неготовности и перейти в асинхронный режим. По своему усмотрению устройство может не сообщать о неготовности (если поспевает за процессором).

Фаза данных (цикл записи):

1. Процессор выставляет на шину данных код данных, сопровождая это стробом записи **IOW**.
2. Устройство принимает код данных с шины данных.
3. Если устройство не успевает, оно снимает тот самый сигнал, который говорит процессору о том, что устройство способно поддерживать обмен данными на текущей скорости. Снятие этого сигнал (готовности канала) можно рассматривать как передачу процессору сообщения о том, что устройство не успевает за работой процессора и необходим переход в асинхронный режим.



**Рисунок 2.5.** Цикл программного обмена по синхронной не мультимплексированной магистрали. Цикл записи

## Вопросы

1. Почему знание принципов организации обмена информацией по шинам микропроцессорной системы важнее знания особенностей конкретных микропроцессоров?
2. Что такое цикл обмена информацией?
3. Дайте пример элементарной операции обмена по шине.
4. Какие типы циклов обмена вы знаете? В чём их отличие?
5. Как связаны между собой длительность цикла обмена по шинам микропроцессорной системы и частота тактов процессора?
6. Что определяет шина данных?
7. Что имеют в виду, когда говорят о разрядности магистрали?
8. Что определяет шина адреса?
9. Чем ограничен допустимый объём «прямоадресуемой» внутренней памяти микропроцессорной системы?
10. Как связано количество разрядов шины адреса и количество прямых адресов внутренней памяти микропроцессорной системы? Как вы думаете, почему адреса называются «прямыми»?
11. Что имеют в виду, когда говорят, что шина микропроцессорной системы однонаправленная (или двунаправленная)?
12. Что такое мультиплексирование шин и за счёт чего оно осуществляется?
13. Что определяют сигналы шины управления?
14. Какие вы знаете типы обмена данными?
15. Дайте пример мультиплексированной асинхронной магистрали. Где она применяется?
16. Что такое цикл чтения, цикл записи, фаза адреса и фаза данных? Как они связаны друг с другом? В какой последовательности?
17. Дайте пример синхронной не мультиплексированной магистрали. Где она используется?

18. Каким образом синхронная магистраль может перейти в асинхронный режим?
19. Можно ли не меняя аппаратной части микропроцессорной системы из мультиплексированной магистрали сделать не мультиплексированную?

## Лекция 4

### 2.3. Циклы обмена по прерываниям

#### Типы прерываний

Типы прерывания:

1. **Векторные.** Требуют проведения цикла чтения по магистрали. В данном случае устройство передаёт процессору *номер прерывания*. Процессор проводит цикл чтения по магистрали, по шине данных получает код номера прерывания. Такой способ обмена реализован в магистрали **Q-bus**.
2. **Радиальные.** Не требуют цикла обмена по магистрали. В магистрали количество линий запроса прерываний есть *количество различных прерываний*. Отдельное прерывание запрашивается по отдельной линии. В систему включается дополнительная микросхема *контроллера прерываний*. Реализацию описанного метода можно встретить в магистрали **ISA**.

#### Обмен по векторным прерываниям

Рассмотрим обмен по прерываниям на магистрали Q-bus (рис. 2.6 – 2.7).

Шаги обмена:

1. Запрос прерывания осуществляется сигналом **VIRQ**. Этот сигнал формируется одним из устройств.
2. Процессор прекращает выполнение текущей команды и предоставляет прерывание (свои услуги).
3. Процессор выставляет сигнал **Din** (чтение данных) и сигнал **IAKO** (предоставление прерывания).

4. Сигнал **IAKO** последовательно распространяется через все устройства. Устройство, запросившее прерывание, не пропускает сигнал далее.
5. Устройство, получив сигнал **IAKO**, снимает сигнал **VIRQ**.
6. Процессор проводит цикл безадресного чтения номера прерывания.
7. Устройство в ответ на сигналы **Din** и **IAKO** выдаёт на шину адреса/данных код номера прерывания (адрес вектора прерывания) и выставляет сигнал подтверждения **Rply**.
8. Процессор читает код номера прерывания и заканчивает цикл чтения снятием сигналов **Din** и **IAKO**.

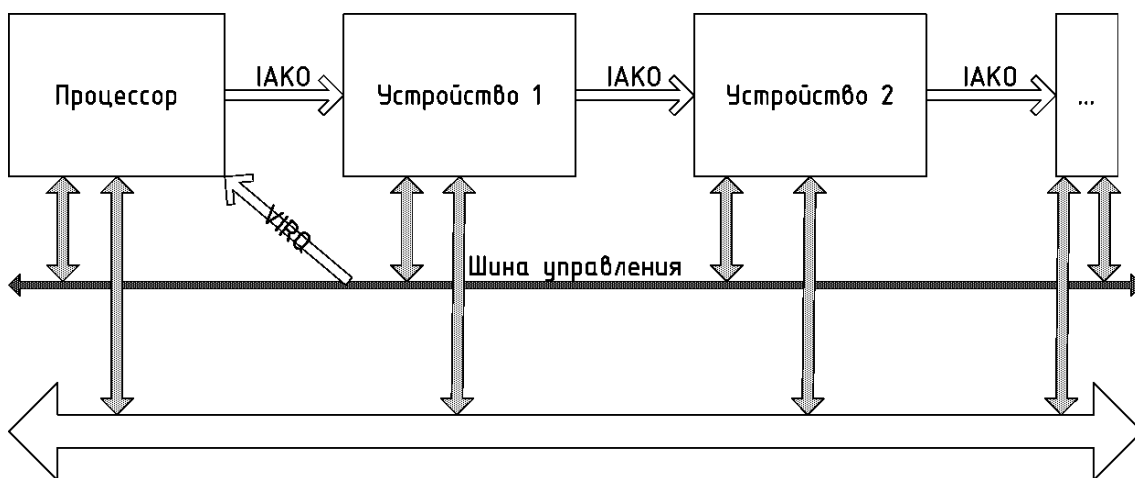


Рисунок 2.6. Распространение сигналов по магистрали Q-bus



Рисунок 2.7. Цикл запроса и предоставления прерывания магистрали Q-bus

## Обмен по радиальным прерываниям

Рассмотрим пример обмена по прерываниям на магистрали ISA (рис. 2.8):

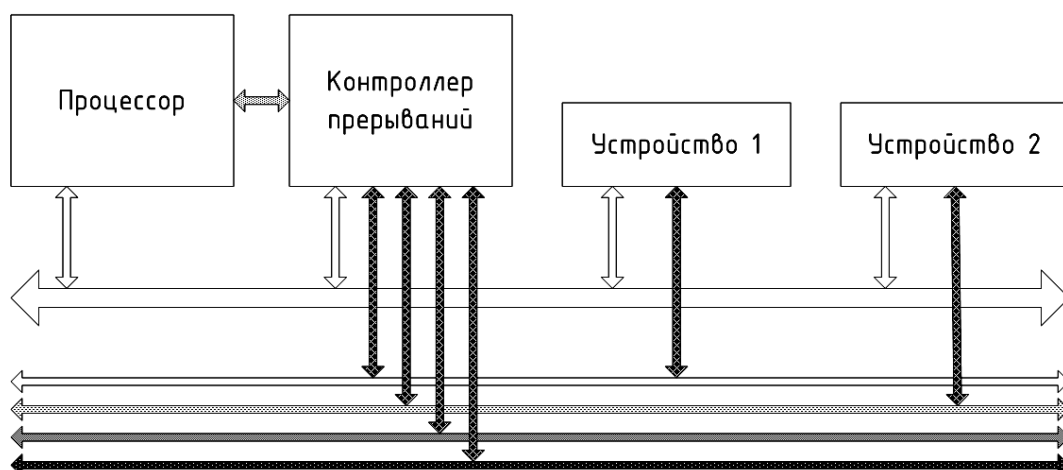


Рисунок 2.8. Магистраль ISA. Обмен по прерываниям

### Особенности:

1. Процессор общается с контроллером прерываний *по магистрали* для задания режима работы и *вне магистрали* для обработки запросов на прерывания.
2. Сигналы запросов прерываний распределяются между всеми устройствами магистрали.
3. На каждую линию прерывания приходится одно устройство.
4. Порядок обслуживания прерываний определяет контроллер прерываний.

## 2.4. Циклы обмена в режиме прямого доступа к памяти

### Обмен в режиме прямого доступа к памяти

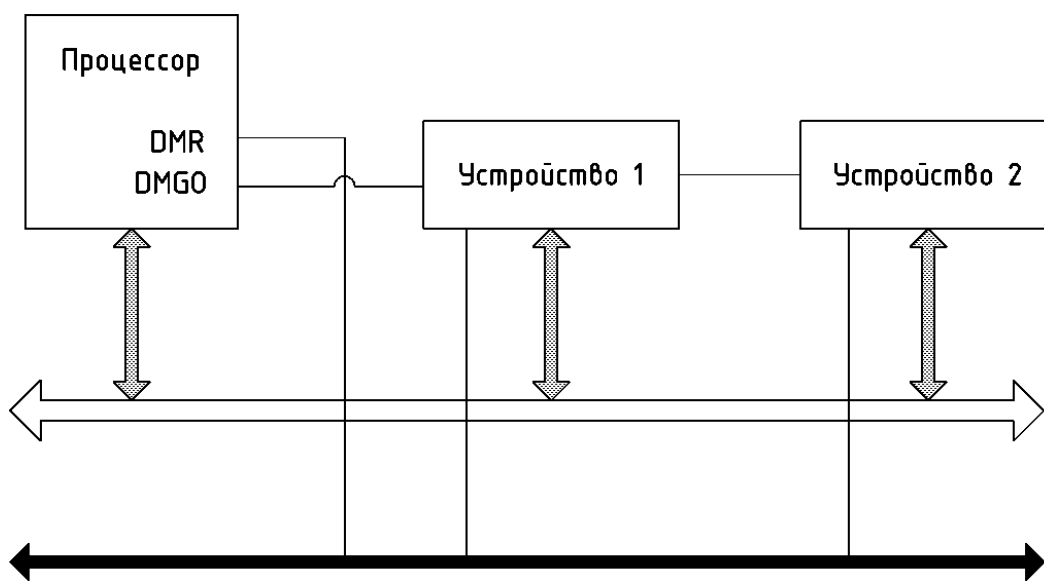
Принцип – устройство запрашивает прямой доступ к памяти и получает его. Процессор отключается от магистрали.

Рассмотрим пример обмена на магистрали Q-bus (рис. 2.9):

### Шаги обмена:

1. Устройство, нуждающееся в прямом доступе к памяти, передаёт сигнал запроса **DMR**.

2. Получив сигнал **DMR**, процессор выдаёт сигнал предоставления прямого доступа к памяти – **DMGO**. Сигнал последовательно распространяется через устройства. Устройство, нуждающееся в прямом доступе к памяти, не распространяет сигнал далее.
3. Устройство, получившее прямой доступ к памяти, проводит циклы обмена по магистрали.



**Рисунок 2.9.** Магистраль Q-bus. Прямой доступ к памяти

Рассмотрим далее пример обмена на магистрали **ISA** (рис. 2.10).

Основные моменты:

1. Сигналы запроса прямого доступа к памяти (**DRQ**) поступают к контроллеру от устройств.
2. Контроллер прямого доступа к памяти распределяет сигналы предоставления доступа (**DACK**) устройствам.
3. К каждому каналу прямого доступа к памяти подключается только одно устройство, запрашивающее доступ.
4. После рассылки сигнала **DACK**, контроллер прямого доступа проводит циклы обмена по магистрали между устройствами.

5. На магистрали **ISA** используются отдельные стробы записи в память **MEMW** и запись в устройства ввода/вывода **IOW**, а также отдельные стробы чтения из памяти **MEMR** и чтения из устройств ввода/вывода **IOR**.
6. Адрес памяти выставляется на шину адреса. Адрес устройств ввода/вывода может быть заменён специальным сигналом.

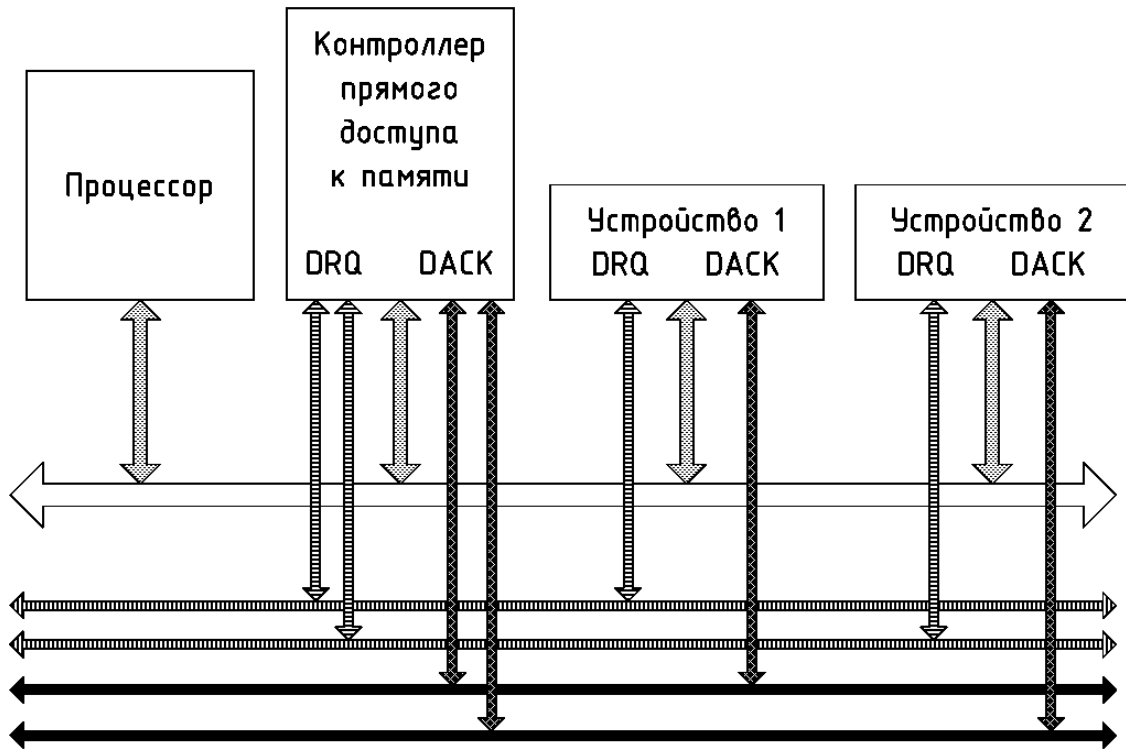


Рисунок 2.10. Магистраль ISA. Прямой доступ к памяти

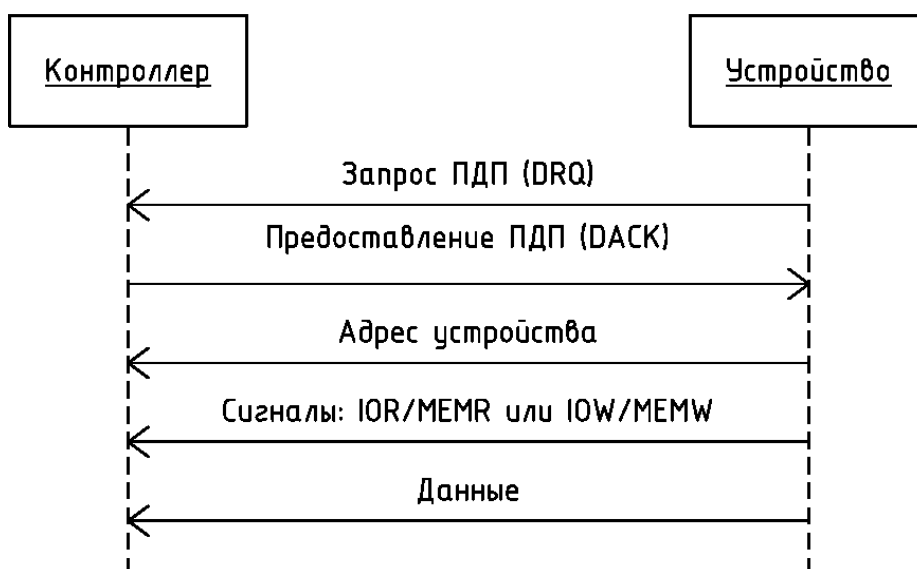


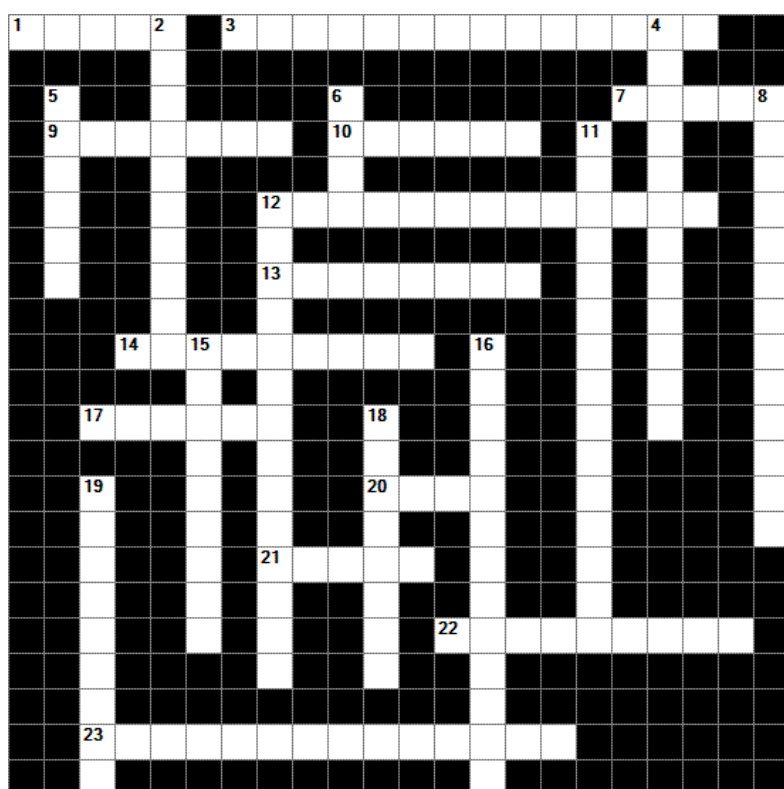
Рисунок 2.11. Магистраль ISA

## Вопросы

1. В чём отличие векторных прерываний от радиальных?
2. В чём состоят особенности обмена по радиальным прерываниям?
3. Зачем нужен контроллер прерываний?
4. Зачем нужен контроллер прямого доступа к памяти?

## Промежуточный контроль

Для закрепления (и проверки) знаний основных понятий и определений студентам предлагается решить кроссворд.



**По горизонтали:** **1.** Структурная составляющая процессора. **3.** Показатель скорости выполнения электронной системой своей задачи. **7.** Сигнал шины управления, который перезагружает устройства. **9.** Структурная составляющая процессора. **10.** Требование к электронной системе в виде набора функций. **12.** Структурная составляющая процессора. **13.** Совокупность устройств, образующих единое целое. **14.** Тип прерываний, требующий проведения цикла чтения по магистрали. **17.** Характеристика системы команд, которая выражается количеством занимаемых разрядов ячеек памяти. **20.** Совокупность взаимосвязанных элементов и

узлов устройства, выполняющих определённую функцию. **21.** Сигнал шины управления, который говорит о том, что код выставлен на шину. **22.** Правила обмена информацией между устройствами, участвующими в обмене. **23.** Микропроцессорная система с развитыми средствами сопряжения с внешними устройствами.

**По вертикали:** **2.** Тип прерываний, не требующий цикла обмена по магистрали. **4.** Степень соответствия возможностей электронной системы решаемой задаче. **5.** Самостоятельное устройство. **6.** Часть устройства. **8.** Характеристика системы команд, которая выражается кругом решаемых задач. Может быть узкой или широкой. **11.** Узел системы, выполненный в виде одной микросхемы. **12.** Ключевой элемент микропроцессорной системы. Заменяет «жесткую» логику. **15.** Универсальная микропроцессорная система. **16.** Шина данных определяет \*\*\* информационного обмена. **18.** Способность электронной системы подстраиваться под определённые задачи. **19.** Набор цифровых кодов, расшифровав которые, процессор узнаёт, что ему надо делать.



## ТЕМА 3. ФУНКЦИИ УСТРОЙСТВ МАГИСТРАЛИ

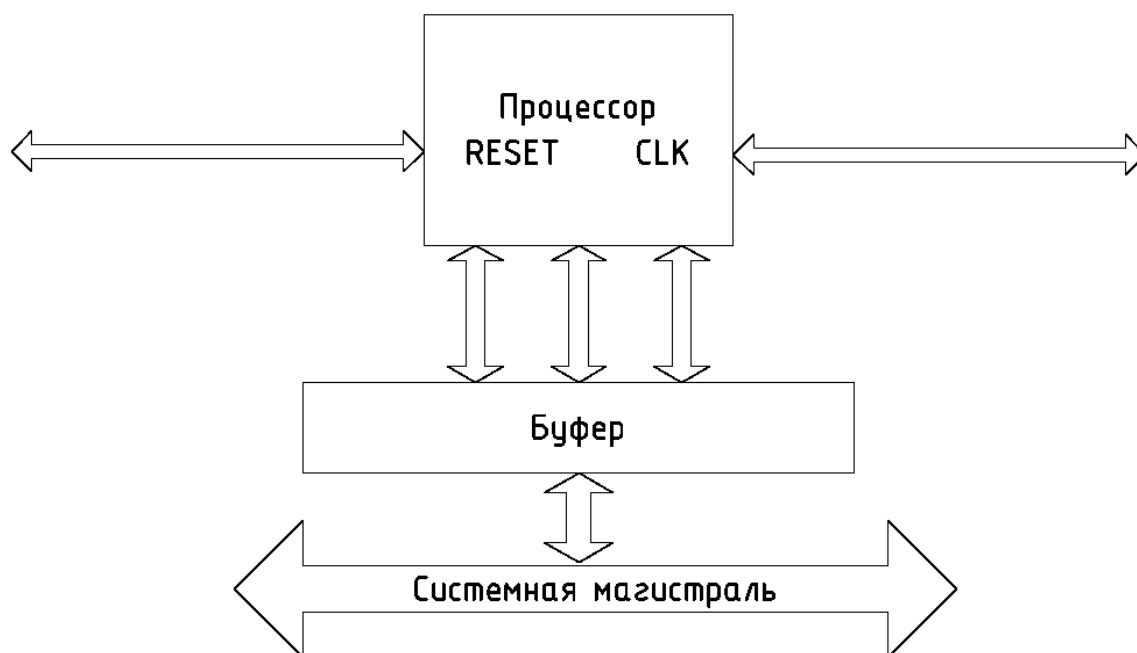
### Лекция 5

#### 3.1. Устройство процессора

##### Процессор

Микросхема процессора имеет выходы трёх шин: *адреса, данных, управления*; вывод для подключения внешнего тактового сигнала **CLK**; вывод для сигнала начального сброса **RESET**.

Схема включения процессора представлена на рис 3.1.



**Рисунок 3.1.** Схема включения процессора

Смысл буферных микросхем:

1. Демultipлексирование сигналов.
2. Согласование протоколов обмена.

После включения питания процессор переходит в первый адрес программы начального пуска и выполняет эту программу, находящуюся в постоянной

памяти. Далее процессор выполняет программу из ПЗУ или ОЗУ (например, программу загрузчика операционной системы).

### **Схема управления выборкой команд**

Схема управления выборкой команд необходима для чтения команд и их дешифрации.

В 16-разрядных процессорах появился *конвейер* (очередь) команд, позволяющий выбирать несколько следующих команд, пока выполняется предыдущая команда.

### **Арифметико-логическое устройство**

Выполняемые операции:

1. *Логические* операции: И (AND), ИЛИ (OR), исключающее ИЛИ (XOR).
2. *Арифметические* операции: сложение (+), вычитание (–), умножение (\*), деление (/).

Пути повышения производительности:

1. Довести время выполнения команды до одного такта.
2. Уменьшить количество выполняемых АЛУ команд (т.е. сократить сложность устройства).
3. Использование нескольких параллельно работающих АЛУ.

### **Регистры**

Функции регистров:

1. Временное хранение кодов.
2. Предоставление ячеек очень быстрой памяти.

*Внутренняя разрядность процессора* – разрядность регистров и АЛУ. Может не совпадать с внешней разрядностью (т.е. разрядностью магистрали).

### Подходы к назначению регистров:

1. Отведение регистру строго определённой функции (как в процессорах от Intel).
2. Равноправие регистров (пример – процессор DEC T-11).
3. Промежуточное решение (яркий пример – процессор Motorola MC68000).

### Регистр признаков (состояния):

1. Содержимое – слово состояния процессора.
2. Каждый бит – *флаг*, который содержит информацию о результате выполнения предыдущей команды.
3. Бывают флаги управления, определяющие режим выполнения некоторых команд.

### **Управление**

#### Схема управления прерываниями:

1. Обрабатывает запросы на прерывание.
2. Определяет адрес начала программы обработки прерывания.
3. Сохраняет в памяти текущее состояние регистров процессора. Обеспечивает переход к программе обработки прерывания.

#### Схема управления ПДП (функции):

1. Приостановка работы процессора.

#### Логика управления (функции):

1. Организация взаимодействия узлов процессора.
2. Перенаправление данных.
3. Синхронизация работы процессора с внешними сигналами.
4. Реализация процедур ввода и вывода информации.

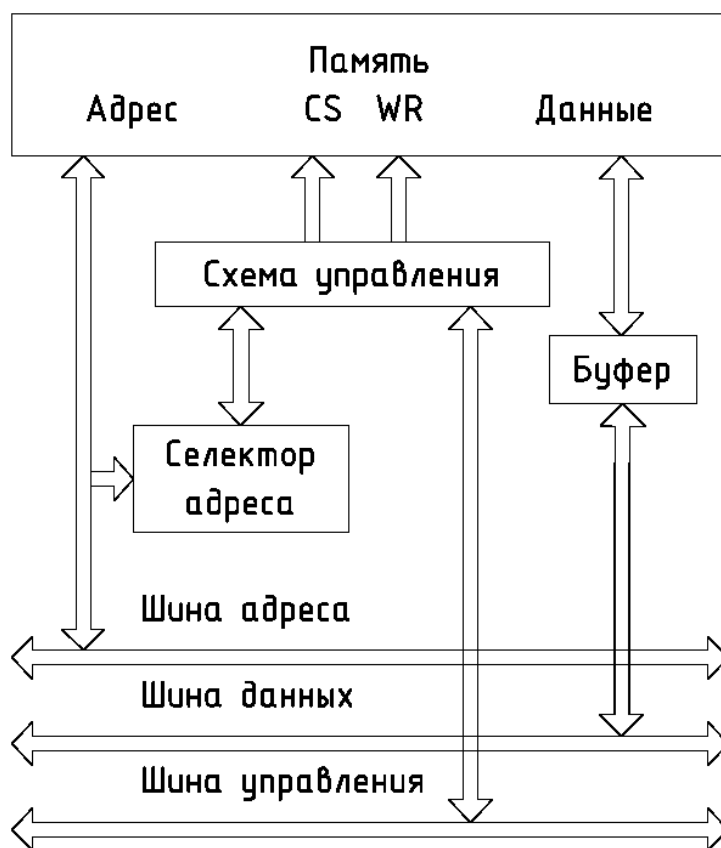
## 3.2. Память

### Устройство памяти

Характеристики устройства памяти:

1. Временное или постоянное хранение данных и команд.
2. Информация хранится в ячейках (у каждой ячейки есть адрес).
3. Для подключения к системной магистрали используются блоки сопряжения: дешифратор (*селектор*) адреса, схема обработки управляющих сигналов, буферы данных.

На рис. 3.2 представлена схема подключения устройства памяти к магистрали микропроцессорной системы.



**Рисунок 3.2.** Схема включения памяти

### Особенности:

1. Память обычно состоит из модулей. Каждый модуль работает в своей области пространства.
2. Селектор адреса (дешифратор) определяет, какая область адресов пространства памяти отведена данному модулю памяти.
3. Схема управления вырабатывает в нужные моменты сигналы разрешения работы памяти (**CS**) и сигналы разрешения записи в память (**WR**).
4. Буфер является посредником между памятью и магистралью.

### Память программы начального запуска:

1. Программа запуска находится в ПЗУ или флэш-памяти.
2. Программа запуска считывается во время включения питания или перезапуска системы.

### Стек (функции и характеристики):

1. Часть оперативной памяти. Временное хранение данных.
2. Заданный и неизменный способ адресации.
3. Запись в стек – по адресу, содержащемуся в регистре указателя стека, после чего содержимое указателя стека уменьшается (на длину записываемых в стек данных). Чтение из стека – из адреса, содержащегося в указателе стека, после чего содержимое указателя стека увеличивается.
4. В стеке можно прятать не только содержимое внутренних регистров процессора, но и содержимое регистра признаков, можно хранить произвольные данные.

### Таблица векторов прерывания:

1. Прерывания обрабатываются через таблицу векторов (указателей) прерываний.
2. Таблица векторов содержит адреса начала программ обработки прерываний, которые называются *векторами*.

3. Обычно таблица располагается в начале пространства памяти (ячейках памяти с малыми адресами).
4. В случае аппаратных прерываний номер прерывания задаётся устройством (передаётся по шине данных), запросившем прерывание (*векторное прерывание*) или номером линии запроса прерываний (*радиальное прерывание*).
5. В конце программы обработки прерываний располагается команда выхода из прерывания, после которой процессор возвращается к выполнению прерванной основной программы (адрес возврата извлекается из стека).
6. Параметры процессора на время выполнения программы обработки прерывания сохраняются в стеке (в том числе и адрес следующей после текущей команды).

Алгоритм обработки прерывания на примере:

**Шаг 1.** Процессор выполняет программу. Текущая команда расположена по адресу 5000. Поступает запрос прерывания с номером 4.

**Шаг 2.** Процессор заканчивает выполнение команды по адресу 5000. Сохраняет в стеке значение счётчика команд (5001 – т.е. адрес следующей команды) и текущее значение **PSW** (слово состояния процессора).

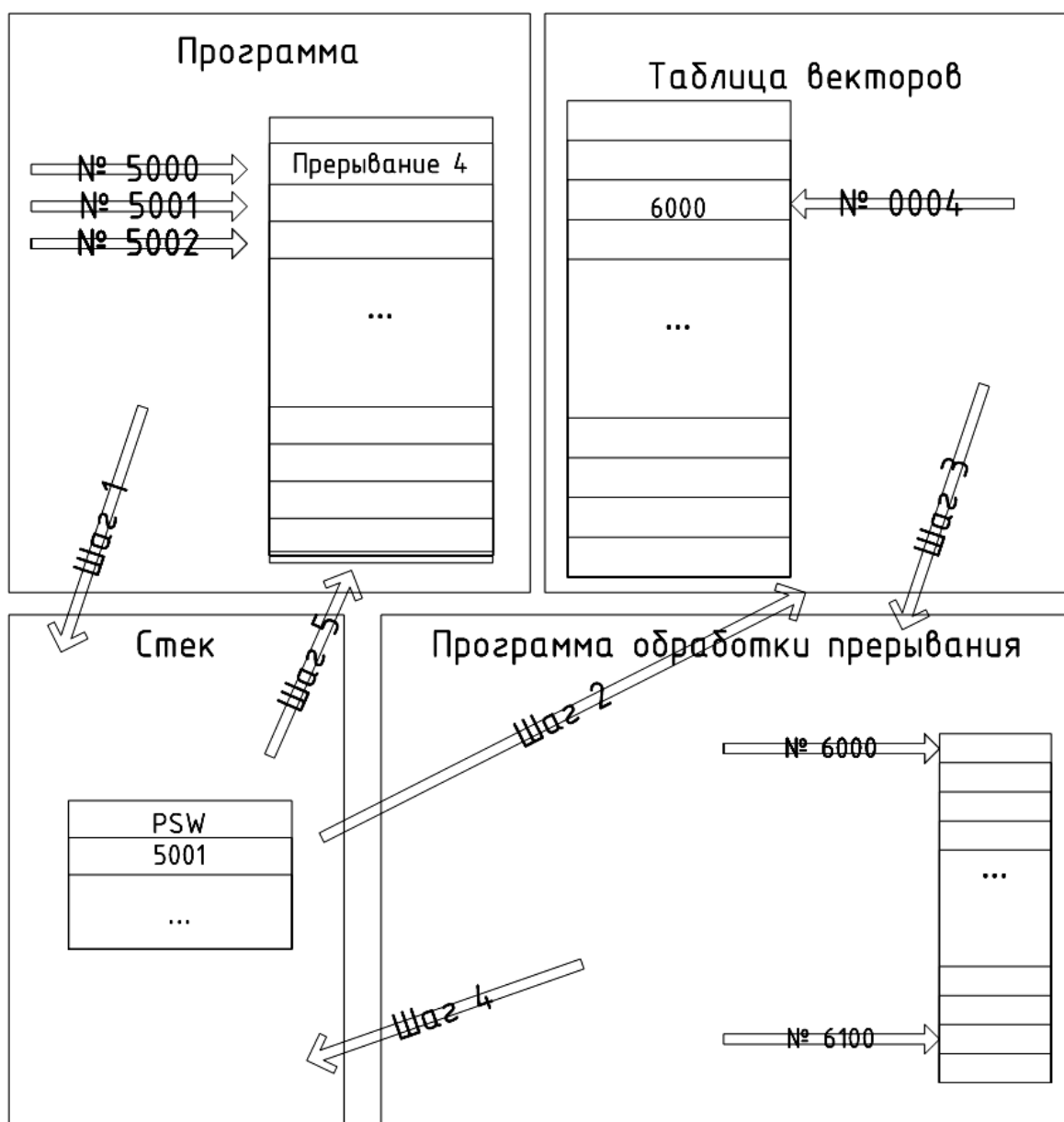
**Шаг 3.** Процессор читает из адреса 4 памяти код вектора прерывания. Он равен 6000. Процессор переходит в адрес 6000 и выполняет программу обработки прерывания.

**Шаг 4.** Дойдя до места, где программа обработки заканчивается (в нашем примере – 6100), процессор возвращается к выполнению прерванной программы.

**Шаг 5.** Процессор извлекает из стека значение адреса 5001 и **PSW**. Процессор читает команду по адресу 5001 и последовательно выполняет команды программы.

Возможно многократное вложение прерываний, когда во время выполнения программы обработки прерывания поступает новый запрос на прерывание.

В сложных случаях таблица векторов прерываний содержит дескрипторы прерываний. Результат обработки дескриптора – адрес начала программы обработки прерывания.



**Рисунок 3.3.** Алгоритм обработки прерывания

### Память устройств, подключенных к шине:

Подходы к разделению адресов памяти и устройств:

- Выделение в общем адресном пространстве области адресов для устройств ввода/вывода.
- Разделение адресных пространств памяти и устройств ввода/вывода.

В 1-м случае используются те же команды, что и при работе с памятью. Объём памяти уменьшается.

В 2-м случае применяются специальные команды и специальные стробы обмена по магистрали. Память не сокращается.

### Устройства ввода/вывода:

1. Обмен информацией осуществляется по тем же принципам, что и с памятью.
2. Устройства ввода/вывода взаимодействуют не только процессором, но и с внешними устройствами: цифровыми, аналоговыми.

### Структура устройств ввода/вывода (рис 3.4):

1. **Схема управления** – для обработки стробов обмена.
2. **Входной порт** (имеется в виду, что он «входной» для устройства ввода/вывода) – параллельный регистр, в который процессор записывает информацию.
3. **Выходной порт** – однонаправленный буфер, через который процессор читает информацию.

Входной порт и выходной могут комбинироваться в один вход/выход.

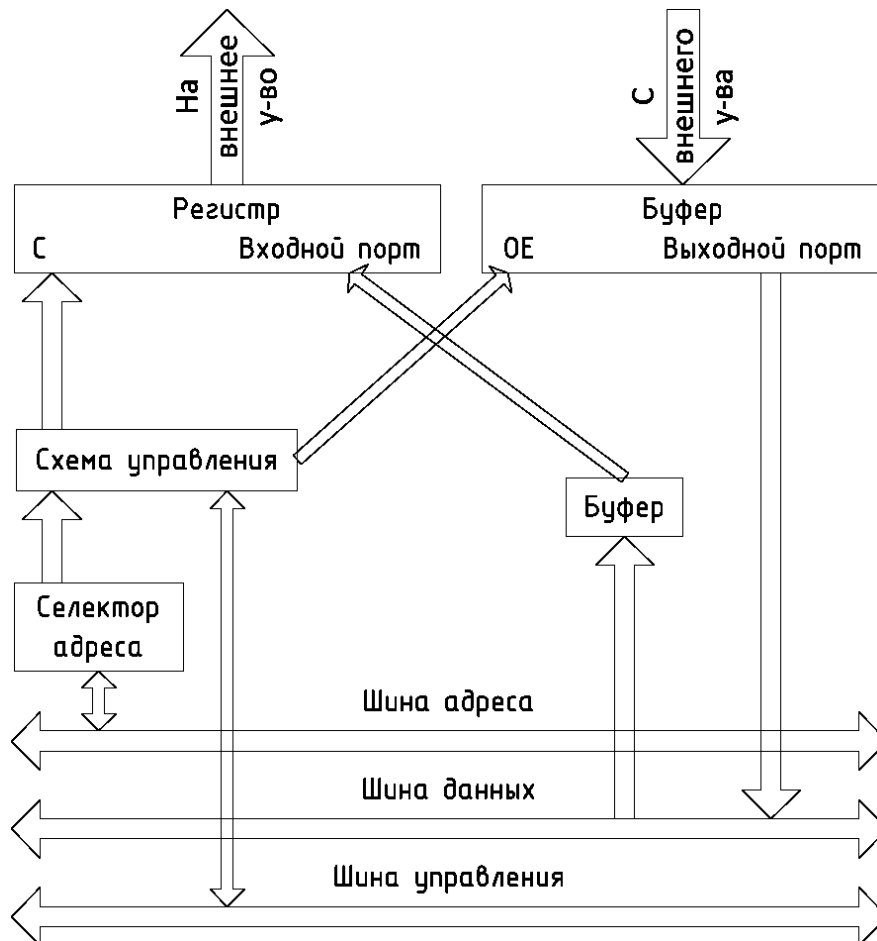
4. **Селектор адреса** распознаёт адрес со стороны магистрали.
5. **Схема управления** выдаёт внутренние стробы обмена в ответ на магистральные стробы обмена.
6. **Буфер** осуществляет согласование шины данных с устройством.

Из шины данных данные записываются в регистр по сигналу **С** и выдаются на внешнее устройство. Сигнал **ОЕ** – разрешение на чтение с внешнего устройства.



### Группы устройств ввода/вывода:

1. Устройства *интерфейса пользователя*.
2. Устройства ввода/вывода для *длительного хранения информации*.
3. *Таймерные* устройства.
4. Устройства для *подключения к информационным сетям*.



**Рисунок 3.4.** Структура устройств ввода/вывода

### **Вопросы**

1. Какие выводы (контакты, разъёмы) могут быть обнаружены на микросхеме процессора?
2. Что происходит при включении питания процессора?
3. Зачем нужна схема управления выборкой команд?
4. Что такое конвейер команд?
5. Какие операции выполняет арифметико-логическое устройство?

6. Как можно повысить производительность процессора?
7. Как вы думаете, почему производительность процессора можно сделать выше, если уменьшить количество команд, выполняемых арифметико-логическим устройством?
8. Что такое внутренняя разрядность процессора?
9. Какие существуют подходы к назначению регистров процессора?
10. Что такое регистр признаков? Какая от него польза?
11. Каковы функции схемы управления прерываниями?
12. Каковы функции схемы управления прямым доступом к памяти?
13. Что делает схема логики управления?
14. Перечислите характеристики устройства памяти.
15. Что определяет селектор (дешифратор) адреса?
16. Что вы можете сказать о памяти программы начального запуска?
17. Что такое стек?
18. Что вы знаете об обработке прерываний?
19. Что вы можете сказать об устройствах ввода/вывода?

## **ТЕМА 4. АДРЕСАЦИЯ ОПЕРАНДОВ И РЕГИСТРЫ ПРОЦЕССОРА**

### **Лекция 6 (часть 1)**

#### **4.1. Адресация операндов**

##### **Адресация**

##### Общие моменты:

1. Система команд определяет логику работы процессора и его реакцию на те, или иные комбинации внешних событий.
2. Каждая команда, выбираемая из памяти процессором, определяет алгоритм поведения процессора на ближайшие несколько тактов.

3. Код команды говорит о том, какую операцию предстоит выполнить процессору и с какими операндами, где взять исходную информацию для выполнения команды и куда поместить результат.
4. Процессор обычно узнаёт размер команды из первого прочитанного байта.
5. Код команды расшифровывается и преобразуется в набор микроопераций, выполняемых отдельными узлами процессора.

### **Методы адресации**

Определение 1. Методы адресации – методы, с помощью которых по коду команд можно определить (задача микропроцессора), откуда брать входной операнд и куда помещать выходной.

#### Непосредственная адресация

Операнд находится в памяти непосредственно за кодом команды. Операнд обычно представляет собой константу.

#### Прямая (абсолютная) адресация

Операнд находится в памяти по адресу, код которого находится внутри программы сразу же за кодом команды.

#### Регистровая адресация

Операнд находится во внутреннем регистре процессора.

#### Косвенно-регистровая (косвенная) адресация

Во внутреннем регистре процессора находится не сам операнд, а его адрес в памяти.

#### Автоинкрементная адресация

Отличается от косвенной адресации тем, что после выполнения команды содержимое регистра увеличивается.

Данный метод даёт удобство при обработке кодов из массива данных (в этом случае содержимое регистра увеличится на размер элемента массива).

### Автодекрементная адресация

Данный тип адресации противоположен типу *автоинкрементной адресации*.

Совместное использование *автоинкрементной* и *автодекрементной* адресаций позволяет организовать память стекового типа.

### Индексные методы адресации

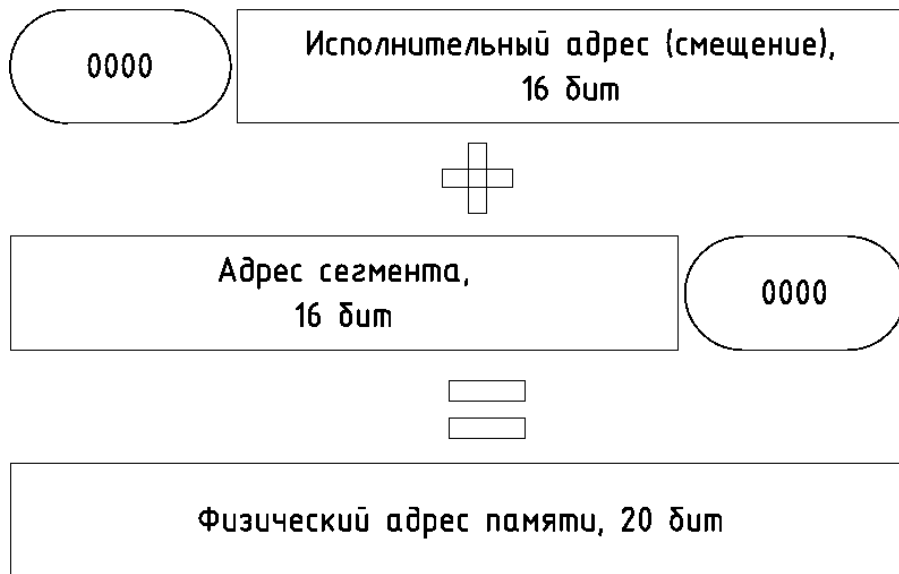
Адрес операнда вычисляется путём прибавления к содержимому регистра заданной константы (индекса).

## **4.2. Сегментирование памяти**

### **Сегментирование в Intel 8086**

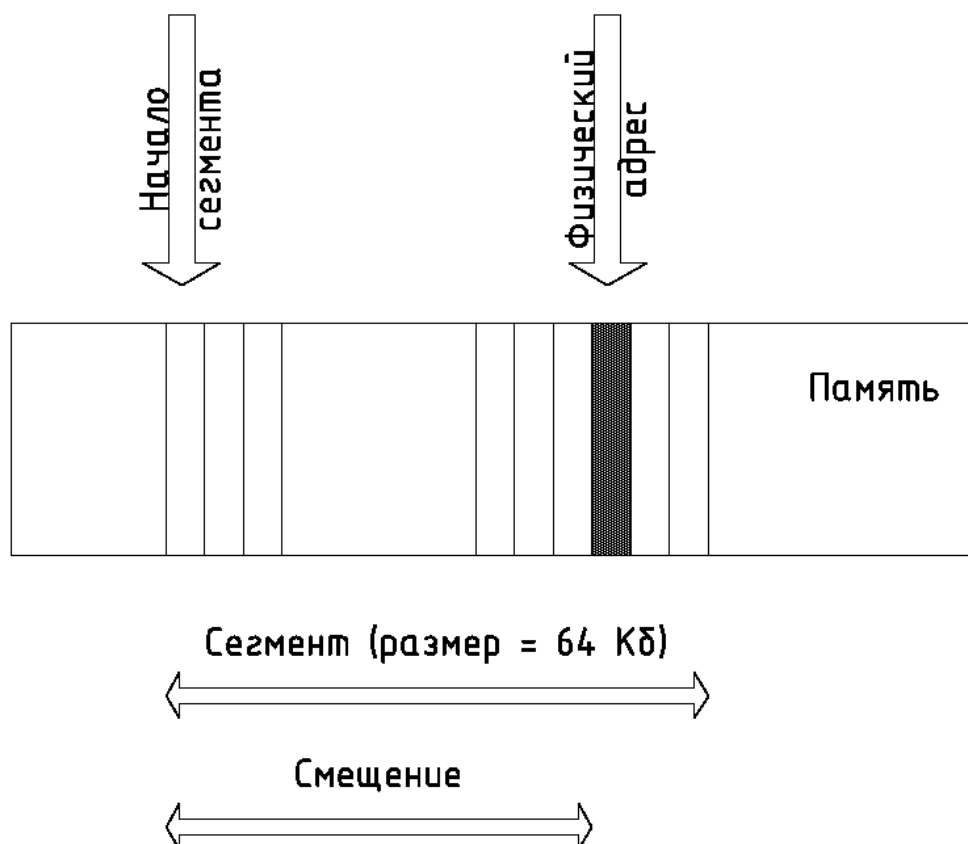
1. Память системы представлена в виде сегментов, размером по 64 Кб, их положение в пространстве памяти можно изменять программным путём.
2. Для хранения кодов адресов памяти используются пары регистров:
  - *Сегментный регистр* определяет адрес начала сегмента (положение в памяти).
  - *Регистр указателя* (смещения) определяет положение рабочего адреса внутри сегмента.
3. Шина адреса 20-разрядная. Физический адрес формируется так, как показано на рис 4.1.
4. Сегмент начинается только на 16-байтной границе памяти, т.к. адрес начала сегмента имеет 4 младших нулевых разряда (адрес кратен 16). Эти границы сегментов называются *границами параграфов*.
5. Суть введения сегментирования – внутренние регистры процессора – 16-разрядные, а физический адрес памяти – 20-разрядный (поскольку шина

адреса – 20-разрядная). Следовательно, это нужно для того, чтобы адресовать больше памяти.



**Рисунок 4.1.** Сегментирование в Intel 8086

В процессоре **Motorola MC 68000** такой проблемы не возникает, т.к. внутренние регистры (как и магистраль) – 32-разрядные.



**Рисунок 4.2.** Сегментирование

## Метод сегментирования в Intel 80286

В защищённом режиме (рис. 4.3):

1. Содержание сегментного регистра – код селектора.
2. Код селектора определяет адрес в памяти, по которому хранится дескриптор сегмента (в таблице дескрипторов).
3. Дескриптор сегмента содержит базовый адрес сегмента (24 бита, что обеспечивает адресацию 16 Мб физической памяти,  $2^{24} = 16\ 777\ 216$ ), размер сегмента (1 – 64 Кб) и его атрибуты.

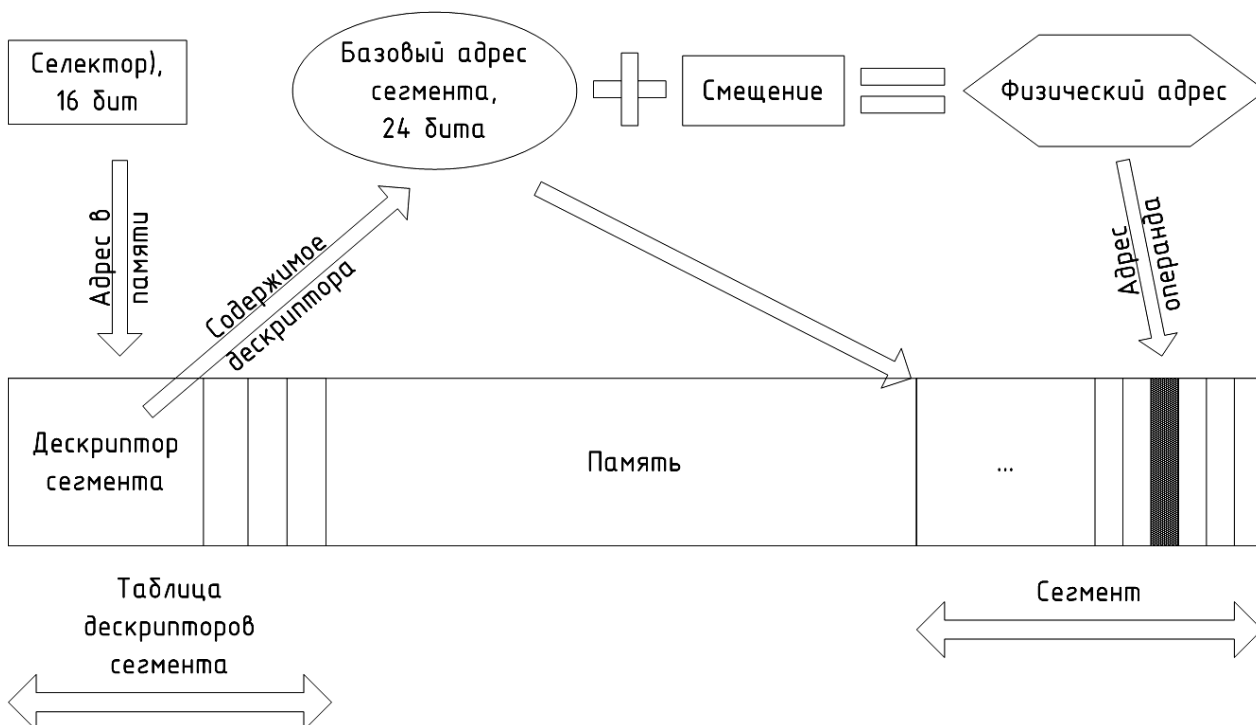


Рисунок 4.3. Сегментирование в Intel 80286

## Адресация памяти и сегментирование в Intel 80386

**1 этап** – вычисляется *эффективный адрес* (32 разряда): база (начало массива) + индекс (элемент массива) + смещение (сдвиг относительно элемента массива), в некоторых случаях индекс умножается на масштаб (1, 2, 4, 8) (в этом случае масштаб задаёт область, в рамках которой имеет смысл применять смещение).

**2 этап** – блок сегментации вычисляет *линейный адрес* (32 разряда): базовый адрес сегмента (содержание сегментного регистра) + эффективный адрес (полученный на предыдущем этапе).

**3 этап** – получение *физического адреса* (32 бита): блок страничной переадресации преобразует линейный адрес. Таким образом, линейный адрес переводится в физический, страницами по 4 байта.

### 4.3. Регистры процессора

#### Регистры процессора Intel 8086

В данной модели процессора 14 регистров, по 16 бит каждый.

#### Регистры данных:

1. **AX** (аккумулятор) – умножение, деление, обмен с устройствами ввода/вывода.
2. **BX** (база) – базовый регистр в вычислениях адреса.
3. **CX** (счётчик) – счётчик циклов.
4. **DX** (данные) – определение адреса ввода/вывода.

#### Сегментные регистры:

1. **CS** – сегмент команд.
2. **DS** – сегмент данных.
3. **ES** – дополнительный сегмент данных.
4. **SS** – сегмент стека.

Регистры-указатели (определяют смещение в пределах сегмента): SP, BP, SI, DI, IP.

1. **CS + IP** – счётчик команд процессора.
2. **SS + SP** – указатель стека.
3. **SI, DI** – используются в операциях со строками.

Регистр состояния процессора – **FLAGS**, 16 разрядов, часто используемые разряды:

1. **CF** – перенос (старшего бита).
2. **PF** – чётность (чётное количество единиц младшего байта, например **AL** регистра **AX**).
3. **AF** – дополнительный перенос (перенос из третьего бита).
4. **ZF** – нулевой результат.
5. **SF** – флаг знака.
6. **TF** – пошаговый режим (используется при отладке программ).
7. **IF** – разрешение аппаратных прерываний.
8. **DF** – флаг направления при строковых операциях.
9. **OF** – флаг переполнения (произошла установка или сброс знакового бита).

Флаги устанавливаются и очищаются в зависимости от результата предыдущей операции или специальными командами процессора.

## **Вопросы**

1. Как связана адресация и система команд?
2. Что такое методы адресации?
3. Какие вы знаете методы адресации?
4. В чём смысл непосредственной адресации? Что такое операнд? Что такое константа?
5. Почему прямая адресация называется «прямой»?
6. Что такое сегментация?
7. Что такое граница параграфа?
8. Назовите группы регистров процессора Intel 8086.
9. Что вы знаете о регистре состояния процессора Intel 8086?



## ТЕМА 5. СИСТЕМА КОМАНД ПРОЦЕССОРА

### Лекция 6 (часть 2)

#### 5.1. Команды процессора

##### Команды пересылки данных

###### Функции:

1. Загрузка (запись) содержимого во внутренние регистры процессора.
2. Сохранение в памяти содержимого внутренних регистров процессора.
3. Копирование содержимого из одной области памяти в другую.
4. Запись в устройства ввода/вывода и чтение из них.

###### Примеры команд:

1. **MOV, MOVB** – байтовая пересылка.
2. **LOAD** – для загрузки регистров.
3. **PUSH/POP** – работа со стеком.
4. **MOVS, MOVSB** – для цепочной пересылки данных.
5. **IN/OUT** – для работы с устройствами ввода/вывода.
6. **INS/OUTS** – цепочный ввод/вывод.
7. **SWAP** – обмен между двумя половинами одного регистра.

##### Арифметические команды

Операции с фиксированной запятой (работают как с числами со знаком, так и с числами без знака):

1. **ADD** – сумма двух кодов.
2. **SUB** – разность двух кодов.
3. **MUL** – произведение. **IMUL** – произведение с учётом знака.
4. **DIV** – деление. **IDIV** – деление с учётом знака.

Операции с плавающей запятой (формат числа – порядок, мантисса):  
вычисление тригонометрических, логарифмических функций.

### Прочие команды:

1. Команда очистки **CLR** – запись нулевого кода в регистр или ячейку памяти.
2. **INC** – команда инкремента.
3. **DEC** – команда декремента.
4. **CMR** – сравнение операндов – изменяет биты в регистре состояния процессора.

### **Логические команды**

Коды операндов рассматриваются как набор битов.

### Операции:

1. Логическое И (AND), логическое ИЛИ (OR), сложение по модулю 2 (XOR).
2. Логические, арифметические и циклические сдвиги.
3. Проверка битов.
4. Установка и очистка битов регистра состояния процессора.

### Примеры:

1. **AND** – принудительная очистка заданных битов.
2. **OR** – принудительная установка заданных битов.
3. **XOR** – инверсия заданных битов.
4. **TST** – проверка операнда на равенство нулю и на знак.
5. **BIT** – проверка отдельных битов.

### **Команды переходов**

#### Предназначение:

1. Организация циклов, ветвлений, вызовов подпрограмм и т.д. Нарушение последовательного хода выполнения программы.
2. Команды записывают в регистр-счётчик команд новое значение, тем самым вызывая переход процессора к определённой команде в памяти. Если предусмотрен возврат, то текущие параметры процессора сохраняются в стеке.

### Команды перехода без возврата:

1. Безусловных переходов.
2. Условных переходов.

Для проверки условий перехода применяется команда сравнения – **СМР**, предшествующая команде условного перехода.

**INT** – вызов прерываний, команда перехода с возвратом. Сохраняет содержимое регистра состояния процессора в стеке.

**IRET/RTI** – возврат из подпрограммы, вызванной командой прерывания.

### Факторы, влияющие на быстродействие процессора:

1. *Тактовая частота процессора* – скорость выполнения команд, однако разные процессоры выполняют одинаковые команды за разное количество тактов. Это зависит от сложности команды и методов адресации операндов.
2. *Структура системы команд процессора*. На одну операцию может понадобиться одна или несколько команд.
3. *Методы адресации*, наличие сегментированной памяти, способы взаимодействия с устройствами ввода/вывода.
4. *Разрядность* процессора.
5. *Объём системной памяти*.

### **Вопросы**

1. Что такое команды пересылки данных?
2. Какие вы знаете команды пересылки данных?
3. Какие вы знаете арифметические команды?
4. Какие ещё команды вы знаете?
5. Зачем нужны команды переходов?
6. Назовите факторы, влияющие на быстродействие процессора.

## ТЕМА 6. ОРГАНИЗАЦИЯ МИКРОКОНТРОЛЛЕРОВ

### Лекция 7

#### 6.1. Устройство микроконтроллера

##### Классы микроконтроллеров

Микроконтроллер объединяет в *одном кристалле*: центральный процессор, ПЗУ, ОЗУ, порты ввода/вывода, таймеры.

Классы микроконтроллеров:

1. *8-разрядные* микроконтроллеры для встраиваемых систем.
2. *16 и 32 разрядные* микроконтроллеры.
3. *Цифровые сигнальные процессоры (DSP)*.

Применимость 1-го класса – промышленность, бытовая и компьютерная техника.

Фирмы производители – Motorola, Microchip, Intel, Zilog, Atmel.

Отличительные признаки 8-разрядных микроконтроллеров:

1. Модульная организация.
2. Закрытая архитектура (отсутствие линий магистрали на выводах корпуса).
3. Использование типовых функциональных модулей: таймеры, АЦП, ... .
4. Расширение числа режимов работы периферийных модулей.

##### Модульная организация микроконтроллера

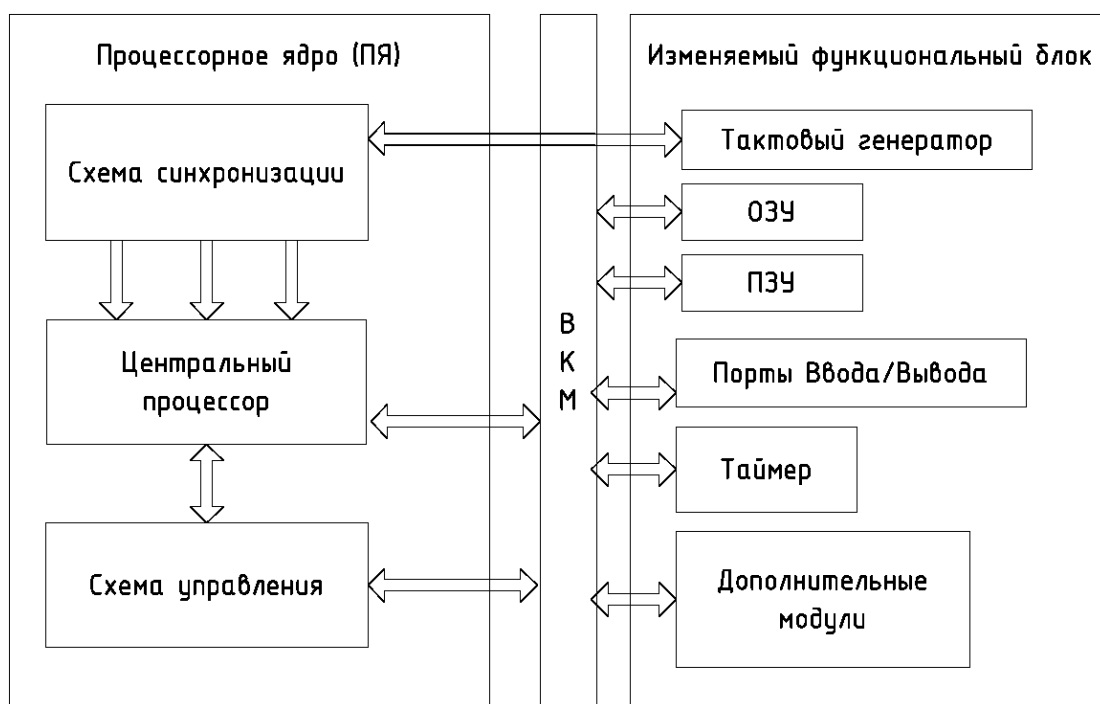
Процессорное ядро (состав):

1. Центральный процессор.
2. Внутренняя контроллерная магистраль (ВКМ) в составе шин адреса, данных, управления.
3. Схема синхронизации микроконтроллера.
4. Схема управления режимами работы.

Изменяемый функциональный блок:

1. Модули памяти.
2. Порты ввода/вывода.
3. Модули тактовых генераторов.
4. Таймеры.
5. Дополнительные модули: компаратор напряжения, АЦП.

Каждый модуль учитывает протокол ВКМ.



**Рисунок 6.1.** Организация микроконтроллера

В каждой модели одной серии микроконтроллеров используют одинаковое процессорное ядро. В разных моделях одной серии микроконтроллеров используют различные изменяемые функциональные блоки.

Производительность процессорного ядра микроконтроллера определяется:

1. Набором регистров для хранения промежуточных данных.
2. Системой команд процессора.
3. Способами адресации операндов в пространстве памяти.
4. Организацией процессов выборки и исполнения команды.

Используют один из двух принципов построения процессоров: **CISC, RISC**.

CISC (характеристики):

1. Примеры: Intel с ядром MCS-51, семейства HC05, HC08, HC11 фирмы Motorola.
2. Команды переменной длины.
3. Различные способы адресации операндов.
4. Время выполнения команды – от 1 до 12 тактов (команды могут быть различной сложности).

RISC (характеристики):

1. Примеры: AVR от Atmel, PIC16 и PIC17 фирмы Microchip.
2. Фиксированная длина команд.
3. Выборка и исполнение команд может производиться за один такт процессора.

Более сложная система команд (CISC) иногда позволяет реализовать более эффективный способ выполнения операции.

В микроконтроллере применяется одна из двух архитектур микропроцессорных систем: *принстонская* (фон Неймановская) и *гарвардская*.

*Архитектура фон Неймана* – гибкость и универсальность.

*Гарвардская архитектура* – сокращение длины команд, ускорение поиска информации в памяти данных, возможность реализации параллельных операций, недостаточная гибкость для реализации некоторых программных процедур.

## **6.2. Система команд микроконтроллера**

**Система команд**

Виды команд:

1. Арифметические и логические команды, менее развиты, чем у микропроцессора.
2. Команды переходов.

*Следующие команды более развиты, чем у микропроцессоров:*

3. Команды битового управления (разрядами регистров).
4. Команды управления ресурсами (настройка режимов работы портов ввода/вывода, управление таймером и т.п.).
5. Команды пересылки данных.

### **Память программ**

Обычно это ПЗУ для хранения команд и констант. Энергонезависимая память.

#### Типы:

1. ПЗУ *масочного типа* (mask-ROM). Содержимое ячеек заносится при изготовлении с помощью масок и не может быть, в последствие, заменено. Высокая надёжность хранения информации (программирование в заводских условиях, контроль результата).
2. ПЗУ, *программируемые пользователем, с ультрафиолетовым стиранием* (EPROM). Программируется электрическими сигналами и стирается с помощью ультрафиолетового облучения. Допускает многократное программирование.
3. ПЗУ, *программируемые пользователем, с электрическим стиранием* (EEPROM). Стирание и программирование происходит, не снимая микроконтроллер с платы. Ограниченный объём памяти.
4. ПЗУ, *однократно программируемые пользователем* (OTPROM).
5. *Флэш-память* (flash-ROM). В EEPROM стирание производится отдельно для каждой ячейки. Здесь стирание производится блоками.

### **Память данных**

На основе статического ОЗУ. Содержимое сохраняется при снижении тактовой частоты.

Определение 1. *Напряжение хранения информации* – напряжение, при котором работа программы выполняться не будет, но информация в ОЗУ сохраняется. Нужно для возможности перевода микроконтроллера на питание от автономного источника (батареи).

### **6.3. Дополнительные вопросы организации микроконтроллеров**

#### **Регистры микроконтроллера**

Регистры процессора: аккумулятор, регистры состояния, индексные регистры.

Регистры управления: управление прерыванием, таймеры.

Регистры, обеспечивающие ввод/вывод данных: регистры данных, управления параллельным, последовательным и аналоговым вводом/выводом.

В некоторых микроконтроллерах все регистры и память данных располагаются в одном адресном пространстве. В других – отдельное пространство у ввода/вывода (регистров) от общего пространства памяти (преимущество процессорам с гарвардской архитектурой, обеспечивается возможность считывать команды из памяти во время обращения к регистру ввода/вывода).

#### **Стек микроконтроллера**

В *фон Неймановской* архитектуре единая область памяти используется и для реализации стека. Производительность снижается (нет возможности доступа к различным видам памяти).

В *гарвардской архитектуре* стековые операции производятся в специально выделенной для этой цели памяти.

#### **Внешняя память**

Микроконтроллер может содержать специальные аппаратные средства для подключения внешней памяти.



Более часто используются порты ввода/вывода для подключения внешней памяти, обращение к памяти реализуется программными средствами. В этом случае происходит снижение быстродействия.

### **Порты ввода/вывода**

#### Общие моменты:

1. Линии ввода/вывода объединены в многоразрядные (чаще 8-разрядные) параллельные *порты* ввода/вывода.
2. В памяти микроконтроллера каждому порту ввода/вывода соответствует свой адрес регистра данных (поэтому такие регистры ещё называют портами).
3. Обращение к регистру данных порта ввода/вывода производится теми же командами, что и обращение к памяти данных.
4. В некоторых микроконтроллерах отдельные разряды портов могут быть опрошены или установлены специальными командами.

#### Типы параллельных портов:

1. **Однонаправленные порты** (только ввод или только вывод информации).
2. **Двунаправленные порты**. Ввод или вывод определяется в процессе инициализации микроконтроллера.
3. **Порты с альтернативной функцией** (мультиплексированные порты). Отдельные линии используются разными встроенными периферийными устройствами.

Порты производят временное согласование функционирования микроконтроллера и объекта управления.

#### Типы алгоритмов обмена информацией между микроконтроллером и внешним устройством через порты ввода/вывода:

1. Режим простого программного ввода/вывода.
2. Режим ввода/вывода со стробированием.
3. Режим ввода/вывода с полным набором сигналов подтверждения обмена.

## **Модуль прерываний микроконтроллера**

Используются те же принципы, что и в микропроцессорных системах.

### Источники внутренних запросов прерываний:

1. Переполнение таймеров/счётчиков.
2. Готовность памяти EEPROM.
3. Сигналы от дополнительных модулей микроконтроллера, включая завершение передачи или приёма информации по одному из последовательных портов.
4. Запрос прерывания поступает на обработку, если прерывания разрешены и разрешено прерывание по данному запросу.

### Аппаратные прерывания с немаскируемым характером:

1. Включение питания.
2. Подача сигнала «сброс».
3. Переполнение сторожевого таймера.

## **Режимы работы микроконтроллера**

1. **Активный режим.** Основной режим, когда доступны все ресурсы.
2. **Режим ожидания.** Центральный процессор прекращает работу, периферийные модули продолжают функционировать.
3. **Режим останова.** Работу прекращают: центральный процессор и периферийные модули.

2,3 – режимы пониженного электропотребления.

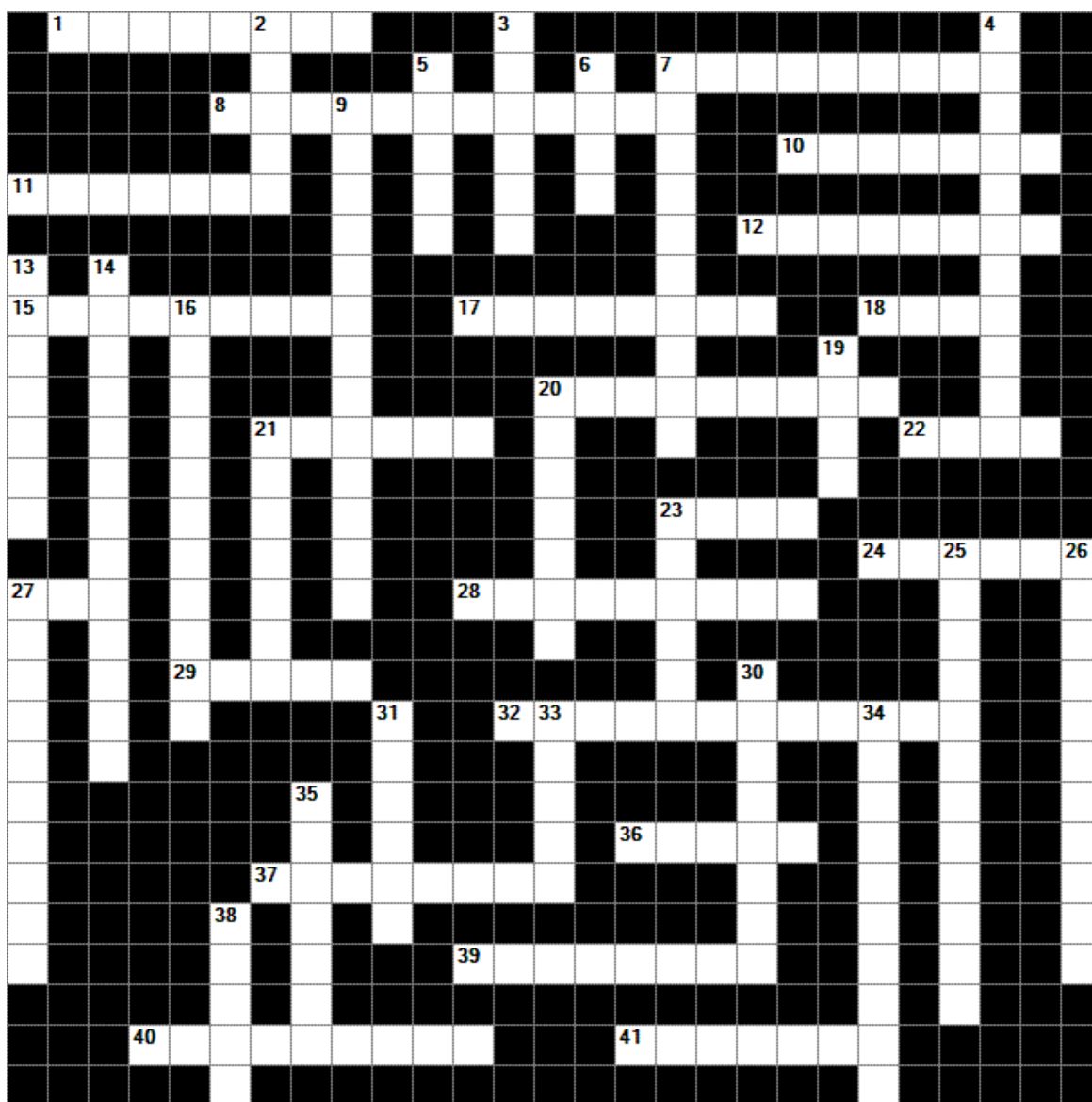
## **Вопросы**

1. Из чего состоит микроконтроллер?
2. Какие основные отличительные признаки 8-разрядных микроконтроллеров вы можете назвать?
3. Из чего состоит процессорное ядро?
4. Что такое изменяемый функциональный блок?

5. Что такое «модель» микроконтроллера? Чем она отличается от «серии»?
6. Чем определяется производительность процессорного ядра микроконтроллера?
7. Какие вы знаете принципы построения процессоров?
8. В чём отличие гарвардской (фон Неймана) архитектуры от принстонской?
9. Что такое напряжение хранения информации?
10. Какие вы знаете группы регистров микроконтроллера?
11. Что такое порт микроконтроллера и зачем он нужен? Какие бывают порты?
12. Какие могут быть источники внутренних запросов прерываний микроконтроллера?
13. Какие вы знаете режимы работы микроконтроллера?
14. Какие вы знаете режимы пониженного электропотребления?

## Лекция 8

### Промежуточный контроль



**По горизонтали:** 1. Команда XOR – \*\*\*. 7. Универсальная микропроцессорная система. 8. Выражается быстродействием, гибкостью и удобством использования. 10. Регистр CX – \*\*\* циклов. 11. CF, как флаг регистра состояния процессора указывает на \*\*\*. 12. В 16-разрядных процессорах появился \*\*\* команд, позволяющий выбирать несколько следующих команд, пока выполняется предыдущая команда. 15. Набор цифровых кодов, расшифровав которые, процессор узнаёт, что ему надо делать. 17. Шина данных определяет \*\*\* системы. 18. Регистр, в который или из которого процессор пишет/читает информацию. 20. Команда CMP – \*\*\*. 21. Выражается количеством занимаемых разрядов ячеек памяти. 22. Один из способов организации реакции на внешнее событие – метод опроса \*\*\*. Что в данном случае опрашивает процессор? 23. Элемент организации микроконтроллера 24. Требование к электронной системе в виде набора функций. 27. \*\*\* данных выставляется на шину данных. 28. Регистр AX – \*\*\*, деление, обмен с устройствами ввода/вывода. 29. \*\*\* выставляется на шину адреса. 32. Одна из функций буферных микросхем – \*\*\* протоколов обмена. 36. Цикл, когда процессор записывает информацию. 37. Особенность микропроцессорной системы

заключается в возможности изменения \*\*\* работы, не изменяя аппаратной части. **39.** Способность электронной системы подстраиваться под определённые задачи. **40.** Элемент микропроцессорной системы, который выполняет набор команд. **41.** Команда IRET/RTI – \*\*\* из подпрограммы, вызванной командой прерывания.

**По вертикали:** **2.** Сигнал шины управления, возвращающий систему в исходное состояние. **3.** Цикл, когда процессор вводит информацию. **4.** Один из отличительных признаков 8-разрядных микроконтроллеров – закрытая \*\*\* (отсутствие линий магистрали на выводах корпуса). **5.** Одна из составляющих процессора. **6.** Часть оперативной памяти для временного хранения данных. **7.** Шина данных определяет максимально возможное \*\*\* команд. **9.** Создание, активация, подготовка к работе, определение параметров. Приведение программы или устройства в состояние готовности к использованию. **13.** В случае непосредственной адресации \*\*\* находится в памяти непосредственно за кодом команды. \*\*\* обычно представляет собой константу. **14.** Сигнал шины управления, свидетельствующий о получении кода. **16.** Адресация, при которой операнд находится во внутреннем регистре процессора. **19.** \*\*\* обмена информацией – временной интервал, в течении которого происходит выполнение одной элементарной операции обмена по шине. **20.** Регистр SS – \*\*\* стека. **21.** Одна из составляющих процессора. **23.** Каждое устройство, каждая \*\*\* имеют адрес. **25.** \*\*\* цикла включает в себя несколько периодов сигнала тактовой частоты. **26.** Тип обмена, при котором процессор заканчивает операции, когда устройство-исполнитель подтверждает выполнение операции специальным сигналом. **27.** Микропроцессорная система, выполненная в виде отдельного модуля. **30.** Отсутствие аппаратной избыточности ведёт к тому, что система может работать в полную \*\*\*. **31.** Адрес начала программы обработки прерываний. **33.** Шина \*\*\* определяет допустимый объём памяти. **34.** Аппаратная избыточность приводит к увеличению стоимости и потребляемой мощности, а также снижению \*\*\* и быстродействия. **35.** Команда DIV – \*\*\*. **38.** Сигнал шины управления, сообщающий о том, что код выставлен на шину.

## ТЕМА 7. ПРОЕКТИРОВАНИЕ УСТРОЙСТВ

### Лекция 9

*Микропроцессорные системы на основе микроконтроллеров часто используются как встроенные системы для решения задач управления некоторым объектом.*

Важный аспект, принимаемый во внимание – работа в *реальном времени*.

Методология проектирования микропроцессорного устройства:

- 1. Составление технического задания.** *Формулировка требований к контроллеру. Что пользователь или заказчик хочет от контроллера? Что разрабатываемый прибор должен делать? На основании требований составляется функциональная спецификация (функции выполняемые контроллером для пользователя). Описание форматов данных.*
- 2. Разработка алгоритма управления.** *Выбор алгоритма на основе соотношения объёма программных и аппаратных средств.*
- 3. Выбор типа микроконтроллера.** *Учитываются разрядность, быстродействие, набор команд и способов адресации, объём ПЗУ и ОЗУ, наличие документации (или её полноты).*
- 4. Разработка структуры аппаратных и программных средств.** *Определяются следующие элементы. Состав имеющихся и подлежащих разработке аппаратных модулей, протоколы обмена между модулями, типы разъёмов. Вырабатывается конструкция контроллера. Определяется состав и связи программных модулей, язык программирования. Осуществляется выбор средств проектирования и отладки.*
- 5. Разработка и отладка аппаратных средств.** *Разработка общей принципиальной схемы. Разводка топологии плат. Монтаж макета и его автономная отладка.*

**6. Разработка и отладка программы.** Использование интегральной среды разработки, высокоуровневое программирование, низкоуровневое программирование. Оптимизация кода. Использование программных симуляторов.

**7. Интеграция** разработанного контроллера в систему.

**8. Испытание.**

**9. Эксплуатация.**

## **7.1. Задачи проектирования микропроцессорной системы**

### **Технико-экономический анализ и обоснование**

Применение микропроцессорной техники (далее – МТ) при проектировании устройств должно быть *технически и экономически обоснованным*.

*Техническое обоснование* – это выявление характеристик, функциональных возможностей устройства, которые значительно улучшаются или расширяются при использовании выбранных микропроцессорных средств (далее – МС). В рамках такого мероприятия должно быть показано, что достичь этих характеристик, используя другую элементную базу (*базовые матричные кристаллы, программируемые технологическим путём в заводских условиях; программируемые логические интегральные схемы*), сложно или технически нецелесообразно.

*Экономическое обоснование* предполагает определение прямой или косвенной экономии денежных средств, которая возникает при производстве либо при эксплуатации данного устройства в случае использования выбранных МС.

### **Использование микропроцессорной техники**

Характер применения МТ в проектируемых устройствах:

**1.** Использование МТ для **реализации главных управляющих блоков** в проектируемой системе, которые обеспечивают управление и координацию

работы всех остальных узлов проектируемой системы. Это разработка **микропроцессорной системы** (МПС), которая может включать несколько *микропроцессорных контроллеров*, взаимодействующих между собой, большое количество *периферийных устройств, датчиков и устройств сопряжения*. Эта система может быть распределённой иерархической и включать управляющие компьютеры верхних уровней.

2. Использование МТ для **выполнения строго ограниченных функций** в проектируемой электронной системе, в некоторых случаях вспомогательных (обеспечение ввода/вывода информации, первичной обработки информации, обслуживание различных устройств хранения информации). Это разработка так называемого **локального устройства на базе МС** в рамках проектируемой электронной системы.

Разработка подразумевает подборку и составление следующих документов (в идеальном случае):

1. **Подробное описание** назначения, работы и структурную схему объекта, для которого осуществляется проектирование системы.
2. **Анализ подобных существующих систем** с выбором способа построения.
3. **Общий алгоритм работы** проектируемой системы.
4. Технические требования и требуемые характеристики системы (техническое задание к проекту).
5. **Структурная схема** проектируемой системы (в первом пункте речь шла об описании структурной схемы объекта, для которого проектируется система, здесь же составляется структурная схема проектируемой для объекта системы).
6. **Предварительная оценка** способов и микропроцессорной базы построения системы с техническим и экономическим обоснованием.
7. **Разбиение функций** системы на программно и аппаратно реализуемые.

В некоторых случаях требуется не разработка, а **модернизация** уже существующих как микропроцессорных, так и электронных систем.



Модернизация может заключаться в следующем:

1. В **построение и добавление** дополнительного микропроцессорного устройства к уже существующей системе.
2. В **переработке** существующих устройств системы с использованием современной МТ.

Задачи в рамках модернизации системы:

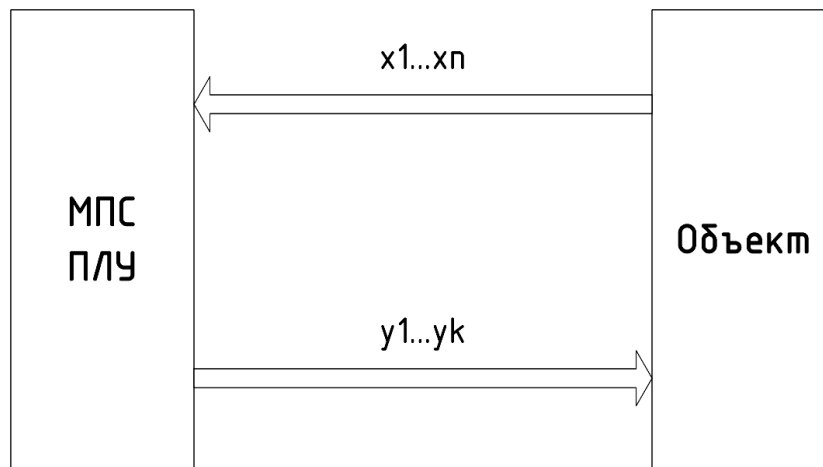
1. **Подробное описание** назначения, структуры и работы существующей системы.
2. **Выявление недостатков** системы, требующих её модернизации.
3. **Обоснование** технической и экономической целесообразности модернизации.
4. **Определение** (на основе анализа подобных систем) основных путей модернизации.
5. **Оценивание** требуемых технических характеристик проектируемого устройства, составление технического задания на проект.
6. **Детализация** (до уровня принципиальных, функциональных схем) работы и принципов построения модернизируемого блока или блоков, с которыми непосредственно взаимодействует модернизируемый блок.
7. **Составление алгоритма работы** модернизируемого блока и структурной схемы модернизированной системы.
8. **Предварительное разбиение функций** системы на программно и аппаратно реализуемые.

## **7.2. Классификация микропроцессорных устройств**

Типы микропроцессорных устройств:

1. **Системы программно-логического управления (ПЛУ).** Основные алгоритмы управления в таких системах базируются на *вычислениях логических функций и выходных значений конечных логических автоматов с памятью.*

Входные сигналы ( $x_1 \dots x_n$ ) и выходные, подаваемые на объект ( $y_1 \dots y_k$ ) носят логический характер.



**Рисунок 7.1.** Система программно-логического управления

Возможные варианты систем:

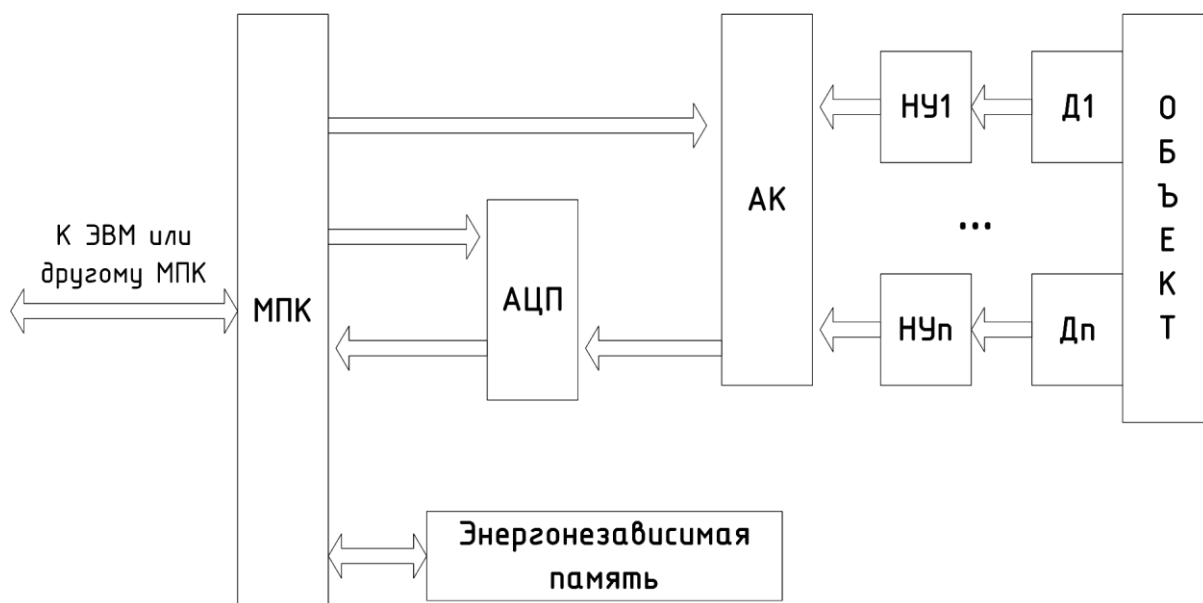
- МПС защиты силовых устройств (трансформаторов, генераторов);
- системы передачи данных с кодированием и декодированием передаваемой информации;
- автоматические телефонные системы (автодозвона, определения номера и др.);
- разветвлённые системы охранной и пожарной сигнализации;
- автоматические регистраторы срабатывания реле защит с определением и ликвидацией аварийных ситуаций;
- системы управления некоторым бытовым и промышленным оборудованием.

**2. Системы сбора и обработки информации (СОИ).** Основные функции: сбор информации с большого числа распределённых датчиков, циклический контроль измеряемых параметров, сохранение измеренных значений и первичная обработка.

Основные алгоритмы микропроцессорных контроллеров в таких системах реализуют *обработку сигналов* на основе: *цифровой фильтрации,*

спектрального анализа, группировки и распознавания сигналов, идентификации процессов, сжатия и архивации данных, реализации протоколов передачи данных и другие первичные преобразования. Такие системы могут использовать распределённые базы данных для хранения накопленной информации.

**Основные элементы:** микропроцессорный контроллер (МПК), аналогово-цифровой преобразователь (АЦП), аналоговый коммутатор (АК), нормирующие усилители сигналов (НУ) с датчиков (Д).



**Рисунок 7.2.** Система сбора и обработки информации

При проектировании таких систем необходимо изначально чётко определить функции, выполняемые каждым контроллером и компьютером в системе. Эти системы могут предполагать подробную разработку информационного обеспечения в форме различных распределённых баз данных, моделей данных и т.д.

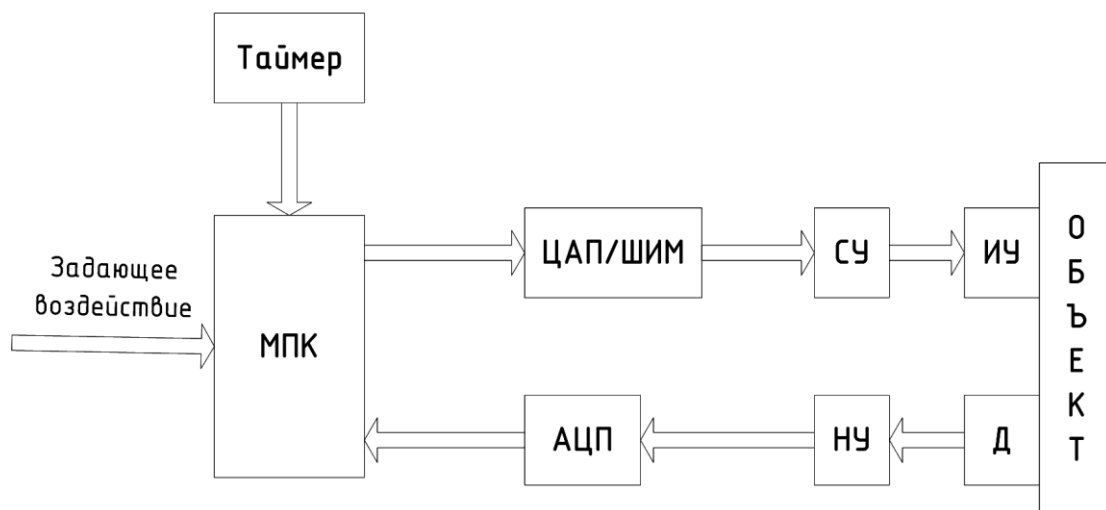
Варианты систем:

- системы контроля потребления энергии и других ресурсов;
- МПС мониторинга окружающей среды;
- МПС медицинской диагностики;

- системы технической диагностики и поиска неисправностей;
- системы контроля параметров сложных технических объектов;
- технологические системы учёта количества и проверки качества выпускаемой продукции;
- тренажёрные системы;
- системы на основе глобальных и локальных вычислительных сетей.

**3. Системы цифрового автоматического управления (ЦАУ).** Основное назначение – *реализация управления*, путём воздействия на объект, в соответствии с законом управления на основе измеренных и вычисленных параметров. Алгоритмы этих систем реализуют *функции регуляторов, наблюдателей состояния, идентификации объектов или процессов*.

Сигналы управления ЦАУ формируются устройствами типа цифро-аналогового преобразователя (ЦАП), широтно-импульсного преобразователя (ШИМ). Важную роль в ЦАУ играет таймер, задающий интервалы измерения сигналов и выдачи управляющих воздействий.



**Рисунок 7.3.** Система цифрового автоматического управления

**НУ** – нормирующее устройство.

**СУ** – согласующее устройство.

**ИУ** – исполнительное устройство.

**Д** – датчик.

Имеют широкое распространение в технике и на производстве. Им свойственны более сложное алгоритмическое и программное обеспечение. Характеризуются различными показателями качества управления.

Теоретическая база проектирования – теория цифровых систем управления.

#### Варианты систем:

- МПС управления приводами транспортных средств;
- системы программного управления приводами станков с ЧПУ (числовым программным управлением);
- системы управления технологическими процессами в электроэнергетике, химической, пищевой или лёгкой промышленности;
- локальные системы управления параметрами различных технических, транспортных или технологических объектов.

Разработку каждой из перечисленных систем можно рассматривать как триединую задачу:

- 1. Разработка математических алгоритмов** функционирования, на основе расчёта и проектирования регуляторов, фильтров, логических функций или автоматов, наблюдателей, систем распознавания и идентификации и т.п.
- 2. Разработку аппаратных средств** системы.
- 3. Разработку программного и информационного обеспечения.** Разработка информационного обеспечения осуществляется в случае реализации в системе информационной базы данных, зачастую распределённой.

#### **Вопросы**

- 1.** Что включает в себя составление технического задания?
- 2.** Что учитывается при выборе типа микроконтроллера?
- 3.** Что такое технико-экономическое обоснование?
- 4.** Каков характер применения микропроцессорной техники в проектируемых микропроцессорных системах?

5. Какие документы необходимы для поддержания качественной инженерной дисциплины проектирования микропроцессорных систем?
6. В чём заключается модернизация электронной системы?
7. Какие задачи необходимо решить в рамках модернизации электронной системы?
8. Перечислите типы микропроцессорных устройств.
9. Приведите примеры программно-логических микропроцессорных систем.
10. Каковы функции систем сбора и обработки информации?
11. Что такое системы цифрового автоматического управления?

## **Лекция 10**

### **7.3. Этапы разработки и модернизации**

#### Основные этапы:

1. **Системный анализ** объекта контроля и проектируемой МПС.
2. **Выбор типа** используемого микропроцессора, однокристального микроконтроллера (в котором есть микропроцессор) и определение структуры микропроцессорного контроллера.
3. **Разработка аппаратных средств** МПС.
4. **Разработка программного и информационного обеспечения** МПС.
5. Совместная **отладка** программного обеспечения и аппаратных средств.
6. **Оценка характеристик** построенной системы.

#### **Системный анализ**

#### Составляющие:

1. **Описание работы** объекта контроля или управления.
  - Описывается назначение объекта.
  - На основе построенной структурной схемы объекта составляется словесный алгоритм работы объекта.
  - Приводятся основные технические характеристики объекта.

- Определяются основные цели и задачи, решаемые проектируемой МПС.

## **2. Анализ вариантов построения системы.**

- Патентный поиск, обзор периодической и иной литературы.
- Определение положительных и отрицательных сторон каждого из перечисленных вариантов реализаций и разработка своего варианта построения системы.
- Определяется требуемое количество контроллеров и компьютеров в системе, а также структура их соединения (радиальная, магистральная, смешанная) и характер взаимодействия (с централизованным или децентрализованным управлением, иерархическая или нет, связанная или несвязная).
- Определяются количество и диапазон изменения величин контролируемых сигналов с объекта, а также сигналов управления объектом.

## **3. Разработка технического задания (ТЗ) на проект и обобщённой структурной схемы МПС.**

Дается краткое экономическое и техническое обоснование предложенного варианта.

ТЗ определяет:

- назначение;
- выполняемые функции;
- структуру разрабатываемой системы;
- основные технические параметры;
- элементную базу;
- условия эксплуатации;
- параметры надёжности работы устройства.

## **4. Осуществление математического описания системы и математическое моделирование работы.**

- Математическое описание используемого в системе алгоритма.

- Идентификация объекта, т.е. построение математической модели объекта.
- Расчёт требуемых для реализации конкретного алгоритма параметров.
- Определение характеристик, форматов данных для хранения и преобразования информации.

## 5. Составление общего алгоритма функционирования системы.

После определения состава реализуемых математических алгоритмов разрабатывается *общий (укрупнённый) алгоритм работы* МПС, он может состоять из отдельных алгоритмов работы каждого контроллера или компьютера в системе.

Осуществляются построения: **общей схемы работы системы, общей схемы программ** или **схем взаимодействия программ**. Здесь же определяются **требования к микропроцессорным контроллерам**: а) по точности и скорости АЦП и ЦАП; б) по скорости расчёта регулирующего воздействия (что связано с частотой дискретизации сигналов в системе) или по времени реакции на изменение входной величины (для логического регулятора); в) по объёму хранимой предыстории при управлении объектом, объёму иных хранимых параметров, что определяет требуемый объём памяти; г) по скорости передачи информации другим контроллерам или компьютерам и др.

Формулируются требования к датчикам и исполнительным устройствам системы, нормирующим усилителям, согласующим устройствам. Эти требования включают диапазон изменения и иные временные, частотные параметры входных и выходных величин, максимальную допустимую погрешность преобразования, требования гальванической развязки и др.

Если необходимо, осуществляется выбор того или иного *принципа формирования управляющего воздействия* – *модуляции* (амплитудно-импульсная – АИМ, широтно-импульсная – ШИМ, частотно-импульсная – ЧИМ).



Определяются основные параметры устройств формирования управляющих воздействий.

**6. Разбиение функций** системы на программно и аппаратно реализуемые. Большинство функций могут быть реализованы либо программным, либо аппаратным способом. В первом случае требуется минимальное количество дополнительных аппаратных устройств, но скорость выполнения функции может быть низкой. Во втором случае требуется использование специальных аппаратных средств, реализующих функцию, но при этом скорость выполнения функции выше, чем при программной реализации. Например, реализация арифметических операций в формате с плавающей точкой в системе может реализовываться либо в виде специальных подпрограмм, либо с использованием специального устройства – арифметического сопроцессора или специализированного устройства, реализованного на основе программируемых логических интегральных схем (ПЛИС).

Для реализации быстрых алгоритмов фильтрации, преобразования Фурье, вычисления свёрток, разностных уравнений могут применяться специализированные *сигнальные процессоры* (DSP), что обеспечит при необходимости высокую скорость обработки измеренной информации.

**7. Определение требований** к микропроцессорным контроллерам, компьютерам и другим устройствам системы.

**8. Построение структурной схемы** системы.

#### **Выбор типов используемых средств**

Основа любой МПС – *микропроцессорные контроллеры* (МПК). МПК реализуется на основе *однокристалльных микроконтроллеров* (ОМК) или *микропроцессоров* (ОМП).

Характеристики, влияющие на выбор ОКМ и ОМП:

- архитектура и разрядность процессора;
- быстродействие;

- объём адресуемой памяти (ОЗУ и ПЗУ);
- особенности системы команд.

Процессоры ОМК реализуют с использованием следующих архитектур:

1. **RISC** (Reduced Instruction Set Computer) – с сокращённым набором команд.
2. **CISC** (Complex Instruction Set Computer) – с традиционной расширенной системой команд.
3. **ARM** (Advanced RISC-machine) – расширенная RISC-архитектура.

RISC-архитектура нацелена на реализацию максимальной производительности процессора.

Основные черты RISC:

1. Минимальное число команд процессора (первоначально – несколько десятков, современная тенденция – несколько сотен).
2. Число регистров общего назначения (РОН) процессора максимально (приблизительно несколько тысяч). В качестве РОН используются ячейки резидентной памяти данных.
3. Каждая команда выполняется за минимально возможное время, что достигается применением в процессоре устройств управления (УУ) с «жёсткой логикой». Использование УУ ограничивает число команд из-за увеличения числа требуемых электронных блоков на кристалле.
4. Увеличена разрядность команд процессора (12, 14, 16 бит и более) по сравнению с разрядностью обрабатываемых данных. Это позволяет реализовать эффективную систему 1 – 2 словных команд, быстро выбираемых из памяти и быстро выполняемых в УУ процессора.

ARM-архитектура используется в современных 32-разрядных ОМК. В ОМК реализуется 2 системы команд: **стандартная 32-разрядная** (ARM-система) и **16-разрядная** (THUMB-система). THUMB-команды в два раза плотнее размещаются в памяти, но в то же время могут работать с 32-разрядными операндами, как и 32-разрядные ARM-команды. Использование THUMB-

команд позволяет в два раза экономнее расходовать память при сохранении или увеличении производительности.

При реализации ОМК используют две основные архитектуры памяти:

1. **Гарвардская** (увеличение скорости обмена данными и командами).
2. **Фон Неймановская** (упрощение системы команд и уменьшение числа требуемых команд, т.к. одни и те же команды используются как для обращения к памяти, так и к устройствам ввода/вывода).

Для реализации преимуществ как гарвардской, так и фон Неймановской архитектур в ряде контроллеров (**PIC16f87x** фирмы Microchip, **AVR**-микроконтроллеры фирмы Atmel) с гарвардской архитектурой реализуется оперативно перепрограммируемая непосредственно во время работы контроллера flash-память программ. Тем самым в гарвардской архитектуре реализуются загружаемые во время работы программы. Недостаток – время записи во flash память выше, чем в оперативную память при фон Неймановской архитектуре. Данную архитектуру иногда называют *гарвардской с Flash CMOS*. Кроме flash-памяти программ в контроллерах с данной архитектурой зачастую реализуют электрически перепрограммируемую память данных (EEPROM) объемом несколько сотен байт.

Немаловажную роль при выборе ОМК и ОМП играют:

1. Номенклатура встроенных периферийных устройств (таймеров, последовательных интерфейсов, параллельных портов ввода/вывода, АЦП, модулей ШИМ, модулей обработки прерываний и др.).
2. Возможность режимов пониженного электропотребления, характеристики питающего напряжения.
3. В некоторых случаях промышленного применения контроллеров важны: диапазон рабочих температур, потребляемая мощность.

Классификация ОМК:

**1. 8-разрядные периферийные ОМК.** Примеры: **РІС**-контроллеры от Microchip, контроллеры семейства **UPI-42** от Intel, контроллеры **AVR** от Atmel.

Особенности:

- гарвардская архитектура памяти;
- RISC-архитектура процессора;
- небольшой объём адресуемой памяти – до 10-100 тысяч слов;
- 8-разрядный процессор и система команд, не нацеленная на решение сложных вычислительных задач;
- не предусмотрена возможность аппаратного формирования сигналов внешней системной магистрали;
- малые габаритные размеры и электропотребление (КМОП (CMOS) технология).

Используются для построения:

- простых систем логического управления;
- небольших локальных систем автоматического управления, не требующих высокой скорости и точности;
- иногда используются для построения «интеллектуальных» датчиков систем сбора и обработки информации;
- как вспомогательные процессоры управления вводом/выводом в контроллерах высокой и средней производительности.

**2. 8-разрядные универсальные ОМК.** Примеры: **MCS-48, MCS-51/151/251** от Intel и Atmel, контроллеры семейств **HC5-HC11** от Motorola, **Z8** от Zilog. Используются для построения локальных МПК систем ПЛУ, СОИ, ЦАУ.

Характеристики:

- достаточная производительность;
- простота системы команд;
- простота внутренней архитектуры;

- большая номенклатура встроенных в ОМК дополнительных устройств – причина широкого распространения в настоящее время.

**3. 16-разрядные универсальные ОМК.** Примеры: **MCS-96/196/296** от Intel, **C16X** от Infineon Technologies, **HC12** и **HC16** от Motorola.

Это высокопроизводительные микроконтроллеры, предназначенные для использования в различных системах реального времени: управления, сбора и обработки информации, системах связи, обработки речи и изображений, где требуется высокая скорость реакции на внешние события.

Характеристики:

- система команд оптимизирована по быстродействию (RISC-архитектура) либо ориентирована на быструю обработку сигналов;
- широкая номенклатура встроенных устройств (процессор событий, коммуникационный процессор, модули быстрой обработки прерываний) позволяет строить на его базе высокоэффективные распределённые микропроцессорные системы.

**4. 32-разрядные ОМК.** Примеры: контроллеры семейства **ARM** на основе расширенной RISC архитектуры – **THUMB** с экономией памяти **ARM7DTMI** от Atmel и Intel, контроллеры на основе процессорных ядер **CPU32** и **ColdFire** от Freescale, **PowerPC** от Motorola.

Используются в специальных системах, требующих достаточно высокой производительности, они реализуют команды сигнальной обработки.

Быстродействие: 15-16 MIPS (тактовая частота до 350 МГц).

**5. Контроллеры (процессоры) цифровой обработки сигналов,** к которым относятся процессоры семейства **TMS320C3X/4X** от Texas Instruments, **SHARC** от Analog Device, **K1813BE1** из России, семейство **MIPS** контроллеров (Mobile Internet Phone System), предназначенных для мобильных систем Интернет-телефонии.

Характеризуются либо высокой скоростью выполнения операций с плавающей точкой, либо системой команд, ориентированной на выполнение операций быстрого преобразования Фурье, фильтрации сигналов.

Используются как вспомогательные процессоры в эффективных системах управления реального времени. Имеют разрядность АЛУ 16-32 бита и производительность до 120 MFLOPS (миллионов операций с плавающей точкой в секунду).

ОМП, применяемые в МПК – 16, 32 или 64 разрядные, в основном IBM-используемые (**I80X86, Pentium II, III, IV, K1810BM86**), либо 16-64 разрядные архитектуры фирмы Motorola или простейшие 8-разрядные (**I8085, K1821BM85, Z80, MC68XX**) фирм Intel, Motorola, Zilog.

## Вопросы

1. Перечислите основные этапы разработки и модернизации электронных систем.
2. Какие работы предпринимаются на этапе системного анализа?
3. Какие характеристики влияют на выбор микроконтроллеров и микропроцессоров?
4. Дайте классификацию однокристальных микроконтроллеров.

## Лекция 11

*Правильность выбора* ОМП или ОМК для МПК по быстродействию в простейшем случае может быть оценена путём написания пробного фрагмента программы. Для этого выбирается наиболее критичная ко времени выполнения программно реализуемая функция системы. Эта функция кодируется на языке ассемблера выбранного ОМК (ОМП) и затем подсчитывается время её выполнения как суммарное время выполнения всех команд фрагмента. Если полученное время меньше либо равно максимально возможному, то выбор осуществлён верно. Иначе требуется либо применять другой ОМК (ОМП), либо

данную функцию реализовать аппаратно. Однако, нужно помнить, что применение высокопроизводительных ОМК (ОМП) увеличивает стоимость системы.

**Определение времени** выполнения фрагмента программы удобно осуществлять, используя инструментальные средства разработки ПО МПК – *программы-симулятора* (программно-логические модели).

**Структура МПК** выбирается или разрабатывается на основе выбранного ОМК (ОМП) исходя из требуемого количества каналов аналогового и дискретного ввода/вывода, объёма требуемой памяти, требуемой номенклатуры периферийных устройств (таймеров, последовательных интерфейсов и т.д.). При этом важен выбор типа системной магистрали контроллера (**ISA, PCI, VME, MultiBus II, I<sup>2</sup>C**).

МПК, как правило, имеет магистрально-модульную структуру, т.е. имеется общая системная магистраль (**СМ**), к которой подключаются основные модули контроллера. Помимо системных могут быть и дополнительные – локальные магистрали и магистрали расширения.

#### Упрощённая структурная схема МПК

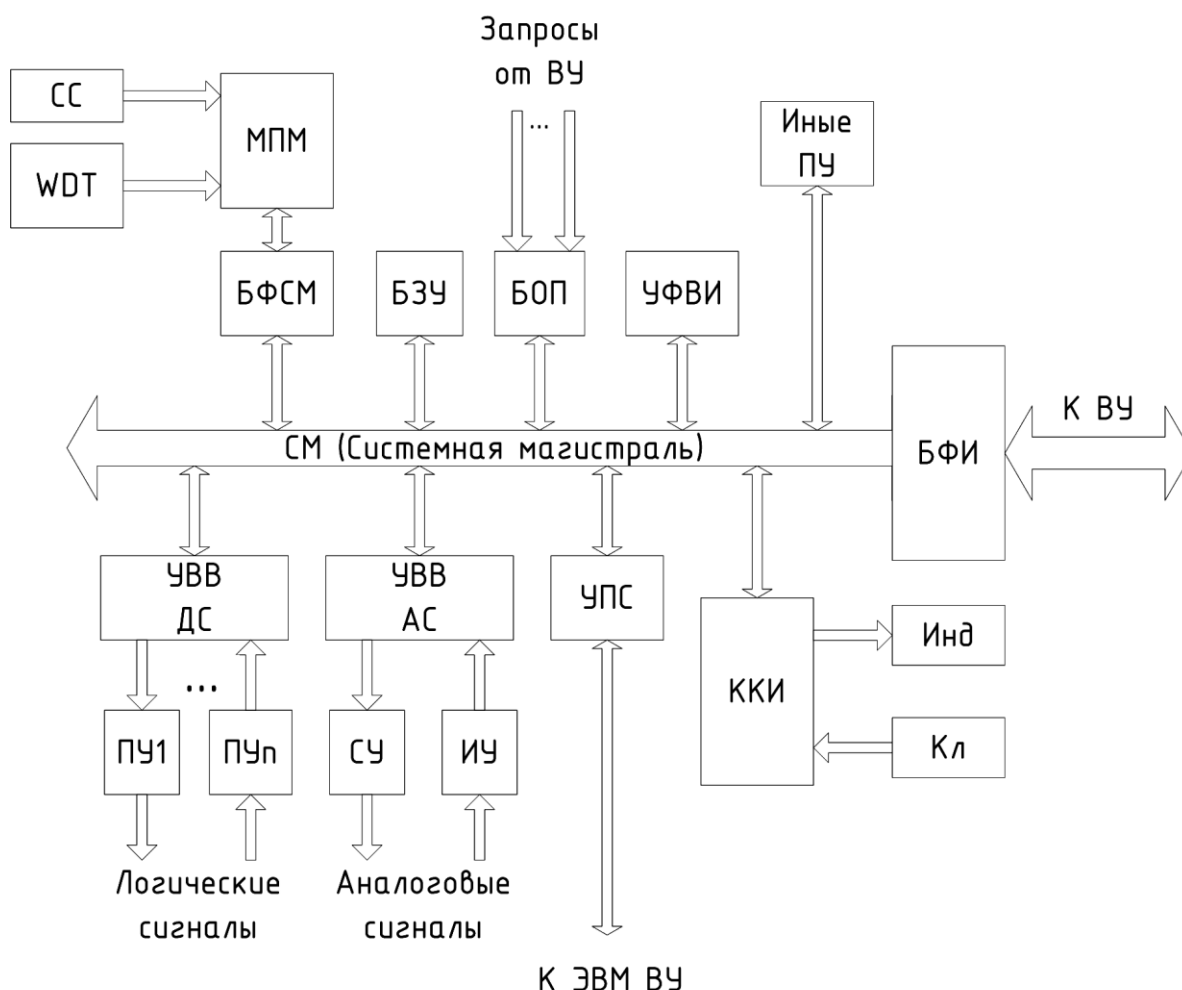
**МПМ** – микропроцессорный модуль, реализуемый на основе ОМП или ОМК.

**СС** – схема сбора и синхронизации работы МПМ, включает генератор тактовой частоты.

**БЗУ** – блок запоминающих устройств, включающий ОЗУ и ПЗУ МПК, в том числе память типа FLASH, EEPROM, «твёрдые диски».

**БОП** – блок обработки запросов прерываний от различных устройств МПС, включает контроллеры по обработке прерываний, процессоры событий, серверы периферийных транзакций и т.п.

**УВФИ** – устройство формирования и измерения временных интервалов, включает: таймеры, модули захват-сравнение, используемые для подсчёта внешних импульсов, измерения временных интервалов.



**Рисунок 7.4.** Упрощённая структурная схема контроллера

**WDT** (watch dog timer) – сторожевой таймер, исключающий ошибочные закливания ПО МК.

**УВВ ДС** – устройства ввода/вывода дискретных логических сигналов (порты ввода/вывода), при этом сигналы при вводе или выводе могут преобразовываться к тому или иному логическому уровню (ТТЛ, КМОП) с помощью преобразователей уровней (ПУ).



**УВВ АС** – устройства ввода/вывода аналоговых сигналов, при этом устройства ввода реализуются на основе АЦП, а вывода – ЦАП или ШИМ, вводимые сигналы усиливаются (ослабляются) до требуемого уровня с помощью нормирующих устройств (НУ), а выводимые – с помощью согласующих устройств (СУ).

**УПС** – устройства последовательной связи с компьютером верхнего уровня (ЭВМ ВУ) или другими МПК, представляют собой последовательные интерфейсы или промышленные вычислительные сети, с помощью которых по 2-4 проводной линии осуществляется взаимодействие между контроллерами и компьютерами верхнего уровня.

**ККИ** – контроллеры клавиатуры и индикаторов (жидкокристаллические, светодиодные и т.п.), обеспечивающие работу клавиатуры и индикаторов в составе МПК.

**БФСМ** – блок формирования внешней системной магистрали – сигналов шин адреса, данных, управления.

**БФИ** – блок формирования интерфейса с внешними системными устройствами, обеспечивает усиление сигналов системной магистрали при выходе их на разъём.

МПК называется с *открытой архитектурой*, если есть возможность доступа к его системной магистрали (СМ) через разъём. Иначе архитектура – *закрытая*.

При разработке или выборе МПК для проектируемой системы следует придерживаться следующей последовательности действий:

1. Подобрать промышленный микропроцессорный контроллер (ПМПК), удовлетворяющий техническим и экономическим требованиям, реализованный на основе выбранного ОМП (ОМК).
2. Реализовать микропроцессорный контроллер самостоятельно на основе ОМК, если подобрать ПМПК не удалось.

3. Реализовать микропроцессорный контроллер самостоятельно на основе ОМП, если это выгодно с экономической или технической точки зрения.
4. При выборе промышленного МПК немаловажную роль играет наличие развитого инструментального обеспечения по его программированию: компиляторов с языков высокого уровня, программ отладчиков, операционных систем реального времени, инструментальных сред разработки программного обеспечения с использованием CASE-средств и т.п.

### **Разработка аппаратных средств**

Начинается с *построения функциональной схемы* микропроцессорной системы или разрабатываемых блоков. При этом каждый из блоков разбивается на субблоки. Предварительно определяется *элементная база, тип интегральных схем* для реализации каждого субблока.

*Разрабатывается карта распределения памяти* контроллера и *карта распределения устройств ввода/вывода* в адресном пространстве с указанием адресов и назначения основных используемых ячеек и областей памяти. Оценивается требуемый объём ОЗУ и ПЗУ контроллера.

*Разработка принципиальных схем* начинается с определения элементной базы реализации каждого блока системы. Осуществляется выбор и обоснование технологии используемых интегральных схем.

### **Разработка программного обеспечения**

На основе общего алгоритма работы (схемы программы) микропроцессорной системы *разрабатываются алгоритмы или схемы программ* работы каждого контроллера и компьютера верхнего уровня, если такой есть в системе, в соответствии с выполняемыми ими функциями.

*Определяется использование стандартных функций* той или иной операционной системы или операционной системы реального времени (ОС РВ),

функций программы «монитор контроллера», функций библиотек стандартных подпрограмм.

Использование ОС РВ в контроллере или компьютере позволяет регламентировать время реакции контроллера на те или иные внешние события, осуществлять параллельное слежение за несколькими событиями одновременно в строго определённых временных рамках (планировщик реального времени). В настоящее время используются следующие ОС РВ: **RTX, OS-9, LynxOS, QNX** и т.д.

При разработке алгоритмов осуществляется разбивка на *подпрограммы, подпрограммы обработки прерываний*.

Осуществляется выбор методов программной реализации основных функций:

1. Расчёта разностных уравнений (передаточных функций).
2. Расчёта логических функций и выходных значений конечных логических автоматов.
3. Реализация арифметических функций с плавающей точкой (операции умножения, деления).
4. Реализации основных математических функций (синус, косинус, радикал и т.д.).
5. Реализации быстрого преобразования Фурье.

Могут *разрабатываться схема программ и схема взаимодействия программ* системы, на которой отображаются все основные составные части программного обеспечения и их размещение на контроллерах и компьютерах системы.

*Разрабатывается таблица распределения основных переменных и констант в памяти* контроллера с указанием имени, размещения в памяти и их назначения.

Если в системе реализуются взаимодействия между контроллерами и компьютерами верхнего уровня, то в проекте может *разрабатываться протокол взаимодействия*, форматы передаваемых сообщений (кадры). При этом могут использоваться модификации стандартных протоколов: **HDLC**, **SDLC**, **Profibus-DP**, **CAN**, **CSMA/CD**, **TCP/IP**.

*Определяется язык программирования, среда разработки* (**ISaGRAF** для систем от фирм PEP и Motorola, **UltraLogic** для систем от Octagon Systems, **HI+** для систем от Motorola, **Studio-96** для Intel, ...).

Ряд сред разработки (**ISaGRAF**, **UltraLogic**, **Genesis**) поддерживают языки визуального графического программирования: язык последовательных логических схем, язык функциональных блоковых диаграмм, язык релейных диаграмм и т.д. в этом случае написание программы заменяется составлением той или иной графической схемы, которая затем автоматически компилируется в язык высокого уровня или коды требуемого микропроцессорного устройства.

*Правильность разработанного программного обеспечения проверяется* с использованием программ-симуляторов, при этом осуществляется контрольный расчёт с использованием конкретных числовых данных. Полученный результат сравнивается с эталонным. Эталонный результат можно получить на профессиональных моделирующих программах для систем. После сравнения делается *вывод о работоспособности*. Осуществляется *оценка времени выполнения* основных запрограммированных функций.

Программное обеспечение компьютеров верхнего уровня системы разрабатывается с использованием языков высокого уровня или языков визуального программирования в среде ОС типа **Windows** или **OS/2**. При этом реализуются функции распределённых СУБД, алгоритмы экспертных систем и другие требуемые функции управления и контроля.

## **Совместная отладка**

1. Выбираются приборы.
2. Выявляются неисправности.
3. Разрабатываются инструкции по наладке и устранению выявленных неисправностей.

Отладка программ может осуществляться на макетном образце системы или с использованием оценочных модулей, имитирующих работу системы:

1. Программа записывается в оценочный модуль или макет.
2. Осуществляется измерение характеристик системы и сравнение с требуемыми, при этом осуществляется имитация внешних сигналов с объекта.
3. Результаты измерения приводятся в виде временных или иных диаграмм.
4. Делаются выводы.
5. Возможна разработка тестовых программных процедур для проверки работоспособности основных узлов МПК с описанием их схем программ и принципов использования.

## **Вопросы**

1. Как в простейшем случае может быть определена правильность выбора процессора или микроконтроллера?
2. Что делать если выбор микроконтроллера не привёл к положительному результату?
3. Чего следует придерживаться при разработке или выборе микропроцессорного контроллера для проектируемой системы?
4. С чего начинается разработка аппаратных средств?
5. Как производится совместная отладка системы?

# ВОПРОСЫ ДЛЯ ПРОВЕРКИ ПО ДИСЦИПЛИНЕ

## Тема 1. Основы микропроцессорной техники

1. Основные *понятия* и *определения*: электронная система, задача, быстродействие, гибкость, избыточность, интерфейс, микропроцессорная система. Традиционная цифровая система (система на «жёсткой» логике), преимущества и недостатки, применимость.
2. *Микропроцессор*, функции, определение. *Программа, система команд* и её характеристики. Структура процессора, составляющие. Требования к разработчику микропроцессорных систем.
3. Структура *связей* между устройствами: классическая структура, *шинная структура* связей (характеристики, недостатки). Структура микропроцессорной системы. *Магистраль*. Ввод входного кода в микропроцессорную систему. Вывод выходного кода из микропроцессорной системы. Виды *обмена информацией*: программный обмен информацией, обмен по прерываниям, прямой доступ к памяти.
4. *Архитектура микропроцессорной системы* с общей шиной для данных и команд, с отдельными шинами команд и данных. *Типы микропроцессорных систем*: микроконтроллеры, контроллеры, микрокомпьютеры, компьютеры.

## Тема 2. Циклы обмена информацией по шинам микропроцессорной системы

1. *Цикл* обмена информацией. *Типы* циклов, длительность циклов. *Шины* данных, адреса, управления. Типы обмена данными.
2. Циклы *программного обмена*. Обмен по мультиплексированной асинхронной магистрали Q-bus от фирмы DEC. Цикл чтения, цикл записи, цикл чтение-модификация-запись.
3. Циклы *программного обмена*. Обмен по синхронной не мультиплексированной магистрали ISA от фирмы IBM. Цикл чтения, цикл записи.

4. Циклы *обмена по прерываниям*. Типы прерывания (векторные, радиальные). Циклы обмена по прерываниям на примерах Q-bus и ISA.
5. Циклы *обмена в режиме прямого доступа к памяти* на примерах Q-bus и ISA.

### Тема 3. Функции устройств магистрали

1. Схема *подключения процессора*, смысл буферных микросхем. Схема управления выборкой команд. Арифметико-логическое устройство. Регистры. Подходы к назначению регистров. Регистры признаков (состояния).
2. Схема *управления прерываниями*. Схема управления прямым доступом к памяти. Логика управления. *Память*. Память программы начального запуска. Стек.
3. Таблица *векторов прерывания*. Алгоритм обработки прерывания.
4. Память устройств подключенных к шине. *Устройства ввода/вывода*. Структура устройств ввода/вывода.

### Тема 4. Адресация операндов и регистры процессора.

1. *Адресация* операндов. Методы адресации. Непосредственная адресация. Прямая адресация. Регистровая адресация. Косвенная адресация. Автоинкрементная и автодекрементная адресация. Индексные методы адресации
2. *Сегментирование* памяти. Сегментирование в Intel 8086, Intel 80286, Intel 80386.
3. *Регистры* процессора Intel 8086.

### Тема 5. Система команд процессора

1. *Команды* пересылки данных. Арифметические команды. Логические команды. Команды переходов. Факторы влияющие на *быстродействие* процессора.

## Тема 6. Организация микроконтроллеров

1. **Классы** микроконтроллеров. Отличительные признаки 8-разрядных микроконтроллеров. Модульная организация микроконтроллеров. **Производительность** процессорного ядра микроконтроллера.
2. **Система команд** микроконтроллера. **Типы памяти** хранения данных: mask-ROM, EPROM, OTPROM, EEPROM, flash-ROM. Память программ. Память данных.
3. **Регистры** микроконтроллера. **Стек** микроконтроллера. **Внешняя память**. **Порты** ввода/вывода. Типы портов. **Модуль прерываний** микроконтроллера. Источники внутренних запросов прерываний. Аппаратные прерывания. **Режимы работы** микроконтроллера.

## Тема 7. Проектирование микропроцессорных устройств

1. Методологические основы проектирования микропроцессорных устройств.



## **СПИСОК ЛИТЕРАТУРЫ**

1. *Безуглов Д.А., Калиенко И.В.* Цифровые устройства и микропроцессоры. Ростов н/Д: Феникс. 2008. 468 с.
2. *Гивоне Д., Россер Р.* Микропроцессоры и микрокомпьютеры: Вводный курс. М.: Мир. 1983. 464 с.
3. *Гук М.Ю.* Аппаратные средства IBM PC. Энциклопедия. СПб.: Питер.2008. 1072 с.
4. *Гуров В.В.* Архитектура микропроцессоров. М.: Интернет-Университет Информационных Технологий; БИНОМ. 2010. 272 с.
5. *Новиков Ю.В., Скоробогатов П.К.* Основы микропроцессорной техники. М.: Интернет-Университет Информационных Технологий; БИНОМ. 2011. 357 с.
6. *Таненбаум Э.* Архитектура компьютера. СПб.: Питер. 2011. 844 с.
7. *Электроника и микропроцессорная техника / Под ред. Д.т.н., проф. В.И. Лачина.* Ростов н/Д: Феникс, 2007. 576 с.