

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ

**«БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**Кафедра Управление в технических системах**

УТВЕРЖДАЮ:

Проректор по учебной работе

\_\_\_\_\_ Е.И. Луковникова

« \_\_\_\_\_ » \_\_\_\_\_ 201\_\_ г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СИСТЕМЫ И СЕТИ**

**Б1.Б.13**

**НАПРАВЛЕНИЕ ПОДГОТОВКИ**

**27.03.04 Управление в технических системах**

**ПРОФИЛЬ ПОДГОТОВКИ**

**Управление и информатика в технических системах**

Программа академического бакалавриата

Квалификация (степень) выпускника: бакалавр

<b>1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ .....</b>	<b>3</b>
<b>2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ .....</b>	<b>3</b>
<b>3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ</b>	
3.1 Распределение объёма дисциплины по формам обучения.....	4
3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости .....	4
<b>4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ .....</b>	<b>5</b>
4.1 Распределение разделов дисциплины по видам учебных занятий .....	5
4.2 Содержание дисциплины, структурированное по разделам и темам .....	8
4.3 Лабораторные работы.....	55
4.4 Семинары / практические занятия.....	55
4.5. Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат.....	55
<b>5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>57</b>
<b>6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ.....</b>	<b>58</b>
<b>7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....</b>	<b>58</b>
<b>8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>58</b>
<b>9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....</b>	<b>58</b>
9.1. Методические указания для обучающихся по выполнению лабораторных работ .....	58
9.2. Методические указания по выполнению курсовой работы .....	63
<b>10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ .....</b>	<b>64</b>
<b>11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ .....</b>	<b>64</b>
<b>Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....</b>	<b>65</b>
<b>Приложение 2. Аннотация рабочей программы дисциплины .....</b>	<b>69</b>
<b>Приложение 3. Протокол о дополнениях и изменениях в рабочей программе .....</b>	<b>70</b>

# 1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

## Вид деятельности выпускника

Дисциплина охватывает круг вопросов, относящихся к научно-исследовательскому виду профессиональной деятельности выпускника в соответствии с компетенциями и видами деятельности, указанными в учебном плане.

## Цель дисциплины

Приобретение умений и навыков исследования проблем в своей предметной области, выбора методов и средств их решения, анализа результатов теоретических и экспериментальных исследований.

## Задачи дисциплины

Формирование способностей анализа результатов исследований, выбора методов и средств решения проблем в своей предметной области.

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
1	2	3
ОК-7	способность к самоорганизации и самообразованию	<b>знать:</b> - основные этапы развития вычислительных систем; <b>уметь:</b> - организовать взаимодействие двух персональных компьютеров, персонального компьютера и периферийного устройства; <b>владеть:</b> - навыками конфигурирования адресов стека TCP/IP.
ОПК-7	способность учитывать современные тенденции развития электроники, измерительной и вычислительной техники, информационных технологий в своей профессиональной деятельности	<b>знать:</b> - достоинства и недостатки основных типов вычислительных машин и систем; <b>уметь:</b> - решать стандартные задачи по проектированию, настройке и обслуживанию распределённых вычислительных систем; <b>владеть:</b> - методами настройки и обслуживания коммуникационных устройств логической структуризации сети.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.Б.13 Вычислительные машины, системы и сети относится к базовой части.

Дисциплина Б1.Б.7 Вычислительные машины, системы и сети базируется на знаниях, полученных при изучении дисциплины Б1.Б.12 Информационные технологии, Б1.Б.6 Физика, Б1.В.ОД.7 Информатика.

Основываясь на изучении перечисленных дисциплин, Б1.Б.7 Вычислительные машины, системы и сети представляет основу для изучения дисциплин: Б1.Б.16 Технические средства автоматизации и управления, Б1.В.ДВ.9 Микроконтроллеры и микропроцессоры в системах управления.

Такое системное междисциплинарное изучение направлено на достижение требуемого ФГОС уровня подготовки по квалификации бакалавр.

### 3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ

#### 3.1. Распределение объема дисциплины по формам обучения

Форма обучения	Курс	Семестр	Трудоемкость дисциплины в часах						Курсовая работа (проект), контрольная работа, реферат, РГР	Вид промежуточной аттестации
			Всего часов	Аудиторных часов	Лекции	Лабораторные занятия	Практические занятия	Самостоятельная работа		
1	2	3	4	5	6	7	8	9	10	11
<b>Очная</b>	2	4	108	54	18	36	-	54	КР	зачет
<b>Заочная</b>	3	-	108	18	6	-	12	86	КР	зачет
<b>Заочная (ускоренное обучение)</b>	1	-	32	9	4	-	5	23	КР	зачет
<b>Очно-заочная</b>	-	-	-	-	-	-	-	-	-	-

#### 3.2. Распределение объема дисциплины по видам учебных занятий и трудоемкости

Вид учебных занятий	Трудоемкость (час.)	в т.ч. в интерактивной, активной, инновационной формах, (час.)	Распределение по семестрам, час
			4
1	2	3	4
<b>I. Контактная работа обучающихся с преподавателем (всего)</b>	54	12	54
Лекции (Лк)	18	-	18
Лабораторные занятия (ЛЗ)	36	12	36
<b>II. Самостоятельная работа обучающихся (СР)</b>	54	-	54
Подготовка к лабораторным занятиям	44	-	44
Подготовка к зачету	10	-	10
<b>III. Промежуточная аттестация зачет</b>	+	-	+
Общая трудоемкость дисциплины 108 час.	108	-	108
3 зач. ед.	3	-	3

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1. Распределение разделов дисциплины по видам учебных занятий

- для очной формы обучения:

№ раздела и темы	Наименование раздела и тема дисциплины	Трудоемкость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоятельная работа обучающихся*
			лекции	лабораторные занятия	
1	2	3	4	6	7
<b>1.</b>	<b>Теоретические основы построения узлов ЭВМ</b>	<b>12</b>	<b>3</b>	<b>-</b>	<b>9</b>
1.1.	Физические формы представления информации	4	1	-	3
1.2.	Математические модели схем ЭВМ	4	1	-	3
1.3.	Элементы и узлы ЭВМ	4	1	-	3
<b>2.</b>	<b>Процессоры и микропроцессоры</b>	<b>48</b>	<b>3</b>	<b>36</b>	<b>9</b>
2.1.	Основные принципы построения устройств обработки цифровой информации	8	0,5	6	1,5
2.2.	Принципы организации арифметико-логических устройств	8	0,5	6	1,5
2.3.	Структура и формат команд. Кодирование команд	8	0,5	6	1,5
2.4.	Способы адресации	8	0,5	6	1,5
2.5.	Запоминающие устройства ЭВМ	8	0,5	6	1,5
2.6.	Принципы организации систем обработки прерывания программ	8	0,5	6	1,5
<b>3.</b>	<b>Организация ввода/вывода в ЭВМ</b>	<b>12</b>	<b>3</b>	<b>-</b>	<b>9</b>
3.1.	BIOS	4	1	-	3
3.2.	Системные и локальные шины	4	1	-	3
3.3.	Шины ввода/вывода	4	1	-	3
<b>4.</b>	<b>Вычислительные системы</b>	<b>12</b>	<b>3</b>	<b>-</b>	<b>9</b>
4.1.	Классификация вычислительных систем	4	1	-	3
4.2.	Многомашинные вычислительные системы	4	1	-	3
4.3.	Многопроцессорные вычислительные системы	4	1	-	3
<b>5.</b>	<b>Сети компьютеров</b>	<b>12</b>	<b>3</b>	<b>-</b>	<b>9</b>
5.1.	Основные понятия	3	1	-	2
5.2.	Локальные вычислительные сети	3	1	-	2
5.3.	Сетевой и транспортный уровни	2,5	0,5	-	2
5.4.	Структура и функции территориальных сетей	3,5	0,5	-	3
<b>6.</b>	<b>Микроконтроллеры</b>	<b>12</b>	<b>3</b>	<b>-</b>	<b>9</b>
6.1.	Определение микроконтроллера	3	0,5	-	2

6.2.	Классификация микроконтроллеров	2,5	0,5	-	2
6.3.	Тенденции в развитии современных микроконтроллеров	2,5	0,5	-	2
6.4.	Оценка быстродействия микроконтроллеров	2,5	0,5	-	2
6.5.	Энергосберегающие микроконтроллеры ведущих производителей	1,5	0,5	-	1
<b>ИТОГО</b>		<b>108</b>	<b>18</b>	<b>36</b>	<b>54</b>

**- для заочной формы обучения:**

№ раздела и темы	Наименование раздела и тема дисциплины	Трудоемкость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоятельная работа обучающихся*
			лекции	практические занятия	
1	2	3	4	6	7
<b>1.</b>	<b>Теоретические основы построения узлов ЭВМ</b>	<b>15</b>	<b>1</b>	<b>-</b>	<b>14</b>
1.1.	Физические формы представления информации	5,3	0,3	-	5
1.2.	Математические модели схем ЭВМ	5,3	0,3	-	5
1.3.	Элементы и узлы ЭВМ	4,4	0,4	-	4
<b>2.</b>	<b>Процессоры и микропроцессоры</b>	<b>27</b>	<b>1</b>	<b>12</b>	<b>14</b>
2.1.	Основные принципы построения устройств обработки цифровой информации	4,2	0,2	2	2
2.2.	Принципы организации арифметико-логических устройств	4,2	0,2	2	2
2.3.	Структура и формат команд. Кодирование команд	5,2	0,2	2	3
2.4.	Способы адресации	5,2	0,2	2	3
2.5.	Запоминающие устройства ЭВМ	4,1	0,1	2	2
2.6.	Принципы организации систем обработки прерывания программ	4,1	0,1	2	2
<b>3.</b>	<b>Организация ввода/вывода в ЭВМ</b>	<b>15</b>	<b>1</b>	<b>-</b>	<b>14</b>
3.1.	BIOS	5,3	0,3	-	5
3.2.	Системные и локальные шины	5,3	0,3	-	5
3.3.	Шины ввода/вывода	4,4	0,4	-	4
<b>4.</b>	<b>Вычислительные системы</b>	<b>15</b>	<b>1</b>	<b>-</b>	<b>14</b>
4.1.	Классификация вычислительных систем	5,3	0,3	-	5
4.2.	Многомашинные вычислительные системы	5,3	0,3	-	5
4.3.	Многопроцессорные вычислительные системы	4,4	0,4	-	4
<b>5.</b>	<b>Сети компьютеров</b>	<b>16</b>	<b>1</b>	<b>-</b>	<b>15</b>

5.1.	Основные понятия	4,25	0,25	-	4
5.2.	Локальные вычислительные сети	4,25	0,25	-	4
5.3.	Сетевой и транспортный уровни	4,25	0,25	-	4
5.4.	Структура и функции территориальных сетей	3,25	0,25	-	3
<b>6.</b>	<b>Микроконтроллеры</b>	<b>16</b>	<b>1</b>	<b>-</b>	<b>15</b>
6.1.	Определение микроконтроллера	3,2	0,2	-	3
6.2.	Классификация микроконтроллеров	3,2	0,2	-	3
6.3.	Тенденции в развитии современных микроконтроллеров	3,2	0,2	-	3
6.4.	Оценка быстродействия микроконтроллеров	3,2	0,2	-	3
6.5.	Энергосберегающие микроконтроллеры ведущих производителей	3,2	0,2	-	3
<b>ИТОГО</b>		<b>104</b>	<b>6</b>	<b>12</b>	<b>86</b>

- для заочной формы обучения (ускоренное обучение):

№ раздела и темы	Наименование раздела и тема дисциплины	Трудоемкость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоятельная работа обучающихся*
			лекции	практические занятия	
1	2	3	4	6	7
<b>1.</b>	<b>Теоретические основы построения узлов ЭВМ</b>	<b>4,4</b>	<b>0,4</b>	<b>-</b>	<b>4</b>
1.1.	Физические формы представления информации	2,2	0,2	-	2
1.2.	Математические модели схем ЭВМ	1,1	0,1	-	1
1.3.	Элементы и узлы ЭВМ	1,1	0,1	-	1
<b>2.</b>	<b>Процессоры и микропроцессоры</b>	<b>9,6</b>	<b>0,6</b>	<b>5</b>	<b>4</b>
2.1.	Основные принципы построения устройств обработки цифровой информации	2,1	0,1	1	1
2.2.	Принципы организации арифметико-логических устройств	2,1	0,1	1	1
2.3.	Структура и формат команд. Кодирование команд	1,6	0,1	1	0,5
2.4.	Способы адресации	1,6	0,1	1	0,5
2.5.	Запоминающие устройства ЭВМ	1,6	0,1	1	0,5
2.6.	Принципы организации систем обработки прерывания программ	0,6	0,1	-	0,5
<b>3.</b>	<b>Организация ввода/вывода в ЭВМ</b>	<b>4,5</b>	<b>0,5</b>	<b>-</b>	<b>4</b>
3.1.	BIOS	2,2	0,2	-	2
3.2.	Системные и локальные шины	1,2	0,2	-	1
3.3.	Шины ввода/вывода	1,1	0,1	-	1
<b>4.</b>	<b>Вычислительные системы</b>	<b>4,5</b>	<b>0,5</b>	<b>-</b>	<b>4</b>

4.1.	Классификация вычислительных систем	2,2	0,2	-	2
4.2.	Многомашинные вычислительные системы	1,2	0,2	-	1
4.3.	Многопроцессорные вычислительные системы	1,1	0,1	-	1
<b>5.</b>	<b>Сети компьютеров</b>	<b>5</b>	<b>1</b>	-	<b>4</b>
5.1.	Основные понятия	1,25	0,25	-	1
5.2.	Локальные вычислительные сети	1,25	0,25	-	1
5.3.	Сетевой и транспортный уровни	1,25	0,25	-	1
5.4.	Структура и функции территориальных сетей	1,25	0,25	-	1
<b>6.</b>	<b>Микроконтроллеры</b>	<b>4</b>	<b>1</b>	-	<b>3</b>
6.1.	Определение микроконтроллера	0,8	0,2	-	0,6
6.2.	Классификация микроконтроллеров	0,8	0,2	-	0,6
6.3.	Тенденции в развитии современных микроконтроллеров	0,8	0,2	-	0,6
6.4.	Оценка быстродействия микроконтроллеров	0,8	0,2	-	0,6
6.5.	Энергосберегающие микроконтроллеры ведущих производителей	0,8	0,2	-	0,6
	<b>ИТОГО</b>	<b>32</b>	<b>4</b>	<b>5</b>	<b>23</b>

## 4.2. Содержание дисциплины, структурированное по разделам и темам

### Тема 1. Теоретические основы построения узлов ЭВМ

#### 1.1. Физические формы представления информации

Вся информация в ЭВМ кодируется совокупностью цифр. В свою очередь цифры отображаются квантованными по двум уровням сигналами.

В цифровых устройствах сигналы изменяются в дискретные моменты времени, обозначаемые целыми числами ( $t = 0, 1, \dots, n$ ). Временной интервал между соседними моментами дискретного времени называется *тактом*. Эти интервалы являются одинаковыми для синхронных устройств и неодинаковыми для асинхронных устройств.

На физическом уровне сигналы могут быть представлены одним из трех основных способов: потенциальным, импульсным или динамическим.

При *потенциальном* способе «нулю» соответствует низкий уровень напряжения, а «единице» - высокий. Потенциальный сигнал характеризуется амплитудами низкого ( $U_0$ ) и высокого ( $U_1$ ) уровней напряжения, а также временами нарастания и спада сигнала, которые именуются передним ( $t_n$ ) и задним ( $t_z$ ) фронтами соответственно.

При *импульсном* способе «0» и «1» соответствуют импульсы различной полярности, либо «0» соответствует отсутствие, а «1» - наличие импульса. Импульсный сигнал характеризуется амплитудой импульса  $U_m$ , шириной (продолжительностью импульса по основанию)  $t_u$ , и передним  $t_n$  и задним  $t_z$  фронтами импульса. В идеальном случае импульсные сигналы должны появляться в тактовые моменты. В действительности имеет место запаздывание импульсного сигнала относительно тактового момента на время  $t$ .

При *динамическом* способе представления информации двум возможным значениям переменной соответствует наличие либо отсутствие серии импульсов.

В электронных схемах и устройствах, входящих в состав ЭВМ, применяется потенциальный способ представления информации, а для передачи информации между ЭВМ, а также при работе с магнитными носителями информации применяются импульсный и динамический способы.

#### 1.2. Математические модели схем ЭВМ

Наиболее общей моделью любой схемы, узла или устройства ЭВМ является многополюсный черный ящик с  $l$  входами и  $m$  выходами. На входы модели поступают, а на выходах появляются сигналы, квантованные по двум уровням.



где  $x_i$  ( $i = 1, 2, \dots, l$ ) - входные сигналы,

$y_j$  ( $j = 1, 2, \dots, m$ ) - выходные сигналы.

Множество значений, которые может принимать переменная  $x_i$ , называют *алфавитом*. В современных ЭВМ алфавит входных и выходных сигналов состоит из двух букв: 0, 1.

На входы модели поступают в каждый тактовый момент упорядоченные наборы букв, называемые *словами*. Множество всех допустимых наборов слов называется *входным алфавитом*  $X$  данной схемы. Аналогично множество всех допустимых комбинаций, образуемых выходными сигналами, называется *выходным алфавитом*  $Y$ .

Математические модели отражают зависимость между входными и выходными переменными схемы посредством системы уравнений:

$$y_j(t) = f\{x_1(t), x_2(t), \dots, x_l(t), q_1(t), q_2(t), \dots, q_s(t)\} \quad (1.1)$$

где  $j = 1, 2, \dots, m$ , а переменные  $q_1, q_2, \dots, q_s$  отражают внутренние состояния схемы.

Если переменные  $y_j$  не зависят от внутреннего состояния схемы, то одинаковым наборам входных переменных соответствует один и тот же набор выходных переменных. Такие схемы называются *комбинационными*.

При этом система уравнений может быть записана в виде:

$$y_j(t) = f\{x_1(t), x_2(t), \dots, x_l(t)\}, \quad j = 1, 2, \dots, m. \quad (1.2)$$

Функции такого вида могут принимать только конечное число значений, и зависят от аргументов, также принимающих конечное число значений. Такие функции называются *переключательными*.

Далее будем рассматривать переключательные функции, которые могут принимать только два значения: 0 и 1, и аргументы которых также могут принимать только одно из этих двух значений. Такие переключательные функции получили название *булевых функций*.

Если выходные переменные  $y_j(t)$  зависят не только от входных переменных, но и от внутреннего состояния схемы, то для полного ее описания необходимо указать еще одну систему уравнений:

$$q_n(t+1) = f\{x_1(t), x_2(t), \dots, x_l(t), q_1(t), q_2(t), \dots, q_s(t)\}, \quad (1.3.)$$

где  $n = 1, 2, \dots, s$ .

Эта система отражает зависимость внутреннего состояния схемы в  $(t+1)$  такте от ее состояния и входных сигналов в такте  $t$ .

Схемы, описываемые уравнениями (1.1) и (1.3), получили название *цифровых автоматов*.

Для задания цифрового автомата должны быть указаны:

- 1) входной алфавит слов  $X$ ;
- 2) выходной алфавит слов  $Y$ ;
- 3) алфавит внутренних состояний  $Q$ ;
- 4) начальное состояние автомата  $q_0$ ;
- 5) функция переходов  $A(q, x)$ ;
- 6) функция выходов  $B(q, x)$ .

*Функция переходов* определяет зависимость состояния автомата  $q(t+1)$  в момент времени  $t+1$  от состояния автомата  $q(t)$  и входного сигнала  $x(t)$  в момент  $t$ .

*Функция выходов* определяет зависимость выходного сигнала  $y(t)$  от состояния автомата  $q(t)$  и входного сигнала  $x(t)$ .

## 1.3. Элементы и узлы ЭВМ

### 1.3.1. Логические элементы

*Системой элементов ЭВМ* называется функционально полный набор логических элементов, использующий одинаковый способ представления информации и одинаковый тип межэлементных связей.

Система элементов чаще всего избыточна по своему составу, что позволяет строить схемы с более простой топологией межэлементных связей и более экономные по количеству используемых элементов.

Логические элементы классифицируют:

1) По способу представления информации и типу межэлементных связей различают элементы импульсного, потенциального, импульсно-потенциального и динамического типа.

В современных ЭВМ применяются потенциальные и динамические элементы.

2) По функциональному назначению элементы принято разделять на  *типовые*  и  *элементы специального назначения* . К  *типовым*  относятся логические, запоминающие и формирующие элементы. Логические элементы предназначены для преобразования информации, запоминающие - для ее хранения, а формирующие элементы - для восстановления стандартизированных значений физических параметров сигналов, изменяющихся во время прохождения сигналов по электрическим цепям. К  *элементам специального назначения*  относятся усилители слабых сигналов, генераторы токов и напряжений специальной формы и другие элементы, не изменяющие информационного содержания сигналов.

3) В зависимости от используемых физических явлений логические элементы подразделяются на полупроводниковые, магнитополупроводниковые, электромагнитные и др.

### 1.3.2. Основные характеристики логических элементов

*Общие технические характеристики:*

- температурный диапазон,
- надежность,
- стоимость.

*Специфические характеристики:*

- функциональные возможности элемента,
- нагрузочная способность,
- быстродействие,
- помехоустойчивость,
- потребляемая мощность.

Функциональные возможности логического элемента характеризуются выполняемой им операцией и коэффициентами разветвления и объединения, т.е. факторами, влияющими на структуры более сложных схем, построенных с применением данного элемента.

При этом под *коэффициентом разветвления  $n$*  понимают число входов последующих ячеек, которые могут управляться от выхода данной ячейки, а под *коэффициентом объединения  $m$*  - число входов, которое может иметь ячейка. Величины  $m$  и  $n$  ограничиваются условиями сохранения нормального электрического режима ячейки.

*Нагрузочная способность* в общем случае определяется током, который может быть отдан ячейкой во внешние цепи (нагрузку). В случае однородных нагрузок, создаваемых входами идентичных ячеек, нагрузочная способность оценивается коэффициентом разветвления  $n$ .

*Быстродействие* логического элемента определяется скоростями его перехода из состояния "0" в состояние "1" и обратно. Переходные процессы изменения состояния элемента состоят из двух этапов: задержки и формирования фронта или спада сигнала. Длительность задержек и фронтов зависит от динамических свойств логического элемента.

Для оценки быстродействия часто используют обобщенную характеристику - *среднее время задержки*. В этом случае моментом поступления сигнала на ячейку считают момент достижения входным сигналом некоторого определенного уровня (например, 0.5 от установившегося значения). Моментом появления сигнала на выходе также считают момент достижения выходным сигналом этого уровня.

Одна из важнейших характеристик элемента - его *помехоустойчивость*. Различают *статическую* и *динамическую* помехоустойчивость. При определении статической помехоустойчивости помеха рассматривается как длительно действующий уровень потенциала, а при определении динамической помехоустойчивости - как импульс определенной длительности. Устойчивость элемента к воздействию длительной помехи меньше, чем к воздействию кратковременной помехи при одинаковых амплитудах. Устойчивость к воздействию динамической помехи тем ниже, чем выше быстродействие элемента.

С увеличением степени интеграции элементов ЭВМ все большую роль начинает играть такой параметр, как *рассеиваемая мощность*. Следует отметить, что закрытому состоянию соответствует один уровень рассеивания мощности ( $P_z$ ), а открытому - другой ( $P_o$ ). Обычно предполагают, что схе-

ма половину времени находится в открытом состоянии, а половину - в закрытом, и определяют среднюю рассеиваемую мощность следующим образом:

$$P_{cp} = \frac{P_z + P_o}{2}$$

### Основные характеристики логических элементов

*Амплитудная передаточная характеристика*  $U_{вых} = f(U_{вх})$  определяет формирующие свойства логического элемента, его помехоустойчивость, амплитуду и уровни стандартного сигнала. Вид характеристики зависит от типа логического элемента (ЭСЛ, ТТЛ и т.д.) и может изменяться в определенных пределах в зависимости от разброса параметров схем, изменений напряжения питания, нагрузки и температуры окружающей среды.

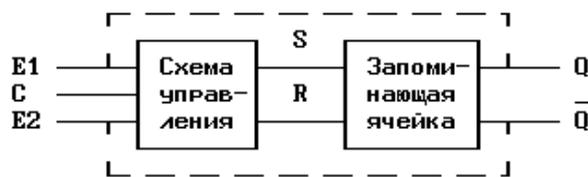
*Входная характеристика*  $I_{вх} = f(U_{вх})$  и *выходная характеристика*  $U_{вых} = f(I_{вых})$  позволяют определить нагрузочную способность элемента, режим его работы и способ согласования переходных процессов в линиях связи.

*Импульсная (динамическая) помехоустойчивость* - это зависимость допустимой амплитуды импульсной помехи от ее длительности  $U_{пом} = f(t_{пом})$ .

### 1.3.3. Триггеры

Практически все устройства ЭВМ совмещают функции переработки и хранения информации. Неотъемлемая часть таких устройств - элемент памяти. В арифметических и логических устройствах для хранения информации чаще всего используют элемент с двумя устойчивыми состояниями - *триггер*.

Структуру триггера можно представить в виде запоминающей ячейки и схему управления:



*Запоминающая ячейка* - это схема, которая имеет два выхода  $Q$  и  $\bar{Q}$ , сигналы на которых всегда противоположны (если на одном «0», то на другом «1»), и два входа - вход установки  $S$  (*set*) и вход сброса  $R$  (*reset*).

**Переключающий** сигнал по входу  $S$  устанавливает запоминающую ячейку в состояние "1", а по входу  $R$  - в состояние "0". В зависимости от типа элементов, из которых построена запоминающая ячейка, переключающим сигналом может являться либо "0", либо "1". Запоминающую ячейку называют также *асинхронным RS-триггером*.

*Схема управления* преобразует информацию, поступающую на входы  $E_1$  и  $E_2$  в сигналы, которые подаются на установочные входы запоминающей ячейки. В некоторых схемах выходные сигналы триггера поступают на вход схемы управления - на рисунке эти соединения показаны пунктиром.

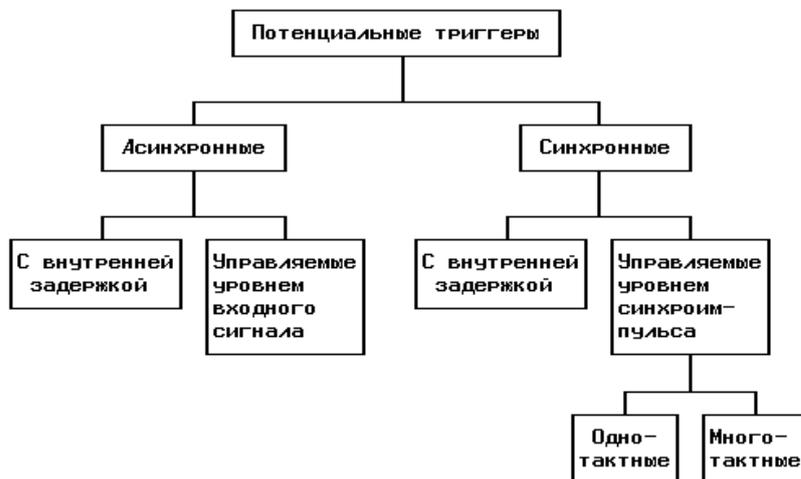
Как правило, триггеры, применяемые в потенциальной системе элементов, имеют еще один вход - вход для синхронизирующих сигналов  $C$ . Импульсы, поступающие на вход  $C$ , не несут логической информации, но определяют момент приема триггером входной информации.

### Классификация триггеров

В основу классификации триггерных устройств положены два основных признака: функциональный признак и способ записи информации в триггер.

Функциональная классификация - это классификация триггеров по типам схем управления. По функциональному признаку различают  $RS$ ,  $S$ ,  $R$ ,  $E$ ,  $T$ ,  $D$ ,  $TV$ ,  $DV$ ,  $RST$  и  $JK$  триггеры.

Классификация по способу записи информации характеризует временную диаграмму работы триггера, т.е. определяет ход процесса записи информации в триггер:



*Временная диаграмма* - это диаграмма, отображающая зависимость внутреннего состояния устройства, сигналов на его выходах и протекающих в нем переходных процессов от времени и сигналов на входах этого устройства.

Отличительной особенностью *асинхронных* триггеров является то, что запись информации в них осуществляется непосредственно в момент поступления информационного сигнала на вход триггера.

Запись информации в *синхронные тактируемые* триггеры осуществляется только при подаче разрешающего импульса (*синхроимпульса*) на синхронный вход *C*. Синхронные триггеры подразделяются на две категории: триггеры, срабатывающие по переднему фронту синхроимпульса ("*по уровню*"), и триггеры, срабатывающие по заднему фронту синхроимпульса ("*по спаду*").

Синхронные триггеры могут быть одноктактными и многотактными. Многотактные триггеры характеризуются тем, что формирование нового состояния триггера завершается с поступлением *n*-го синхроимпульса. Наибольшее распространение получили двухтактные синхронные триггеры.

Законы функционирования триггеров задаются таблицами переходов или составленными в соответствии с этими таблицами логическими уравнениями.

Входы триггеров обозначаются следующим образом:

- C* - вход синхронизации;
- S* (set) - вход установки триггера в 1;
- R* (reset) - вход сброса триггера в 0;
- D* (delay) - "задержка";
- T* (trigger) - "защелка";
- J* - вход установки JK-триггера в 1;
- K* - вход установки JK-триггера в 0;
- V* - управляющий вход DV-триггера.

Выходы триггеров: *Q* - прямой выход,  $\bar{Q}$  - инверсный выход.

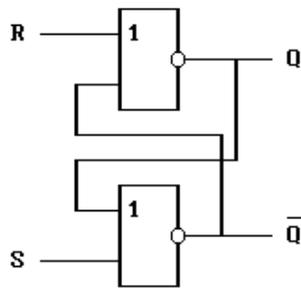
### *Асинхронные триггеры*

Асинхронные триггеры редко непосредственно используются в цифровых схемах, однако на базе асинхронных триггеров строятся все триггерные схемы.

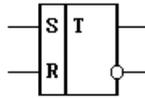
### *Асинхронный RS-триггер*

RS-триггер имеет два информационных входа *R* и *S*. При поступлении на эти входы сигналов  $S=1$  и  $R=0$  триггер принимает состояние  $Q=1$ , при  $S=0$  и  $R=1$  состояние  $Q=0$ , а при  $S=0$  и  $R=0$  триггер сохраняет то состояние, в котором он находился до поступления на его входы нулевых сигналов. Подача единичных сигналов на оба входа *R* и *S* запрещена.

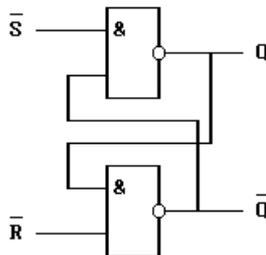
Асинхронный RS-триггер на элементах ИЛИ-НЕ:



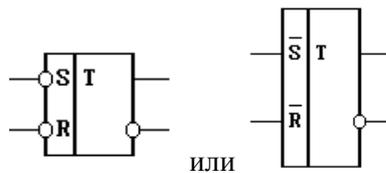
Условное графическое изображение такого триггера:



Асинхронный *RS*-триггер на элементах И-НЕ:

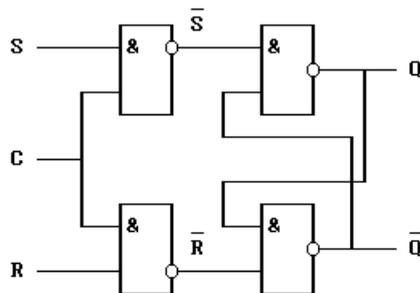


Условное графическое изображение такого триггера:



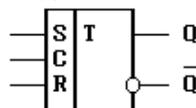
### *Синхронный одноклапный *RS*-триггер*

Синхронные *RS*-триггеры имеют на каждом входе дополнительные схемы совпадения:



Если на входе *C* "ноль", то на выходах схемы совпадения также будут нулевые значения при любых сигналах на входах *R* и *S*. При поступлении синхриимпульса на вход схемы совпадения информация с входов *R* и *S* инвертируется и передается на входы асинхронного триггера.

Графическое обозначение синхронного одноклапного *RS*-триггера:

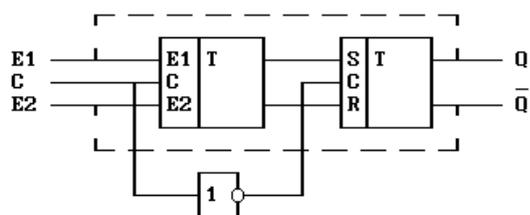


### *Синхронные двухклапные триггеры*

Синхронные двухступенчатые (двухклапные) триггеры построены по принципу "master-slave" (ведущий-ведомый). Триггерная схема состоит из двух частей-триггеров, одновременный прием информации в которые запрещен. Для построения первой и второй ступеней используют одноклапные синхронные триггеры. Информация передается во вторую ступень только после ее приема в первую ступень и окончания синхриимпульса, разрешаю-

шего запись информации в первую ступень. Такая последовательность приема информации достигается включением инвертора в цепь синхронизации для второй ступени.

Все двухтактные триггеры имеют следующую общую структуру:



Наиболее широкое применение в устройствах вычислительной техники находят двухтактные триггеры типов *RS*, *T*, *D* и *JK*.

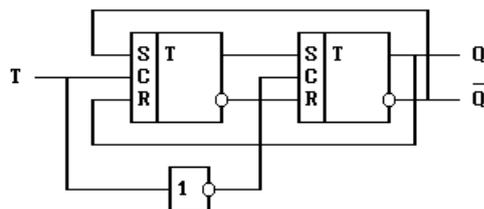
### T-триггеры

*T*-триггер (триггер со счетным входом) изменяет свое состояние на противоположное каждый раз при подаче на вход *T* единичного сигнала.

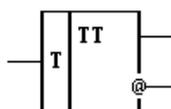
Таблица переходов асинхронного *T*-триггера:

$T(t)$	$Q(t+1)$
0	$Q(t)$
1	$\overline{Q(t)}$

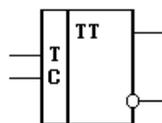
Схема асинхронного *T*-триггера:



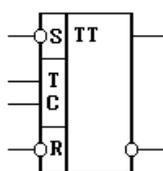
Графическое изображение асинхронного *T*-триггера:



Графическое изображение синхронного *T*-триггера:



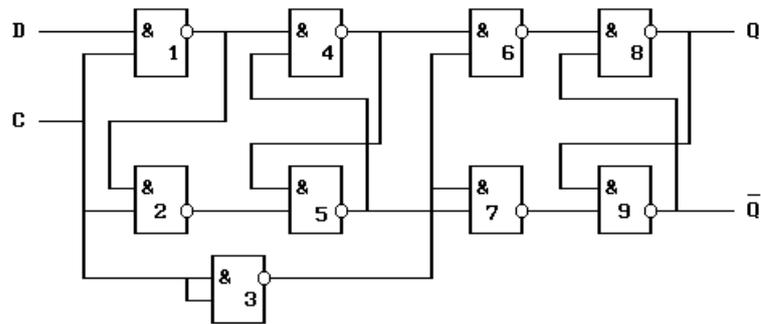
Синхронный *T*-триггер с цепями сброса и установки:



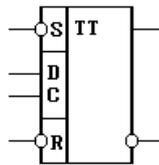
### *D*-триггеры

Триггер *D*-типа - это триггер задержки. Он описывается логическим уравнением  $Q(t+1)=D(t)$ , т.е. состояние *D*-триггера в момент времени  $t+1$  совпадает с кодом входного сигнала, действовавшего в момент времени  $t$ .

Схема *D*-триггера:



*D*-триггер с цепями сброса и установки:



#### *JK*-триггеры

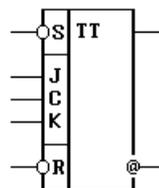
*JK*-триггер имеет обычно не менее 5 входов: входы асинхронной установки *S* и сброса *R*, вход тактовых импульсов *C* и управляющие входы *J* и *K*.

При подаче единиц на входы *J* и *K* *JK*-триггер работает как триггер со счетным входом: при поступлении каждого импульса на вход *C* (точнее, после спада импульса) состояние триггера изменяется на противоположное. Если на входах *J* и *K* установлены нулевые уровни, то состояние триггера при подаче импульсов на вход *C* не меняется. Если на входе *J* единица, а на входе *K* ноль, то после спада синхроимпульса на входе *C* *JK*-триггер устанавливается в состояние единица ( $Q=1$ ). Если на входе *J* ноль, а на входе *K* единица, то после спада синхроимпульса на входе *C* *JK*-триггер устанавливается в состояние ноль ( $Q=0$ ). Если на входе *C* низкий уровень, то изменение сигналов на входах *J* и *K* не влияет на состояние триггеров.

Следует различать "универсальные" и "синхронные" *JK*-триггеры. *Универсальный* триггер при наличии высокого уровня (единицы) на входе *C* и спаде сигнала (переходе из 1 в 0) на входе *J* переходит в состояние единица. При наличии единицы на входе *C* и спаде сигнала на входе *K* универсальный триггер переходит в состояние ноль. Данный режим работы универсальных триггеров позволяет в некоторых случаях упростить построение различных счетчиков.

Основным отличием синхронных триггеров от универсальных является то, что изменение состояния синхронных триггеров может происходить только по спаду импульсов на входе *C* или по поступлению сигналов сброса и установки на асинхронные входы *R* и *S*.

*JK*-триггер с цепями сброса и установки:



#### 1.3.4. Регистры

Электронную схему, выполняющую операции над одним машинным словом, называют узлом ЭВМ.

*Регистр* - это узел ЭВМ, который состоит из системы запоминающих элементов и управляющей этой системой логической схемы и предназначен для выполнения следующих операций:

- 1) сброс регистра в ноль;
- 2) прием  $n$ -разрядного кода числа из другого узла ЭВМ;

- 3) передача кода числа в другой узел;
- 4) хранение кода числа;
- 5) преобразование прямого кода числа в обратный или дополнительный и наоборот;
- 6) сдвиг влево или вправо кода числа на требуемое число разрядов;
- 7) преобразование последовательного кода в параллельный и наоборот;
- 8) поразрядное логическое сложение двух чисел;
- 9) поразрядное логическое умножение двух чисел;
- 10) поразрядное сложение двух чисел по модулю 2.

Конкретные регистры обычно являются специализированными и реализуют лишь некоторые из перечисленных операций.

#### Классификация регистров

По возможности смещения информации различают сдвигающие регистры и регистры памяти (регистры без сдвига).

По количеству тактов управления, необходимых для записи информации, различают одноктактные регистры (с приемом информации в парафазном коде), двухтактные (со сбросом перед записью информации) и многотактные (сдвигающие) регистры.

Сдвигающие регистры классифицируются по способу приема и передачи информации:

а) последовательные (прием и передача кода слова производятся последовательно, разряд за разрядом);

б) параллельные (с одновременным приемом или передачей кодов всех разрядов слова);

в) последовательно параллельные (прием и передача слова производятся группами по несколько разрядов, например, байтами).

Рассмотрим в качестве примера структуру четырехразрядного запоминающего регистра (рис. 1.1, а), работающего в парафазном коде. Изображение данного регистра на принципиальной схеме показано на рис. 1.1, б.

Регистры сдвига строятся на базе двухтактных триггеров *RS*, *D* или *JK*. Рассмотрим в качестве примера реверсивный сдвигающий регистр (рис. 1.2), построенный на базе *D*-триггеров с коммутаторами на базе логических элементов И-ИЛИ-НЕ:

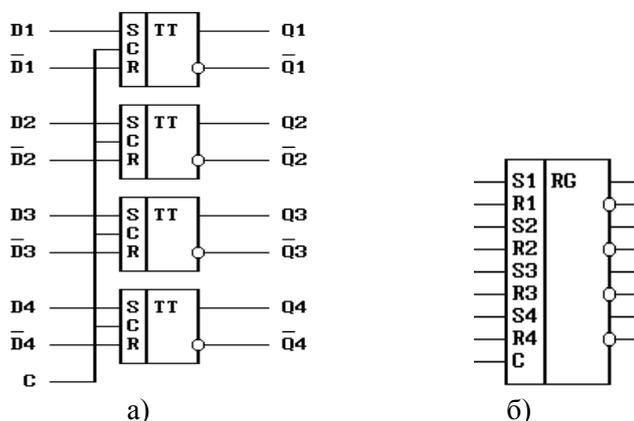


Рис. 1.1. Структура и обозначение четырехразрядного запоминающего регистра

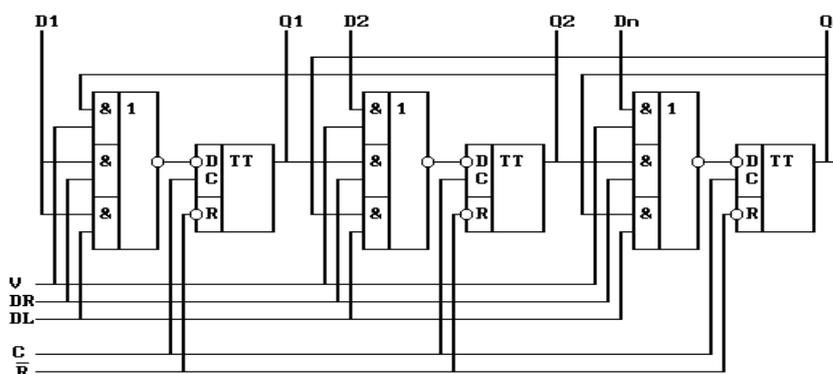


Рис. 1.2.

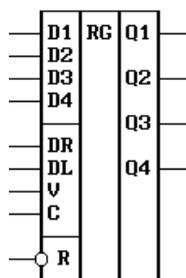
Обозначения, используемые в схеме реверсивного сдвигающего регистра (рис. 1.2):

*D1-Dn* - информационные входы;

*Q1-Qn* - информационные выходы;

- C* - вход синхронизации;
- R* - вход сброса;
- V* - разрешение записи информации (параллельного кода);
- DR* - сдвиг вправо;
- DL* - сдвиг влево;

На принципиальной схеме данный регистр изображается следующим образом:



### 1.3.5. Счетчики

*Счетчиком* называется схема, выполняющая функции подсчета количества единичных сигналов, поступивших на ее вход, а также функции формирования и запоминания некоторого кода, соответствующего этому количеству. Счетчики также иногда могут выполнять функции приема и выдачи кода.

Схемы счетчиков можно классифицировать по следующим признакам:

1) Основание системы счисления. В вычислительных системах используются двоичные и десятичные счетчики. Двоичные счетчики в свою очередь подразделяются на счетчики с модулем пересчета, равным  $2^n$ , и модулем пересчета, не равным  $2^n$ , где  $n$  - разрядность счетчика.

2) Направление переходов счетчика. Счетчики принято разделять на простые (суммирующие или вычитающие), которые могут вести счет только в одном направлении, то есть только прибавлять или вычитать входные сигналы, и реверсивные, которые в зависимости от управляющих сигналов могут вести счет в прямом или обратном направлениях.

3) Способ построения цепей переноса. Различают счетчики с последовательным, сквозным, параллельным и групповым переносом.

4) Способ организации счета. Счетчики могут быть асинхронными и синхронными. В асинхронных счетчиках изменение состояния счетчика осуществляется с поступлением информации только на вход первого каскада. В синхронных счетчиках информационный сигнал поступает одновременно на синхронные входы всех разрядов.

5) Тип элементов, используемых для построения счетчика. Различают счетчики на импульсных, импульсно-потенциальных и потенциальных элементах. Хотя в современной электронной аппаратуре используются все эти три типа, в дальнейшем мы будем рассматривать только счетчики на потенциальных элементах, так как это основной тип счетчиков, используемых в ЭВМ.

6) Тип организации счетного элемента. Счетчики могут быть построены на триггерах со счетным входом и на запоминающих элементах с использованием логических суммирующих схем.

Особую группу составляют счетчики, работающие по принципу циклического сдвигающего регистра (кольцевые счетчики). Данные счетчики отличаются низкой устойчивостью к помехам и сбоям и в ЭВМ практически не применяются.

#### Двоичные счетчики

Схемы счетчиков могут быть построены на базе триггеров. Число триггеров, необходимых для построения счетчика, определяется по формуле:

$$n = \text{ceil}(\log_2 k)$$

где  $k$  - коэффициент (модуль) пересчета - максимальное число внутренних состояний, которое может иметь счетчик,  $\text{ceil}$  - функция округления сверху до целого числа.

Одной из важных характеристик является *быстродействие*, которое в значительной мере определяется построением цепей переноса. Быстродействие характеризуется *временем установления кода*

на выходе счетчика  $T_{уст}$ , то есть интервалом времени между моментом поступления входного импульса и моментом окончания самого длительного переходного процесса в счетчике.

### Асинхронные счетчики

Простейшие схемы двоичных счетчиков выполняются путем последовательного соединения триггеров со счетными входами.  $T$ -триггер является счетчиком с  $K=2$ . Для построения счетчика с  $K=2^n$  требуется  $n$  таких триггеров. Рассмотрим в качестве примера схему асинхронного двоичного счетчика с последовательным переносом (рис. 1.3).

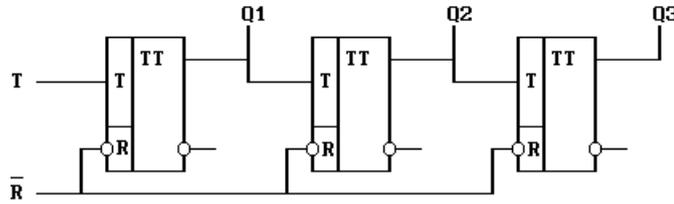


Рис. 1.3.

Основным недостатком счетчиков с последовательным переносом является низкое быстродействие.

### Счетчики с $K \neq 2^n$

Для многих электронных устройств необходимы счетчики с модулем пересчета, отличным от целой степени двойки. Для электронных часов, например, могут потребоваться счетчики с коэффициентами пересчета 3,6,7,10,24 и т.д. В ЭВМ счетчики применяются для задания сетки тактовых частот машины.

Принцип построения счетчиков с  $K \neq 2^n$  заключается в исключении лишних устойчивых состояний у двоичного счетчика с  $K=2^n$ .

Для реализации данного принципа используются следующие основные методы: 1) метод блокирования переноса; 2) принудительный порядок счета; 3) начальная установка кода.

Идея способа *блокирования переноса* заключается в том, что при подаче импульса с номером, несколько меньшим коэффициента пересчета, блокируется поступление счетных импульсов в разряды, находящиеся в нулевом состоянии, а последующие импульсы обнуляют триггеры, оставшиеся в единичном состоянии, так, что с приходом  $K$ -го импульса все триггеры счетчика оказываются в нулевом состоянии.

Рассмотрим троичный счетчик (рис. 1.7).

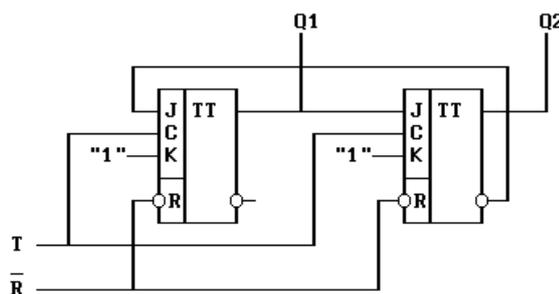


Рис. 1.7.

Временная диаграмма работы счетчика представлена на рис. 1.8.

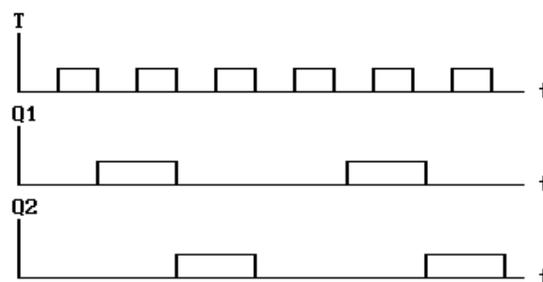


Рис 1.8.

В счетчиках с *принудительным порядком счета* исключение запрещенных состояний достигается за счет принудительной установки отдельных разрядов в состояние 1 в процессе счета. Для этого в схему счетчика вводятся обратные связи.

В счетчике с *начальной установкой кода* необходимое число запрещенных состояний устанавливается перед началом счета по сигналу "Начальная установка кода".

### 1.3.6. Дешифратор

*Дешифратором* называется комбинационная схема с несколькими входами и выходами, преобразующая код, подаваемый на входы, в сигнал на одном из выходов.

В общем случае дешифратор с  $n$  входами имеет  $2^n$  выходов, так как  $n$ -разрядный код входного слова может принимать  $2^n$  различных значений.

Рассмотрим в качестве примера следующую схему (рис.1.9).

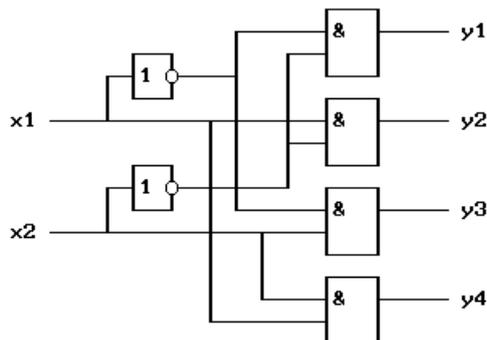
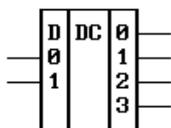


Рис. 1.9.

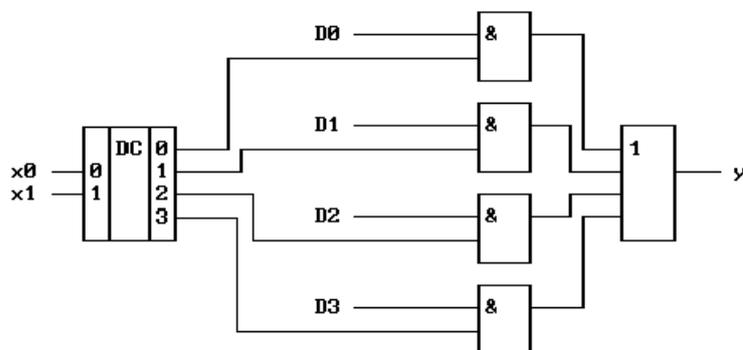
Условное обозначение дешифратора:



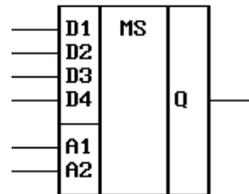
### 1.3.7. Мультиплексор

*Мультиплексором* называется схема, осуществляющая передачу сигналов с одной из входных линий на выходную. Выбор входной (информационной) линии производится с помощью кода, подаваемого на управляющие входы мультиплексора. Мультиплексор с  $k$  управляющими входами имеет  $2^k$  информационными входами.

Рассмотрим в качестве примера следующую схему:



Условное обозначение мультиплексора:



## Тема 2. Процессоры и микропроцессоры

### 2.1. Основные принципы построения устройств обработки цифровой информации

Согласно *принципа академика В.М. Глушкова*: «В любом устройстве обработки цифровой информации можно выделить операционный и управляющий блоки». Такой подход упрощает проектирование, а также облегчает понимание процесса функционирования вычислительного устройства.

Операционный блок состоит из регистров, сумматоров и других узлов, производящих прием из внешней среды и хранение кодов, их преобразование и выдачу результатов работы во внешнюю среду, а также выдачу в управляющий блок и внешнюю среду оповещающих сигналов.

Процесс функционирования во времени устройства обработки состоит из последовательности тактовых интервалов, в которых операционный блок производит элементарные преобразования кодов (передачу кода из одного регистра в другой, взятие обратного кода, сдвиг и т.д.).

Элементарная функциональная операция, выполняемая за один тактовый интервал и приводимая в действие одним управляющим сигналом называется *микрооперацией*. Управляющий блок вырабатывает распределенную во времени последовательность управляющих сигналов, порождающих в операционном блоке нужную последовательность микроопераций.

Последовательность управляющих сигналов (микрокоманд) определяется кодом операции, поступающим извне, состоянием операндов и промежуточными результатами преобразований.

Существует два основных типа управляющих автоматов:

1) Управляющий автомат с жесткой логикой.

Для каждой операции, задаваемой кодом операции команды, строится набор комбинационных схем, которые в нужных тактах возбуждают соответствующие управляющие сигналы.

2) Управляющий автомат с хранимой в памяти логикой.

Каждой выполняемой в операционном устройстве операции ставится в соответствие совокупность хранимых в памяти слов - микрокоманд, содержащих информацию о микрооперациях, подлежащих выполнению в течение одного машинного такта, и указание, какая микрокоманда должна выполняться следующей.

Последовательность микрокоманд, обеспечивающая выполнение некоторой операции (например, умножения), называется *микропрограммой* данной операции.

Функционирование вычислительного устройства может быть описано совокупностью реализуемых в нем микропрограмм, что является одним из способов описания цифровых устройств.

### 2.2. Принципы организации арифметико-логических устройств

*Арифметико-логические устройства* (АЛУ) служат для выполнения арифметических и логических преобразований над словами, называемыми в этом случае операндами. АЛУ служит основной частью операционного блока ЭВМ.

Выполняемые АЛУ операции можно разделить на следующие группы:

- операции двоичной арифметики для чисел с фиксированной запятой;
- операции двоичной арифметики для чисел с плавающей запятой;
- операции десятичной арифметики;
- операции индексной арифметики;
- операции специальной арифметики;
- операции над логическими кодами;
- операции над алфавитно-цифровыми полями.

Современные ЭВМ общего назначения обычно реализуют операции всех приведенных выше групп, а малые и микроЭВМ зачастую не имеют аппаратуры арифметики чисел с плавающей запятой, десятичной арифметики и операций над алфавитно-цифровыми полями. В этом случае недостающие операции реализуются специальными программами.

К арифметическим операциям относятся сложение, вычитание, взятие модулей ("короткие операции"), и умножение и деление ("длинные операции").

Группу логических операций составляют операции дизъюнкция (логическое *ИЛИ*) и конъюнкция (логическое *И*) над многоразрядными двоичными словами, а также операция сравнения кодов на равенство.

Специальные арифметические операции включают в себя нормализацию, арифметический сдвиг (сдвигаются только цифровые разряды, а знаковый остается на месте), логический сдвиг (знаковый разряд сдвигается вместе с цифровыми).

### 2.3. Структура и формат команд. Кодирование команд

Все возможные преобразования дискретной информации могут быть сведены к четырем основным видам:

- 1) передача информации в пространстве;
- 2) хранение информации;
- 3) логические операции;
- 4) арифметические операции.

ЭВМ, являющаяся универсальным преобразователем дискретной информации, выполняет все указанные виды преобразований.

Обработка информации в ЭВМ осуществляется автоматически путем программного управления. *Программа* представляет собой алгоритм обработки информации, записанный в виде последовательности команд, которые должны быть выполнены машиной для получения решения задачи.

*Команда* представляет собой код, определяющий операцию вычислительной машины и данные, участвующие в операции. Команда содержит также в явной или неявной форме информацию об адресе, по которому помещается результат операции, и об адресе следующей команды.

Процесс выполнения программы состоит из отдельных машинных операций. В данном случае под операцией понимается преобразование информации, выполняемое машиной под воздействием одной команды. Содержанием машинной операции могут быть запоминание в памяти, передача, арифметическое или логическое преобразование машинных слов, а также некоторые вспомогательные процедуры.

По характеру выполняемых операций различают следующие основные группы команд:

- 1) команды арифметических операций для чисел с фиксированной и плавающей запятой;
- 2) команды десятичной арифметики;
- 3) команды логических операций;
- 4) команды передачи кодов;
- 5) команды ввода-вывода;
- 6) команды передачи управления;
- 7) команды задания режима работы машины.

В команде, как правило, содержатся не сами операнды, а информация об адресах ячеек памяти или регистрах, в которых они находятся.

Код команды можно представить состоящим из нескольких частей или полей, имеющих определенное функциональное назначение при кодировании командной информации. Команда в общем случае состоит из операционной и адресной частей. В свою очередь, эти части могут состоять из нескольких полей.

*Операционная часть* содержит код операции, который задает вид операции. Адресная часть команды содержит информацию об адресах операндов и результата операции, в некоторых случаях информацию об адресе следующей команды.

Структура команды определяется составом, назначением и расположением полей в команде. *Форматом команды* называют ее структуру с разметкой номеров разрядов (бит), определяющих границы отдельных полей команды, или с указанием числа бит в определенных полях.

### 2.4. Способы адресации

*Адресный код* - это информация об адресе операнда, содержащаяся в команде.

*Исполнительный адрес* - это номер ячейки памяти, к которой производится фактическое обращение.

В современных ЭВМ адресный код, как правило, не совпадает с исполнительным адресом.

Выбор способов адресации, формирования исполнительного адреса и преобразования адресов является одним из важнейших вопросов разработки ЭВМ. Способы адресации, используемые в современных ЭВМ:

1) *Подразумеваемый операнд*. В команде может не содержаться явных указаний об операнде; в этом случае операнд подразумевается и фактически задается кодом операции команды.

2) *Подразумеваемый адрес*. В команде может не содержаться явных указаний об адресе участвующего в операции операнда или адресе, по которому должен быть размещен результат операции, но этот адрес подразумевается.

3) *Непосредственная адресация*. В команде содержится не адрес операнда, а сам операнд. При этом не требуется обращения к памяти для выборки операнда и ячейки памяти для его хранения. Это способствует уменьшению времени выполнения программы и занимаемого ею объема памяти. Непосредственная адресация удобна для хранения различного рода констант.

4) *Прямая адресация*. В адресной части команды может быть непосредственно указан исполнительный адрес.

5) *Относительная (базовая) адресация*. При этом способе адресации исполнительный адрес определяется как сумма адресного кода команды и базового адреса, как правило хранящегося в специальном регистре - регистре базы.

Относительная адресация позволяет при меньшей длине адресного кода команды обеспечить доступ к любой ячейке памяти. Для этого число разрядов в базовом регистре выбирают таким, чтобы можно было адресовать любую ячейку оперативной памяти, а адресный код команды используют для представления лишь сравнительно короткого "смещения". Смещение определяет положение операнда относительно начала массива, задаваемого базовым адресом.

6) *Укороченная адресация*. Для уменьшения длины кода команды часто применяется так называемая укороченная адресация. Суть ее сводится к тому, что в команде задаются только младшие разряды адресов, а старшие разряды при этом подразумеваются нулевыми. Такая адресация позволяет использовать только небольшую часть фиксированных ячеек в начале всей адресуемой области памяти, и поэтому применяется лишь совместно с другими способами адресации.

*Регистровая адресация* является частным случаем укороченной, когда в качестве фиксированных ячеек с короткими адресами используются регистры (ячейки сверхоперативной или местной памяти) процессора. Например, если таких регистров 16, то для адреса достаточно четырех двоичных разрядов. Регистровая адресация наряду с сокращением длины адресов операндов позволяет увеличить скорость выполнения операций, так как уменьшается число обращений к оперативной памяти.

7) *Косвенная адресация*. Адресный код команды в этом случае указывает адрес ячейки памяти, в которой находится адрес операнда или команды. Косвенная адресация широко используется в малых и микроЭВМ, имеющих короткое машинное слово, для преодоления ограничений короткого формата команды (совместно используются регистровая и косвенная адресация).

8) *Адресация слов переменной длины*. Эффективность вычислительных систем, предназначенных для обработки данных, повышается, если имеется возможность выполнять операции со словами переменной длины. В этом случае в машине может быть предусмотрена адресация слов переменной длины, которая обычно реализуется путем указания в команде местоположения в памяти начала слова и его длины.

9) *Стековая адресация*. Стековая память, реализующая безадресное задание операндов, особенно широко используется в микропроцессорах и мини-ЭВМ.

*Стек* представляет собой группу последовательно пронумерованных регистров или ячеек памяти, снабженных указателем стека, в котором автоматически при записи и считывании устанавливается номер (адрес) последней занятой ячейки стека (вершины стека). При операции записи заносимое в стек слово помещается в следующую по порядку свободную ячейку стека, а при считывании из стека извлекается последнее поступившее в него слово.

10) *Автоинкрементная и автодекрементная адресации*. Поскольку регистровая косвенная адресация требует предварительной загрузки регистра косвенным адресом из оперативной памяти, что связано с потерей времени, такой тип адресации особенно эффективен при обработке массива данных, если имеется механизм автоматического приращения или уменьшения содержимого регистра при каждом обращении к нему. Такой механизм называется соответственно автоинкрементной и автодекрементной адресацией. В этом случае достаточно один раз загрузить в регистр адрес первого обрабатываемого элемента массива, а затем при каждом обращении к регистру в нем будет формироваться адрес следующего элемента массива.

При автоинкрементной адресации сначала содержимое регистра используется как адрес операнда, а затем получает приращение, равное числу байт в элементе массива. При автодекрементной адресации сначала содержимое указанного в команде регистра уменьшается на число байт в элементе массива, а затем используется как адрес операнда.

Автоинкрементная и автодекрементная адресации могут рассматриваться как упрощенный вариант индексации.

11) *Индексация*. Для реализуемых на ЭВМ методов решения математических задач и обработки данных характерна цикличность вычислительных процессов, когда одни и те же процедуры выполняются над различными операндами, упорядоченно расположенными в памяти. Поскольку операнды, обрабатываемые при повторениях цикла, имеют разные адреса, без использования индексации требовалось бы для каждого повторения составлять свою последовательность команд, отличающихся адресными частями.

## 2.5. Запоминающие устройства ЭВМ

Запоминающие устройства классифицируют:

1. По типу запоминающих элементов (полупроводниковые, магнитные, конденсаторные, опто-электронные, голографические, криогенные).

2. По функциональному назначению (оперативные (ОЗУ), буферные (БЗУ), сверхоперативные (СОЗУ), внешние (ВЗУ), постоянные (ПЗУ)).

3. По способу организации обращения (с последовательным поиском, с прямым доступом, адресные, ассоциативные, стековые, магазинные).

4. По характеру считывания (с разрушением или без разрушения информации).

5. По способу хранения (статические или динамические).

6. По способу организации (однокоординатные, двухкоординатные, трехкоординатные, двух/трехкоординатные).

*Память ЭВМ* - совокупность всех запоминающих устройств, входящих в состав ЭВМ. Обычно в состав ЭВМ входит несколько различных типов ЗУ.

Производительность и вычислительные возможности ЭВМ в значительной степени определяются составом и характеристиками ее ЗУ.

Основными операциями в памяти в общем случае являются: занесение информации в память - *запись* и выборка информации из памяти - *считывание*. Обе эти операции называются *обращением к памяти*.

При обращении к памяти производится считывание или запись некоторой единицы данных - различной для устройств разного типа. Такой единицей может быть бит, байт, машинное слово или блок данных.

Важнейшими характеристиками отдельных устройств памяти являются емкость памяти, удельная емкость, быстродействие.

*Емкость памяти* определяется максимальным количеством данных, которые могут в ней храниться. Емкость измеряется в двоичных единицах (битах), машинных словах, чаще в байтах.

*Удельная емкость* – это отношение емкости ЗУ к его физическому объему.

*Быстродействие памяти* определяется продолжительностью операций обращения, т.е. временем, затрачиваемым на поиск единицы информации в памяти и на ее считывание, или временем на поиск места в памяти, предназначенного для хранения данной единицы информации, и на ее запись.

Принято разделять все запоминающие устройства на два основных типа: оперативные и внешние. Основным критерием для такого разделения служит скорость доступа к информации.

ОЗУ (оперативное запоминающее устройство) - запоминающее устройство, предназначенное для информации, непосредственно участвующей в процессе выполнения операций, выполняемых процессором. ОЗУ должно обеспечивать поступление новой информации в процессор с той же скоростью, с какой он ее обрабатывает.

ВЗУ (внешнее запоминающее устройство) - запоминающее устройство, предназначенное для длительного хранения массивов информации и обмена ими с ОЗУ. Обычно строятся на базе магнитных носителей информации. Само название этого класса устройств имеет исторический характер и произошло от больших ЭВМ, в которых все ВЗУ, как более медленные и громоздкие, размещались в собственном корпусе.

Внутренняя память ЭВМ организуется как взаимосвязанная совокупность нескольких типов ЗУ. В ее состав, кроме ОЗУ, могут входить следующие типы ЗУ:

Постоянное запоминающее устройство (ПЗУ) - запоминающее устройство, из которого может производиться только выдача хранящейся в нем информации. Занесение информации в ПЗУ производится при его изготовлении.

Полупостоянное (программируемое) ЗУ (ППЗУ) - ЗУ, в котором информация может обновляться с помощью специальной аппаратуры. Если возможно многократное обновление информации, то иногда такое ППЗУ называют репрограммируемым (РППЗУ).

Буферное запоминающее устройство (БЗУ) - запоминающее устройство, предназначенное для промежуточного хранения информации при обмене данными между устройствами ЭВМ, работаю-

щими с различными скоростями. Конструктивно оно может быть частью любого из функциональных устройств.

Местная память (сверхоперативное ЗУ, СОЗУ) - буферное запоминающее устройство, включаемое между ОЗУ и процессором или каналами. Различают местную память процессора и местную память каналов.

Стек - специально организованное ОЗУ, блок хранения которого состоит из регистров, соединенных друг с другом в цепочку, по которой их содержимое при обращении к ЗУ передается (сдвигается) в прямом или обратном направлении.

Кэш-память - разновидность стека, в котором хранятся копии некоторых команд из ОЗУ.

Видеопамять - область ОЗУ ЭВМ, в которой размещены данные, видимые на экране дисплея.

### Адресная память

При адресной организации памяти размещение и поиск информации в запоминающем массиве основаны на использовании адреса хранения слова. Адресом служит номер ячейки массива, в которой это слово размещается.

При записи (или считывании) слова в запоминающий массив инициирующая эту операцию команда должна указывать адрес, по которому производится запись (считывание).

Типичная структура адресной памяти (рис. 2.1) содержит запоминающий массив из  $N$   $n$ -разрядных ячеек (обычно  $n$  равно 1, 4, 8 или 16) и его аппаратное обрамление, включающее регистр адреса РГА, имеющий  $k$  разрядов ( $k$  больше или равно логарифму по основанию 2 от  $N$ ), информационный регистр РГИ, блок адресной выборки БАВ, блок усилителей считывания БУС, блок разрядных усилителей-формирователей сигналов записи БУЗ и блок управления памятью БУП.

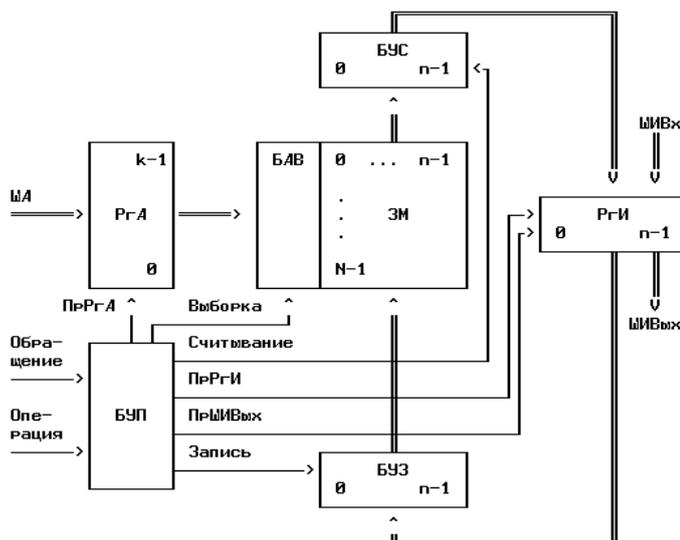


Рис. 2.1.

По коду адреса в регистре адреса блок адресной выборки формирует в соответствующей ячейке памяти сигналы, позволяющие произвести считывание или запись слова в ячейку.

Цикл обращения к памяти инициируется поступлением в блок управления памятью сигнала "Обращение".

Общая часть цикла обращения включает в себя:

- 1) прием в регистр адреса с шины адреса ША адреса обращения;
- 2) прием блоком управления и расшифровка управляющего сигнала "Операция", указывающего вид операции (считывание или запись).

Далее, при считывании:

3) блок адресной выборки дешифрирует адрес и посылает сигналы считывания в заданную ячейку, при этом код записанного в ячейке слова считывается усилителями считывания и передается в информационный регистр РГИ;

4) в памяти с разрушающим считыванием (при считывании все запоминающие ячейки устанавливаются в нулевое состояние) производится регенерация информации в ячейке путем записи в нее из информационного регистра РГИ считанного ранее слова;

5) считанное слово выдается из информационного регистра на выходную информационную шину ШИВых.

При записи:

3) производится прием записываемого слова с входной информационной шины ШИВх в информационный регистр;

4) блок адресной выборки производит выборку и очистку ячейки, заданной в регистре адреса (в памяти с разрушающим считыванием для этого производится считывание без записи в информационный регистр);

5) в выбранную ячейку записывается слово из информационного регистра.

БУП генерирует необходимые последовательности управляющих сигналов, инициирующих работу отдельных узлов памяти.

#### Ассоциативная память

В памяти этого типа поиск нужной информации производится не по адресу, а по ее содержанию (по ассоциативному признаку). При этом поиск по ассоциативному признаку (или последовательно по отдельным его разрядам) происходит параллельно во времени для всех ячеек запоминающего массива. Ассоциативный поиск позволяет существенно упростить и ускорить обработку данных. Это достигается за счет того, что в памяти этого типа операция считывания информации совмещена с выполнением ряда логических операций.

Память этого типа применяется в специализированных вычислительных машинах - машинах баз данных.

#### Стековая память

Стековая память, как и ассоциативная, является безадресной. В стековой памяти ячейки образуют одномерный массив, в котором соседние ячейки связаны друг с другом разрядными цепями передачи слов. Запись нового слова производится в верхнюю ячейку (ячейку 0), при этом все ранее записанные слова сдвигаются вниз, в соседние ячейки с большими на 1 номерами. Считывание возможно только из верхней ячейки памяти. Если производится считывание с удалением, все остальные слова в памяти сдвигаются вверх, в ячейки с меньшими номерами. В этой памяти порядок считывания слов соответствует правилу LIFO: последним поступил, первым обслуживается. В ряде устройств рассматриваемого типа предусматривается также операция простого считывания слова из нулевой ячейки без его удаления и сдвига слов в памяти.

Иногда стековая память снабжается счетчиком стека, показывающим количество занесенных в память слов.

В вычислительных машинах часто стековую память организуют, используя адресную память и регистр - указатель стека.

#### Запоминающие устройства типа 2D

Организация ЗУ типа 2D (рис.2.2.) обеспечивает двухкоординатную выборку каждого запоминающего элемента.

Основу ЗУ составляет плоская матрица из запоминающих элементов, сгруппированных в  $2^k$  ячеек по  $n$  разрядам. Обращение к ячейке задается  $k$ -разрядным адресом, выделение разрядов производится разрядными линиями записи и считывания.

Адрес ячейки  $i$  поступает на схему адресного формирователя АдрФ, управляемого сигналами "Чтение" и "Запись". Основу адресного формирователя составляет дешифратор с  $2^k$  выходами, который при поступлении на его входы адреса формирует сигнал для выборки линии  $i$ . При этом под действием сигнала "Чтение" формируется сигнал, настраивающий ЗЭ на выдачу сигнала состояния, а под действием сигнала "Запись" - на запись.

Выделение разряда  $j$  в  $i$ -ом слове производится второй координатной линией. При записи по линии  $j$  от усилителя записи поступает сигнал, устанавливающий выбранный для записи элемент в 0 или 1. При считывании на усилитель считывания по линии  $j$  поступает сигнал о состоянии элемента.

Используемые запоминающие элементы должны допускать объединение выходов для работы на общую линию с передачей сигналов только от выбранного элемента.

Каждая адресная линия передает три значения сигнала:

- 1) выборка при записи,
- 2) выборка при считывании,
- 3) отсутствие выборки.

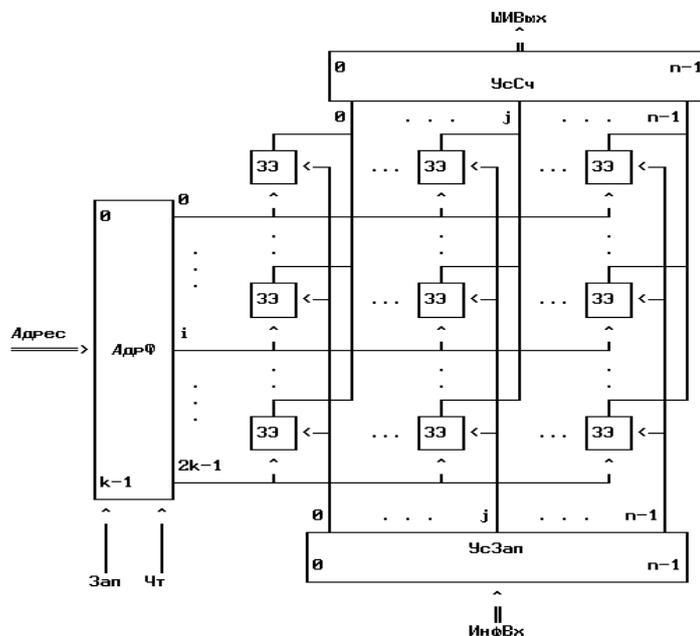


Рис. 2.2.

Каждая разрядная линия записи передает в запоминающий элемент записываемый бит информации, а разрядная линия считывания - считываемый бит информации. Линии записи и считывания могут быть объединены в одну при использовании элементов, допускающих объединение выхода со входом записи. Совмещение функций записи и считывания на разрядной линии широко используется в современных полупроводниковых ЗУ.

Запоминающие устройства типа 2D являются быстродействующими и достаточно удобными для реализации. Однако они неэкономичны по объему оборудования из-за наличия дешифратора с большим числом выходов. Поэтому структура 2D применяется только в ЗУ небольшой емкости.

#### Запоминающие устройства типа 3D

Некоторые ЗЭ имеют два входа выборок. Чтобы выполнялась операция выборки, требуется наличие сигнала выборки на обоих входах. Использование таких элементов позволяет строить ЗУ с трехкоординатным выделением ячеек.

Запоминающий массив ЗУ типа 3D представляет собой пространственную матрицу (рис. 2.3), составленную из  $n$  плоских матриц, представляющих собой запоминающий массив для отдельных разрядов ячеек памяти. Запоминающие элементы для разряда сгруппированы в квадратную матрицу.

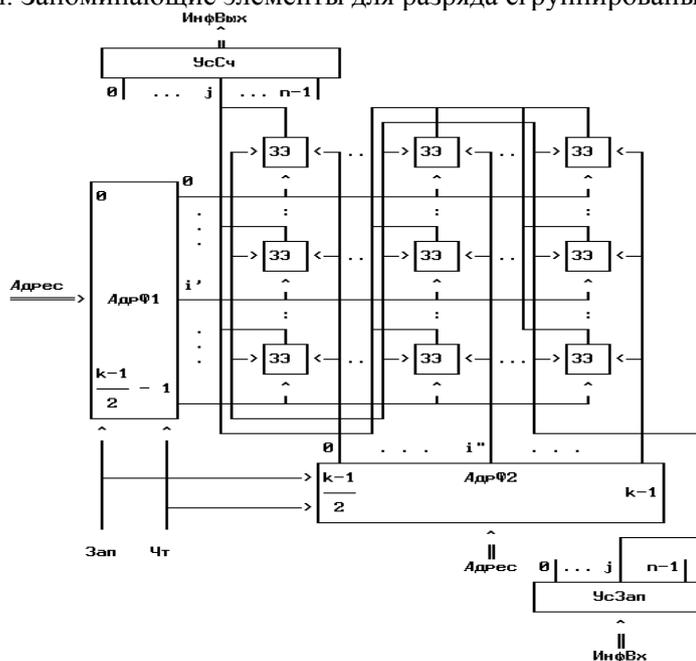


Рис. 2.3.

Для адресной выборки запоминающего элемента выдается две его координаты в массиве. Код ячейки памяти разделяется на старшую и младшую части, каждая из которых поступает на свой адресный формирователь. Адресные формирователи выдают код в соответствующие адресные линии. В

результате в массиве оказывается выбранным элемент, находящийся на пересечении двух адресных линий. Адресные формирователи управляются сигналами "Чтение" и "Запись" и в зависимости от них выдают сигналы выборки для считывания или для записи. При считывании сигнал о состоянии выбранного элемента поступает по линии считывания в усилитель. При записи в запоминающий элемент будет занесена информация, поступившая с соответствующего усилителя записи.

Для полупроводниковых ЗУ характерно объединение в одну линию разрядных линий записи и считывания.

Запоминающие устройства типа 3D более экономичны, чем ЗУ 2D. Однако элементы с тремя входами, используемыми при записи не всегда удается реализовать.

### Статические и динамические ОЗУ

Микросхемы ОЗУ по типу элементов памяти разделяют на статические и динамические. В микросхемах статических ОЗУ в качестве элементов памяти применены статические триггеры на биполярных или МДП-транзисторах. Как известно, статический триггер способен при наличии напряжения питания сохранять свое состояние неограниченное время. Число состояний, в которых может находиться триггер, равно двум, что и позволяет использовать его для хранения двоичной единицы информации.

В микросхемах динамических ОЗУ элементы памяти выполнены на основе электрических конденсаторов, сформированных внутри полупроводникового кристалла. Для обеспечения сохранности информации необходимо периодическое восстановление (регенерация) заряда конденсатора, поскольку из-за токов утечки запоминающий конденсатор может разряжаться. Это осуществляется с помощью периодических циклов регенерации, во время которых информация из элементов памяти считывается и вновь записывается обратно. Периодичность восстановления информации в элементах памяти называется *периодом регенерации*. Период регенерации Трег резко уменьшается с увеличением температуры окружающей среды. Длительность циклов регенерации обычно равна длительности циклов считывания или записи информации, но для полной регенерации информации в микросхеме необходимо несколько сотен таких циклов.

Микросхемы динамических ОЗУ отличаются от микросхем статических ОЗУ большей информационной емкостью, что обусловлено меньшим числом компонентов в одном элементе памяти, и, следовательно, более плотным их размещением в полупроводниковом кристалле. Однако динамические ОЗУ сложнее в применении, поскольку нуждаются в организации принудительной регенерации, в дополнительном оборудовании и более сложных устройствах управления.

Микросхемы статических ОЗУ применяются для создания сверхоперативной памяти ЭВМ, а также в устройствах автоматики, микроконтроллерах и т.п.

## 2.6. Принципы организации систем обработки прерывания программ

Во время выполнения программы внутри ЭВМ и во внешней среде могут возникать события, требующие немедленной реакции на них со стороны машины.

Реакция состоит в том, что машина прерывает обработку текущей программы и переходит к выполнению некоторой другой программы, специально предназначенной для данного события. По завершении этой программы ЭВМ возвращается к выполнению прерванной программы. Такой процесс называется *прерыванием программы*.

Каждое событие, требующее прерывания, сопровождается сигналом, оповещающим об этом событии ЭВМ, который называется *запросом прерывания*. Программу, затребованную запросом прерывания, называют *прерывающей программой*, а программу, выполнявшуюся до появления запроса - *прерываемой*.

Запросы на прерывание могут возникать как внутри самой ЭВМ, так и во внешней среде.

Внутренние запросы на прерывание возникают при сбоях в аппаратуре ЭВМ, переполнении разрядной сетки, попытке деления на 0, нарушении защиты памяти, при затребовании или завершении операции ввода-вывода периферийным устройством.

Внешние запросы могут поступать от датчиков технологического процесса и других ЭВМ.

Запросы прерывания поступают от нескольких параллельно развивающихся во времени процессов, требующих в произвольные моменты времени обслуживания со стороны процессора. К этим процессам относится выполнение основной программы, контроль правильности работы ЭВМ, операции ввода-вывода и т.п.

Система прерываний позволяет значительно эффективнее использовать процессор при наличии нескольких протекающих параллельно во времени процессов.

Для эффективной реализации системы прерываний ЭВМ снабжается соответствующими аппаратными и программными средствами, совокупность которых называется *контроллером прерываний*.

Основными функциями системы прерываний являются:

- запоминание состояния прерываемой программы и осуществление перехода к прерывающей программе;

- восстановление состояния прерванной программы и возврат к ней.

При наличии нескольких источников запросов прерываний должен быть установлен определенный порядок в обслуживании поступающих запросов, т.е. должны быть установлены приоритетные соотношения между запросами.

Приоритетные соотношения определяют, какой из нескольких поступивших запросов подлежит обработке в первую очередь, и устанавливают, имеет или не имеет право данный запрос прерывать ту или иную программу.

Прерывающая программа обычно состоит из трех частей:

1) подготовительной, обеспечивающей сохранение состояния прерванной программы;

2) основной, выполняющей затребованную запросом работу;

3) завершающей, обеспечивающей продолжение работы прерванной программы.

Характеристики системы прерывания:

1) Общее количество типов запросов прерываний (число входов в систему прерывания).

2) Время реакции - время между появлением запроса прерывания и началом выполнения прерывающей программы.

Время реакции на запрос прерывания зависит от того, сколько программ со старшим приоритетом ждет обслуживания. Поэтому в документации обычно указывают время реакции на запрос с наивысшим приоритетом.

3) Затраты времени на переключение программ, которые равны суммарному расходу времени на запоминание и восстановление состояния программы.

4) *Глубина прерывания* - это максимальное число программ, которые могут прерывать друг друга. Если после перехода к прерывающей программе и вплоть до ее окончания прием других запросов запрещается, то говорят, что система имеет глубину прерывания, равную 1. Глубина равна  $n$ , если допускается последовательное прерывание до  $n$  программ. Глубина прерывания обычно совпадает с числом уровней приоритета в системе прерываний. Системы с большим значением глубины прерывания обеспечивают более быструю реакцию на срочные запросы.

Если запрос на прерывание окажется необслуженным к моменту прихода нового запроса от того же источника, то возникает так называемое насыщение системы прерываний. В этом случае предыдущий запрос прерывания от данного источника будет машиной утрачен, что недопустимо. Быстродействие ЭВМ, характеристики системы прерываний, число источников прерывания и частоты возникновения запросов должны быть согласованы таким образом, чтобы насыщение было невозможным.

## Тема 3. Организация ввода/вывода в ЭВМ

### 3.1. BIOS

BIOS (Basic Input/Output System) – основная система ввода/вывода, зашитая в ПЗУ (отсюда название ROM BIOS). Она представляет собой набор программ проверки и обслуживания аппаратуры компьютера и выполняет роль посредника между DOS и аппаратурой. BIOS получает управление при включении и сбросе системной платы, тестирует саму плату и основные блоки компьютера - видеоадаптер, клавиатуру, контроллеры дисков и портов ввода/вывода, настраивает Chipset платы и загружает внешнюю операционную систему. При работе под DOS и Windows BIOS управляет основными устройствами, при работе под OS/2, UNIX, WinNT BIOS практически не используется, выполняя лишь начальную проверку и настройку.

Обычно на системной плате установлено только ПЗУ с системным (Main, System) BIOS, отвечающим за саму плату и контроллеры FDD, HDD, портов и клавиатуры; в системный BIOS практически всегда входит «System Setup» - программа настройки системы. Видеоадаптеры и контроллеры HDD с интерфейсом ST-506 (MFM) и SCSI имеют собственные BIOS в отдельных ПЗУ; их также могут иметь и другие платы - интеллектуальные контроллеры дисков и портов, сетевые карты и т.п.

### 3.2. Системные и локальные шины

Одним из простейших механизмов, позволяющих организовать взаимодействие различных подсистем, является единственная *центральная шина*, к которой подсоединяются все подсистемы. Доступ к такой шине разделяется между всеми подсистемами. Подобная организация имеет два ос-

новных преимуществ: низкая стоимость и универсальность. Поскольку такая шина является единственным местом подсоединения для разных устройств, новые устройства могут быть легко добавлены, и одни и те же периферийные устройства можно даже применять в разных вычислительных системах, использующих однотипную шину. Стоимость такой организации получается достаточно низкой, поскольку для реализации множества путей передачи информации используется единственный набор линий шины, разделяемый множеством устройств.

Главным недостатком организации с единственной шиной является то, что шина создает узкое горло, ограничивая, возможно, максимальную пропускную способность ввода/вывода. Если весь поток ввода/вывода должен проходить через центральную шину, такое ограничение пропускной способности весьма реально. В коммерческих системах, где ввод/вывод осуществляется очень часто, а также в суперкомпьютерах, где необходимые скорости ввода/вывода очень высоки из-за высокой производительности процессора, одним из главных вопросов разработки является создание системы нескольких шин, способной удовлетворить все запросы.

Традиционно шины делятся на шины, обеспечивающие организацию связи процессора с памятью, и шины ввода/вывода. Шины ввода/вывода могут иметь большую протяженность, поддерживать подсоединение многих типов устройств, и обычно следуют одному из шинных стандартов. Шины процессор-память сравнительно короткие, обычно высокоскоростные и соответствуют организации системы памяти для обеспечения максимальной пропускной способности канала память-процессор. На этапе разработки системы, для шины процессор-память заранее известны все типы и параметры устройств, которые должны соединяться между собой, в то время как разработчик шины ввода/вывода должен иметь дело с устройствами, различающимися по задержке и пропускной способности.

С целью снижения стоимости некоторые компьютеры имеют единственную шину для памяти и устройств ввода/вывода. Такая шина часто называется *системной*. Персональные компьютеры, как правило, строятся на основе одной системной шины в стандартах ISA, EISA или MCA. Необходимость сохранения баланса производительности по мере роста быстродействия микропроцессоров привела к двухуровневой организации шин в персональных компьютерах на основе локальной шины. Локальной шиной называется шина, электрически выходящая непосредственно на контакты микропроцессора. Она обычно объединяет процессор, память, схемы буферизации для системной шины и ее контроллер, а также некоторые вспомогательные схемы. Типичными примерами локальных шин являются VL-Bus и PCI.

Если шина синхронная, то она включает сигналы синхронизации, которые передаются по линиям управления шиной, и фиксированный протокол, определяющий расположение сигналов адреса и данных относительно сигналов синхронизации. Поскольку практически никакой дополнительной логики не требуется для того, чтобы решить, что делать в следующий момент времени, эти шины могут быть и быстрыми, и дешевыми. Однако они имеют два главных недостатка. Все на шине должно происходить с одной и той же частотой синхронизации, поэтому из-за проблемы перекоса синхросигналов, синхронные шины не могут быть длинными. Обычно шины процессор-память синхронные.

Асинхронная шина, с другой стороны, не тактируется. Вместо этого обычно используется старт-стопный режим передачи и протокол "рукопожатия" (handshaking) между источником и приемником данных на шине. Эта схема позволяет гораздо проще приспособить широкое разнообразие устройств и удлинить шину без беспокойства о перекосе сигналов синхронизации и о системе синхронизации. Если может использоваться синхронная шина, то она обычно быстрее, чем асинхронная, из-за отсутствия накладных расходов на синхронизацию шины для каждой транзакции. Выбор типа шины (синхронной или асинхронной) определяет не только пропускную способность, но также непосредственно влияет на емкость системы ввода/вывода в терминах физического расстояния и количества устройств, которые могут быть подсоединены к шине. Асинхронные шины по мере изменения технологии лучше масштабируются. Шины ввода/вывода обычно асинхронные.

Одной из популярных шин персональных компьютеров была системная шина, XT-Bus - шина архитектуры XT - первая в семействе IBM PC. Относительно проста, поддерживает обмен 8-разрядными данными внутри 20-разрядного (1 Мб) адресного пространства (обозначается как "разрядность 8/20"), работает на частоте 4.77 МГц. Совместное использование линий IRQ в общем случае невозможно. Конструктивно оформлена в 62-контактных разъемах.

ISA (Industry Standard Architecture - архитектура промышленного стандарта) - основная шина на компьютерах типа PC AT (другое название - AT-Bus). Является расширением XT-Bus, разрядность - 16/24 (16 Мб), тактовая частота - 8 МГц, предельная пропускная способность - 5.55 Мб/с. Разделение IRQ невозможно. Возможна нестандартная организация Bus Mastering, для этого нужен запрограммированный 16-разрядный канал DMA. Конструктивно выполнено в виде 62-контактного разъема XT-Bus с прилегающим к нему 36-контактным разъемом расширения.

EISA (Enhanced ISA) - функциональное и конструктивное расширение ISA. Внешне разъемы имеют такой же вид, как и ISA, и в них могут вставляться платы ISA, но в глубине разъема находятся дополнительные ряды контактов EISA, платы EISA имеют более высокую ножевую часть разъема с дополнительными рядами контактов. Разрядность - 32/32 (адресное пространство - 4 Гб), работает также на частоте 8 МГц. Предельная пропускная способность - 32 Мб/с. Поддерживает Bus Mastering - режим управления шиной со стороны любого из устройств на шине, имеет систему арбитража для управления доступом устройств у шине, позволяет автоматически настраивать параметры устройств, возможно разделение каналов IRQ и DMA.

Bus Mastering - способность внешнего устройства самостоятельно, без участия процессора, управлять шиной (пересылать данные, выдавать команды и сигналы управления). На время обмена устройство захватывает шину и становится главным, или ведущим (master) устройством. Такой подход обычно используется для освобождения процессора от операций пересылки команд и/или данных между двумя устройствами на одной нише.

MCA (Micro Channel Architecture - микроканальная архитектура) - шинacomпьютеров PS/2 фирмы IBM. Не совместима ни с одной другой, разрядность - 32/32, (базовая - 8/24, остальные - в качестве расширений). Поддерживает Bus Mastering, имеет арбитраж и автоматическую конфигурацию, синхронная (жестко фиксирована длительность цикла обмена), предельная пропускная способность - 40 Мб/с. Конструктивно выглядит, как одно- трехсекционный разъем. Первая, основная, секция - 8-разрядная (90 контактов), вторая - 16-разрядное расширение (22 контакта), третья - 32-разрядное расширение (52 контакта). VLB (VESA Local Bus - локальная шина стандарта VESA) - 32-разрядное (дополнение к шине ISA. Конструктивно представляет собой дополнительный разъем (116- контактный, как у MCA) при разъеме ISA. Разрядность - 32/32, тактовая частота - 25..50 МГц, предельная скорость обмена - 130 Мб/с. Электрически выполнена в виде расширения локальной шины процессора - большинство входных и выходных сигналов процессора передаются непосредственно VLB-платам без промежуточной буферизации.

PCI (Peripheral Component Interconnect - соединение внешних компонент) - развитие VLB в сторону EISA/MCA. Не совместима ни с какими другими, разрядность - 32/32 (расширенный вариант - 64/64), тактовая частота - до 33 МГц (PCI 2.1 - до 66 МГц), пропускная способность - до 132 Мб/с (264 Мб/с для 32/32 на 66 МГц и 528 Мб/с для 64/64 на 66 МГц), поддержка Bus Mastering и автоконфигурации.

Существует также расширение MediaBus, введенное фирмой ASUSTek - дополнительный разъем содержит сигналы шины ISA.

PCMCIA (Personal Computer Memory Card International Association - ассоциация производителей плат памяти для персональных компьютеров) - внешняя шина компьютеров класса NoteBook. Другое название модуля PCMCIA - PC Card. Предельно проста, разрядность - 16/26 (адресное пространство - 64 Мб), поддерживает автоконфигурацию, возможно подключение и отключение устройств в процессе работы компьютера. Конструктив - миниатюрный 68-контактный разъем. Контакты питания сделаны более длинными, что позволяет вставлять и вынимать карту при включенном питании компьютера.

### 3.3. Шины ввода/вывода

#### 3.3.1. Шина AGP

Шина персонального компьютера (PC) постоянно терпит множество изменений в связи с повышаемыми к ней требованиями.

Так ускоренный графический порт (AGP) это расширение шины PCI, чье назначение обработка больших массивов данных 3D графики. Intel разрабатывала AGP, для решения двух проблем перед внедрением 3D графики на PCI. Во-первых, 3D графика требуется как можно больше памяти информации текстурных карт (texture maps) и z-буфера (z-buffer). Чем больше текстурных карт доступно для 3D приложений, тем лучше выглядит конечный результат. При нормальных обстоятельствах z-буфер, который содержит информацию относящуюся к представлению глубины изображения, использует ту же память как и текстуры. Этот конфликт предоставляет разработчикам 3D множество вариантов для выбора оптимального решения, которое они привязывают к большой значимости памяти для текстур и z-буфера, и результаты напрямую влияют на качество выводимого изображения.

Разработчики PC имели ранее возможность использовать системную память для хранения информации о текстурах и z-буфера, но ограничение в таком подходе, была передача такой информации через шину PCI. Производительность графической подсистемы и системной памяти ограничиваются физическими характеристиками шины PCI (рис.3.1, а). Кроме того, ширина полосы пропускания

PCI, или ее емкость, не достаточна для обработки графики в режиме реального времени. Чтобы решить эти проблемы Intel разработала AGP.

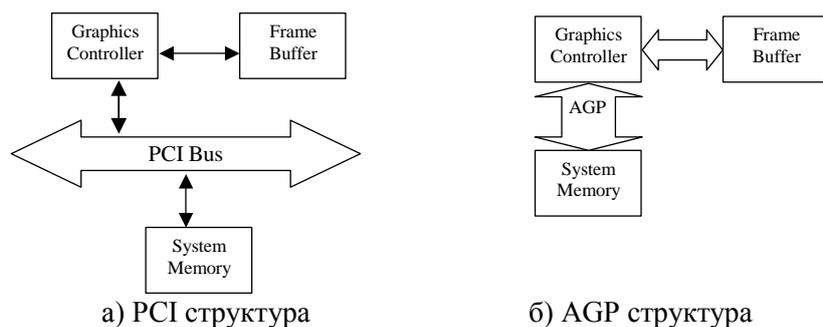


Рис. 3.1. Схемы PCI и AGP

Если определить кратко, что такое AGP, то это - прямое соединение между графической подсистемой и системной памятью (рис.3.1, б). Это решение позволяет обеспечить значительно лучшие показатели передачи данных, чем при передаче через шину PCI, и явно разрабатывалось, чтобы удовлетворить требованиям вывода 3D графики в режиме реального времени. AGP позволит более эффективно использовать память страничного буфера (frame buffer), тем самым увеличивая производительность 2D графики также, как увеличивая скорость прохождения потока данных 3D графики через систему.

Определение AGP, как вид прямого соединения между графической подсистемой и системной памятью, называется соединением point-to-point. В действительности, AGP соединяет графическую подсистему с блоком управления системной памятью, разделяя этот доступ к памяти с центральным процессором компьютера (CPU).

### 3.3.2. Шина USB

Шина USB предназначена для обеспечения обмена данными между компьютером (центральным процессором устройства) и подсоединенными к нему периферийными устройствами (ПУ) в условиях динамического (горячего) изменения конфигурации системы.

При проектировании новой шины особое внимание обращалось на следующие показатели:

- простоту изменения конфигурации системы;
- стоимость законченного решения при пропускной способности до 12 Мбит/с;
- возможность передачи потоков аудио- и сжатых видеоданных в реальном времени;
- обеспечение одновременной передачи разных типов данных;
- адаптацию к существующей инфраструктуре ПК и возможность быстрого включения интерфейса шины в представленное на рынке прикладное ПО;
- стимулирование разработки новых классов устройств, расширяющих возможности ПК.

Возможности USB следуют из ее технических характеристик:

- высокая скорость обмена (full-speed signaling bit rate) - 12 Mb/s;
- максимальная длина кабеля для высокой скорости обмена - 5 м;
- низкая скорость обмена (low-speed signaling bit rate) - 1.5 Mb/s;
- максимальная длина кабеля для низкой скорости обмена - 3м;
- максимальное количество подключенных устройств (включая размножители) - 127;
- возможно подключение устройств с различными скоростями обмена;
- отсутствие необходимости в установке пользователем дополнительных элементов, таких как терминаторы для SCSI;
- напряжение питания для периферийных устройств - 5 V;
- максимальный ток потребления на одно устройство - 500 mA (это не означает, что через USB можно запитать устройства с общим током потребления  $127 \times 500 \text{ mA} = 63.5 \text{ A}$ ).

Сама шина - это многоуровневая иерархическая система. На физическом уровне топология шины представляет собой корневидную структуру (рис. 3.2) - многоуровневую звезду (в терминологии стандарта), при которой соединения могут формировать цепочки и звезды. Закольцовка соединений в системе не допускается (этому, в частности, препятствует разная конструкция разъемов входного и выходного портов шины USB).



Рис. 3.2. Топология шины USB

В самой верхней части "корня" шины USB находится корневой концентратор, обеспечивающий связь периферии с компьютером (хостом). В текущей реализации стандарта допускается наличие только одного корневого концентратора, хотя и делается очень важная оговорка о возможности модернизации в будущем с целью поддержки нескольких корневых узлов в одной системе, что позволит, по мнению автора, создавать своего рода микросети в пределах, например, одного помещения. Уже практически готовы основные, системообразующие, решения и компоненты: концентраторы, мультиплексированная шина, программные и аппаратные средства ее поддержки.

Центром каждой звезды является узел (концентратор) шины USB, который обеспечивает набор двухточечных соединений с другими узлами и/или функциями, лежащими вниз по потоку (т. е. на большем удалении от компьютера). Узел состоит из двух функциональных элементов - повторителя, служащего для управления коммутацией потоков информации между входным и выходными портами узла, и контроллера, предназначенного для управления статусом (состоянием) узла и его портов (рис. 3.2).

### 3.3.3. Шины IDE и SCSI

Одними из наиболее популярных шин ввода-вывода в настоящее время являются шины IDE и SCSI.

Под термином IDE (Integrated Drive Electronics - электроника, встроенная в привод), или ATA (AT Attachment - подключаемый к AT) понимается простой и недорогой интерфейс для PC AT. Все функции по управлению накопителем обеспечивает встроенный контроллер, а 40-проводной соединительный кабель является фактически упрощенным сегментом 16-разрядной магистрали AT-Bus (ISA). Простейший адаптер IDE содержит только адресный дешифратор - все остальные сигналы заводятся прямо на разъем ISA. Адаптеры IDE обычно не содержат собственного BIOS - все функции поддержки IDE встроены в системный BIOS PC AT. Однако интеллектуальные или кэширующие контроллеры могут иметь собственный BIOS, подменяющий часть или все функции системного.

Основной режим работы устройств IDE - программный обмен (PIO) под управлением центрального процессора, однако все современные винчестеры EIDE поддерживают обмен в режиме DMA, а большинство контроллеров - режим Bus Mastering.

Под термином SCSI - Small Computer System Interface (интерфейс малых вычислительных систем) обычно понимается набор стандартов, разработанных Национальным институтом стандартов США (ANSI) и определяющих механизм реализации магистрали передачи данных между системной шиной компьютера и периферийными устройствами. На сегодняшний день приняты два стандарта (SCSI-1 и SCSI-2). Стандарт SCSI-3 находится в процессе доработки.

Стандарт SCSI-2 определяет в частности различные режимы: Wide SCSI, Fast SCSI и Fast-and-Wide SCSI. Стандарт SCSI-1 определяет построение периферийной шины на основе 50-жильного экранированного кабеля, описывает методы адресации и электрические характеристики сигналов. Шина данных SCSI-1 имеет разрядность 8 бит, а максимальная скорость передачи составляет 5 Мбайт/сек. Fast SCSI сохраняет 8-битовую шину данных и тем самым может использовать те же самые физические кабели, что и SCSI-1. Он отличается только тем, что допускает передачи со скоростью 10 Мбайт/сек в синхронном режиме. Wide SCSI удваивает либо учетверяет разрядность шины данных (либо 16, либо 32 бит), допуская соответственно передачи со скоростью либо 10, либо 20 Мбайт/сек. В комбинации Fast-and-Wide SCSI возможно достижение скоростей передачи 20 и 40 Мбайт/сек соответственно.

## Тема 4. Вычислительные системы

### 4.1. Классификация вычислительных систем

Создание вычислительных систем (ВС) способствует разрешению противоречия между непрерывно растущими потребностями в быстродействующих и надежных средствах вычислений и пределом технических возможностей ЭВМ на данном этапе развития.

Вычислительная система представляет собой сложный комплекс, состоящий из разнообразных технических средств и соответствующего программного обеспечения. Как технические, так и программные средства имеют модульную структуру построения, позволяющую наращивать ее в зависимости от назначения и условий эксплуатации системы.

Первыми ВС были однопроцессорные мультипрограммные ЭВМ, высокая производительность которых была достигнута за счет распределения во времени основных устройств системы между программами.

Дальнейшее повышение производительности ЭВМ было достигнуто за счет мультиобработки программ (задач), т. е. за счет разбиения программ на отдельные блоки и параллельной обработки этих блоков на нескольких обрабатывающих устройствах, входящих в состав ВС. Мультиобработка позволяет не только повысить производительность, но и сократить время выполнения отдельных программ, которые могут разбиваться на части и распределяться между различными обрабатывающими устройствами.

Первым типом ВС с мультиобработкой был многомашинный комплекс (МК) - многомашинная ВС. В состав МК объединялись различные ЭВМ с классической структурой, имеющие возможность обмениваться информацией.

На рис. 4.1 представлена структура двухмашинной ВС. Каждая ЭВМ имеет оперативную память (ОП), внешнее запоминающее устройство (ВЗУ), периферийные устройства (ПфУ), подключаемые к центральной части ЭВМ - процессору (ПР) с помощью каналов ввода-вывода (КВВ), и работает под управлением своей операционной системы (ОС). Обмен информацией между ЭВМ1 и ЭВМ2 осуществляется через системные средства обмена (ССО) в результате взаимодействия ОС машин между собой.

Основной недостаток многомашинной ВС - недостаточно эффективно используется оборудование комплекса. Достаточно, в ВС в каждой ЭВМ выйти из строя по одному устройству (даже разных типов), как вся ВС становится неработоспособной.

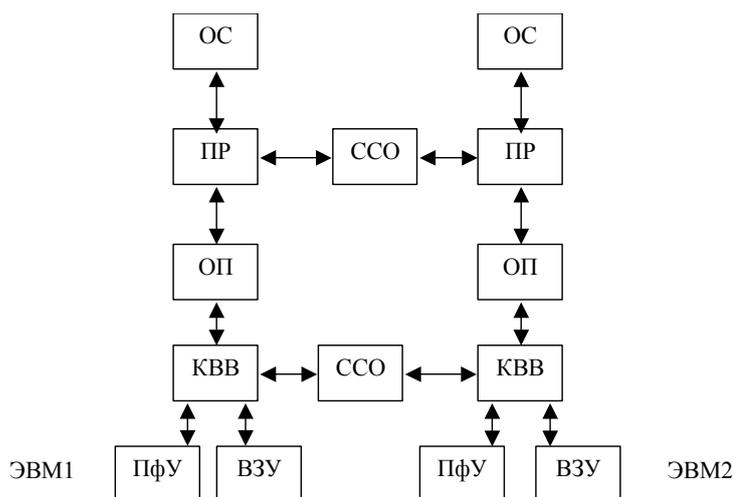


Рис. 4.1. Структура многомашинной ВС

Следующим шагом в направлении дальнейшего увеличения производительности ВС явилось создание многопроцессорных ВС с мультиобработкой, в составе которых содержится два или несколько процессоров (ПР), работающих с единой ОП, общий набор КВВ и ВЗУ (рис. 4.2). Наличие единой ОС делает возможным автоматическое распределение ресурсов системы на различных этапах ее работы. В результате достигается высокая «живучесть» ВС, позволяющая в случае отказа отдельных модулей перераспределить нагрузку между работоспособными, обеспечив тем самым выполнение наиболее важных для ВС функций.

К недостаткам многопроцессорных ВС относят трудности, возникающие при реализации общего поля ОП, ВЗУ, а также при разработке специальной ОС.

Дальнейшее развитие идей мультиобработки привело к созданию крупных многопроцессорных систем высокой производительности, получивших название высокопараллельных ВС. Такие ВС в зависимости от ее структуры могут одновременно обрабатывать множественный поток данных или команд. Под потоком команд понимается последовательность команд, выполняемых ВС, а потоком данных - последовательность данных, обрабатываемых под управлением потока команд.

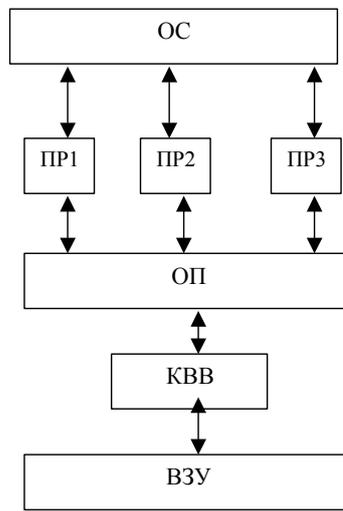


Рис. 4.2. Структура многопроцессорной ВС

Высокопараллельные ВС структуры типа ОКМД (одиничный поток команд и множественный поток данных) получили название матричных ВС (рис. 4.3). Они содержат некоторое количество одинаковых сравнительно простых быстродействующих процессоров (ПР), соединенных друг с другом так, что образуется сетка (матрица), в узлах которой размещаются ПР. Все ПР выполняют одну и ту же команду, но над разными операндами, доставляемыми процессорам из памяти несколькими потоками данных.

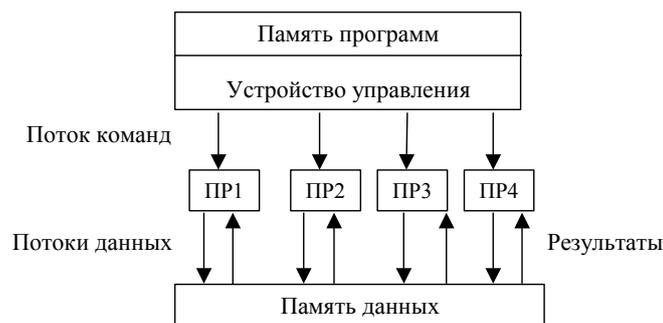


Рис. 4.3. Многопроцессорная ВС структуры типа ОКМД

Высокопараллельные ВС структуры типа МКОД (множественный поток команд и одиничный поток данных) получили название конвейерных. ВС. Такие ВС (рис. 4.4) содержат цепочку последовательно соединенных ПР, так что информация на выходе одного ПР является входной информацией для другого ПР. Каждый ПР обрабатывает соответствующую часть задачи, передавая результаты соседнему ПР, который использует их в качестве исходных данных.

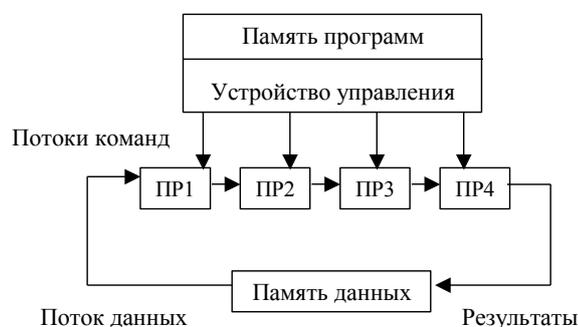


Рис. 4.4. Многопроцессорная ВС структуры типа МКОД

Так, например, операция сложения чисел с плавающей запятой может быть разделена на 4 этапа: сравнение порядков; выравнивание порядков; сложение мантисс; нормализация результата. В

конвейерной ВС все эти этапы вычислений будут выполняться отдельными процессорами, образующими конвейер.

Высокопараллельные ВС по сравнению с многопроцессорными ВС обеспечивают более высокую производительность, надежность, и "живучесть". Однако при этом усиливаются недостатки: усложнение управления системой, трудность программирования и малая загрузка системы.

Первые два недостатка компенсируются благодаря применению специальных языков программирования. Третий недостаток приводит к тому, что большинство высокопараллельных ВС ориентируется на специализированное применение.

Многопроцессорные и многомашинные ВС классифицируют по различным признакам. Рассмотрим некоторые из них.

По назначению ВС делятся на:

- универсальные ВС - предназначены для решения широкого круга задач, специализированные

- специализированные - для решения определенного круга задач. Специализированные ВС, как правило, должны иметь аппаратные и программные средства, предназначенные специально для этой системы.

По типу оборудования ВС подразделяются на:

- однородные системы - содержат несколько однотипных ЭВМ (или процессоров). Основным недостатком однородных ВС – неполная загруженность отдельных ЭВМ (процессоров) во время ее работы.

- неоднородные системы - разнотипные ЭВМ (или процессоры). Используются в целях повышения эффективности использования ЭВМ (процессоров). Например, более производительная центральная ЭВМ системы выполняет обработку информации, а менее производительные ЭВМ1, ЭВМ2 и ЭВМ3 осуществляют ввод и вывод информации, ее предварительную обработку и передачу в центральную ЭВМ. Коммутатор при передаче информации в центральную ЭВМ и выдаче из нее результатов настраивается на выбор соответствующей ЭВМ 2-го уровня (рис. 4.5).

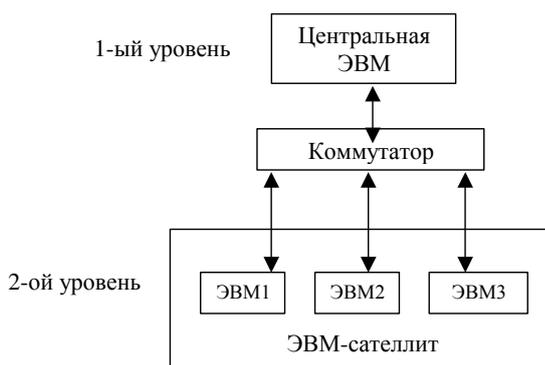


Рис. 4.5. Иерархическая структура многомашинной ВС

Вычислительные системы с иерархической структурой могут иметь и более двух уровней иерархии. ЭВМ, выполняющие предварительную обработку информации, часто, называют *машинами-спутниками*.

По типу структуры (под структурой ВС понимают состав системы и схемы функциональных и управляющих связей между ее элементами) ВС разделяются:

- с постоянной структурой - в процессе функционирования системы не изменяется состав функциональных и управляющих связей между ее элементами.

- с переменной структурой – такую структуру имеют адаптивные системы, т. е. такие, у которых структура изменяется на основе анализа текущей информации. Подобные системы позволяют достичь оптимального состояния в любых изменяющихся условиях функционирования.

По степени централизации управления ВС разделяются на:

- централизованные ВС - все функции управления сосредоточены в одном элементе, в качестве которого используется одна из ЭВМ, называемая машиной-директором, или центральный процессор.

- децентрализованные ВС - каждый процессор или ЭВМ действуют автономно, решая свои задачи.

- со смешанным управлением - ВС разбивается на группы взаимодействующих ЭВМ (или процессоров), в каждой из которых осуществляется централизованное управление, а между группами – децентрализованное

## 4.2. Многомашинные вычислительные системы

В настоящее время наиболее широко используют двухмашинные вычислительные комплексы, которые могут работать в одном из следующих режимов:

1. 100%-ное горячее резервирование. Обе ЭВМ в этом режиме исправны и работают параллельно, выполняя одни те же операции над одной и той же информацией (дуплексный режим). После выполнения каждой команды результаты преобразования сравниваются и при их совпадении процесс вычислений продолжается. При этом в памяти обеих ЭВМ в каждый момент находится одна и та же информация. При обнаружении несовпадения в результатах обработки неисправная ЭВМ выводится на ремонт, а исправная ЭВМ продолжает работать под контролем встроенной в ЭВМ системы автоматического контроля.

2. Одна исправная ЭВМ решает задачи без дублирования, а другая ЭВМ находится в режиме «Профилактика», в котором осуществляется прогон контролирующих тестов. Если основная ЭВМ не в состоянии выполнить задачу, то резервная может прекратить "Профилактику" и начать работу параллельно с основной.

3. Обе ЭВМ работают в автономном режиме со своим набором ПФУ по автономным рабочим программам.

Задание режимов работы вычислительного комплекса возможно программным путем или с помощью команд прямого управления или с пульта управления комплекса.

По *типу организации* многомашинные ВК можно разделить на две группы:

- несвязанные ВК - разрабатывались с целью разгрузить центральный процессор от выполнения операций по вводу-выводу данных извне. Они состоят из центральной и периферийной ЭВМ, между которыми нет прямого физического соединения и отсутствуют какие-либо совместно исполняемые аппаратные средства. Целесообразность их применения определяется тем, что операции ввода-вывода информации и вычисления совмещаются во времени. Небольшая и недорогая ЭВМ выполняет медленные операции ввода-вывода информации (считывание с перфокарт, печать и т.п.), а центральная ЭВМ — высокоскоростные операции, обмениваясь в процессе вычислений с ВЗУ.

- связанные ВК - включают несколько ЭВМ, которые совместно используют общие аппаратные средства, т.е. в этих ВС возможно электрическое сопряжение между процессорами. В таких ВК обе ЭВМ могут выполнять две различные программы автономно или во взаимодействии друг с другом.

Для реализации межмашинной связи могут использоваться как средства, имеющиеся в составе ЭВМ, так и средства, предусмотренные специально для работы в составе данного ВК.

## 4.3. Многопроцессорные вычислительные системы

В настоящее время особое внимание уделяется созданию многопроцессорных ВК. Основной целью при разработке таких ВК является повышение производительности систем за счет: обеспечения возможности параллельного выполнения независимых задач; повышения эффективности работы и улучшения распределения нагрузки в системе; обеспечения наиболее экономичного обслуживания экстренных заданий и заданий при пиковых нагрузках; достижения высокого коэффициента эффективного использования ресурсов для создания новых типов архитектуры комплекса.

В многопроцессорных ВС при решении задач с небольшими емкостями памяти возможно одновременное решение на разных процессорах. Если в какой-либо интервал времени требуется резкое увеличение емкости памяти, то вся память отдается для решения задачи.

Основные особенности построения многопроцессорных ВК заключаются в следующем:

- система включает в себя один или несколько процессоров;
- центральная память системы должна находиться в общем пользовании и к ней должен быть обеспечен доступ от всех процессоров системы;
- система должна иметь общий доступ ко всем устройствам ввода-вывода, включая каналы;
- система должна иметь единую ОС, управляющую всеми аппаратными и программными средствами;
- в системе должно быть предусмотрено взаимодействие элементов аппаратного и программного обеспечения на всех уровнях: на уровне системного программного обеспечения, на программном уровне при решении задач пользователей (возможность перераспределения заданий), на уровне обмена данными и др.

В многомашинных ВС связь может осуществляться только на информационном уровне.

Важнейшее значение для организации многопроцессорной ВС имеют способы соединения между собой различных функциональных блоков системы, так как эффективность такой системы определяется степенью параллельности или совмещения по времени работы всех устройств системы.

## Тема 5. Сети компьютеров

### 5.1. Основные понятия

*Сеть ЭВМ* – комплекс аппаратного и программного обеспечения, поддерживающий функции обмена информацией между отдельно расположенными (на расстояниях от нескольких метров до тысяч километров) компьютерами. *Сеть с централизованным управлением* содержат одну или более выделенных ЭВМ (серверов), управляющих обменом по сети (остальные ЭВМ в этом случае называются *рабочими станциями*), *одноранговая сеть* не содержит выделенных машин (функции управления сетью осуществляются рабочими станциями поочередно).

*Программное обеспечение компьютерных сетей* – комплекс программ, поддерживающий функции обмена информацией между отдельно расположенными ЭВМ.

*Локальная вычислительная сеть* (ЛВС) – система связи отдельно расположенных ЭВМ на относительно небольшом расстоянии (обычно в пределах помещения и/или этажа здания); объединяет до нескольких десятков компьютеров, физическая линия связи - двухпроводной кабель или коаксиальный кабель.

*Корпоративная вычислительная сеть* – сеть, работающая по протоколу TCP/IP и необязательно подключенная к Internet, но использующая коммуникационные стандарты Internet'а и сервисные приложения, обеспечивающие доставку данных пользователям сети; эксплуатируется в пределах крупной организации.

*Глобальная вычислительная сеть* (ГВС) объединяет множество локальных сетей и сотни тысяч разнотипных ЭВМ по всему миру, физическая линия связи - оптокабель или космическая радиолиния связи.

*Рабочая группа (workgroup)* – набор компьютеров, объединенных при просмотре сетевых ресурсов одним именем.

*Домен (domain)* – определенная администратором сети совокупность компьютеров, использующих общую базу данных и систему защиты; каждый домен имеет уникальное имя.

*Узел (host)* – подключенное к сети устройство, идентифицируемое собственным *адресом* (например, в сети Internet host-адресом является уникальное 32-разрядное двоичное число).

*Скорость передачи данных* по компьютерной сети измеряется в *битах в секунду* (bps - *bit per second*) или *боддах (boud)*.

*Трафик (traffic)* – поток сообщений в разделяемой среде передачи данных, часто используется для грубой оценки уровня использования передающей среды (тяжелый, средний, легкий трафик).

*Серверная ЭВМ* – компьютер, обычно обладающий высоким быстродействием и значительным объемом оперативной и дисковой памяти, выполняющий запросы, поступающие с *клиентских ЭВМ*.

*Файл-сервер* – выделенная ЭВМ, выполняющая функции хранения данных и программ, используемых пользователями на *клиентских ЭВМ*.

*Серверное приложение* – выполняющееся ЭВМ приложение, которое может исполнять запросы, генерируемые другим *приложением-клиентом* (выполняющемся на данной или удаленной ЭВМ).

*Клиентская ЭВМ* – пользовательский компьютер (обычно обладающий ограниченными ресурсами), выдающий запросы для исполнения серверу.

*Клиентское приложение* – приложение, обращающееся с целью выполнения отдельных функций к другому *приложению-серверу*.

*Протокол (коммуникационный)* – набор правил и соглашений, согласно которому взаимодействуют два или более компьютеров.

*Топология (topology) сети* – физическая конфигурация машин в сети.

*Временное уплотнение* при передаче данных – метод передачи данных по линии связи, основанный на последовательной по времени передаче пакетов данных, причем каждый пакет снабжен маркером (в состав которого входит адрес машины-получателя пакета и некоторая дополнительная информация).

*Маршрутизация* – процесс определения оптимального пути доступа к объектам сети.

*Пакет (датаграмма)* – определенное количество байт, сгруппированное вместе и посылаемое одновременно.

### 5.2. Локальные вычислительные сети

Понятие *локальная вычислительная сеть* (англ. LAN - Lokal Area Network) относится к географически ограниченными аппаратно-программным реализациям, в которых несколько компьютерных систем связаны друг с другом с помощью соответствующих средств коммуникаций. Благодаря такому соединению пользователь может взаимодействовать с другими рабочими станциями, подключенными к этой ЛВС.

Преимущества, получаемые при сетевом объединении персональных компьютеров в виде внутрипроизводственной вычислительной сети:

- *разделение ресурсов* позволяет экономно использовать ресурсы, например, управлять периферийными устройствами, такими как лазерные печатающие устройства, со всех присоединенных рабочих станций.

- *разделение данных* предоставляет возможность доступа и управления базами данных с периферийных рабочих мест, нуждающихся в информации.

- *разделение программных средств* предоставляет возможность одновременного использования централизованных, ранее установленных программных средств.

- *разделение ресурсов процессора* делает возможным использование вычислительных мощностей для обработки данных другими системами, входящими в сеть. Предоставляемая возможность заключается в том, что на имеющиеся ресурсы не “набрасываются” моментально, а только лишь через специальный процессор, доступный каждой рабочей станции.

- *многопользовательский режим* содействует одновременному использованию централизованных прикладных программных средств, ранее установленных и управляемых, например, если пользователь системы работает с другим заданием, то текущая выполняемая работа отодвигается на задний план.

Все ЛВС работают в одном стандарте принятом для компьютерных сетей - в стандарте Open Systems Interconnection (OSI).

### 5.2.1. Базовая модель OSI (Open System Interconnection)

Для единого представления данных в линиях связи, по которым передается информация, сформирована Международная организация по стандартизации (англ. ISO - International Standards Organization).

ISO предназначена для разработки модели международного коммуникационного протокола, в рамках которой можно разрабатывать международные стандарты. Международная организация по стандартизации разработала базовую модель взаимодействия открытых систем. Модель является международным стандартом для передачи данных и содержит семь отдельных уровней.

Основная идея этой модели заключается в том, что каждому уровню отводится конкретная роль в том числе и транспортной среде. Благодаря этому общая задача передачи данных расчленяется на отдельные легко обозримые задачи. Необходимые соглашения для связи одного уровня с выше- и нижерасположенными называют *протоколом*.

Так как пользователи нуждаются в эффективном управлении, система вычислительной сети (ВС) представляется как комплексное строение, которое координирует взаимодействие задач пользователей.

1) **Физический** - на данном уровне определяются электрические, механические, функциональные и процедурные параметры для физической связи в системах. Физическая связь и неразрывная с ней эксплуатационная готовность являются основной функцией 1-го уровня. Стандарты физического уровня включают рекомендации V.24 МККТТ (CCITT), EIA RS232 и X.21. Стандарт ISDN (Integrated Services Digital Network) в будущем сыграет определяющую роль для функций передачи данных. В качестве среды передачи данных используют трехжильный медный провод (экранированная витая пара), коаксиальный кабель, оптоволоконный проводник и радиорелейную линию.

2) **Канальный уровень** формирует из данных, передаваемых 1-м уровнем последовательности кадров. На этом уровне осуществляются управление доступом к передающей среде, используемой несколькими ЭВМ, синхронизация, обнаружение и исправление ошибок.

3) **Сетевой уровень** устанавливает связь в ВС между двумя абонентами. Соединение происходит благодаря функциям маршрутизации, которые требуют наличия сетевого адреса в пакете. Сетевой уровень должен также обеспечивать обработку ошибок, мультиплексирование, управление потоками данных.

4) **Транспортный уровень** поддерживает непрерывную передачу данных между двумя взаимодействующими друг с другом пользовательскими процессами. Качество транспортировки, безошибочность передачи, независимость ВС, сервис транспортировки из конца в конец, минимизация затрат и адресация связи гарантируют непрерывную и безошибочную передачу данных.

5) **Сеансовый уровень** координирует прием, передачу и выдачу одного сеанса связи. Для координации необходимы контроль рабочих параметров, управление потоками данных промежуточных накопителей и диалоговый контроль, гарантирующий передачу, имеющихся в распоряжении данных. Кроме того, сеансовый уровень содержит функции управления паролями, подсчета платы за пользование ресурсами сети, управления диалогом, синхронизации и отмены связи в сеансе передачи после сбоя вследствие ошибок в нижерасположенных уровнях.

6) Уровень **представления данных** предназначен для интерпретации данных; а также подготовки данных для пользовательского прикладного уровня. На этом уровне происходит преобразование данных из кадров, используемых для передачи данных в экранный формат или формат для печатающих устройств оконечной системы.

7) В **прикладном уровне** необходимо предоставить в распоряжение пользователей уже переработанную информацию. С этим может справиться системное и пользовательское прикладное программное обеспечение.

## 5.2.2. Сетевые устройства и средства коммуникаций

В качестве средств коммуникации наиболее часто используются витая пара, коаксиальный кабель оптоволоконные линии. При выборе типа кабеля учитывают следующие показатели:

- стоимость монтажа и обслуживания,
- скорость передачи информации,
- ограничения на величину расстояния передачи информации (без дополнительных усилителей-повторителей (репитеров)),
- безопасность передачи данных.

Главная проблема заключается в одновременном обеспечении этих показателей, например, наивысшая скорость передачи данных ограничена максимально возможным расстоянием передачи данных, при котором еще обеспечивается требуемый уровень защиты данных. Легкая наращиваемость и простота расширения кабельной системы влияют на ее стоимость.

**Витая пара** (twisted pair) является наиболее дешевым кабельным соединением (витое двухжильное проводное соединение). Она позволяет передавать информацию со скоростью до 10 Мбит/с, легко наращивается, однако является помехозащищенной. Длина кабеля не может превышать 1000 м при скорости передачи 1 Мбит/с. Преимуществами являются низкая цена и легкость установки. Для повышения помехозащищенности информации часто используют экранированную витую пару, т.е. витую пару, помещенную в экранирующую оболочку, подобно экрану коаксиального кабеля. Это увеличивает стоимость витой пары и приближает ее цену к цене коаксиального кабеля.

**Коаксиальный кабель** имеет среднюю цену, хорошо помехозащищен и применяется для связи на большие расстояния (несколько километров). Скорость передачи информации от 1 до 10 Мбит/с, а в некоторых случаях может достигать 50 Мбит/с. Коаксиальный кабель используется для основной и широкополосной передачи информации.

**Широкополосный коаксиальный кабель** невосприимчив к помехам, легко наращивается, но цена его высокая. Скорость передачи информации равна 500 Мбит/с. При передаче информации в базисной полосе частот на расстояние более 1,5 км требуется усилитель, или так называемый репитер (повторитель). Поэтому суммарное расстояние при передаче информации увеличивается до 10 км. Для ВС с топологией шина или дерево коаксиальный кабель должен иметь на конце согласующий резистор (терминатор).

**Ethernet-кабель** также является коаксиальным кабелем с волновым сопротивлением 50 Ом. Его называют еще толстый Ethernet (thick) или желтый кабель (yellow cable). Он использует 15-контактное стандартное включение. Вследствие помехозащищенности является дорогой альтернативой обычным коаксиальным кабелям. Максимально доступное расстояние без повторителя не превышает 500 м, а общее расстояние сети Ethernet - около 3000 м. Ethernet-кабель, благодаря своей магистральной топологии, использует в конце лишь один нагрузочный резистор.

**Cheapernet-кабель** или, как его часто называют тонкий (thin) Ethernet является более дешевым, чем Ethernet-кабель. Это также 50-омный коаксиальный кабель со скоростью передачи информации в десять миллионов бит/с.

При соединении сегментов Cheapernet-кабеля также требуются повторители. ВС с Cheapernet-кабелем имеют небольшую стоимость и минимальные затраты при наращивании. Соединения сетевых плат производится с помощью широко используемых малогабаритных байонетных разъемов (CP-50). Дополнительное экранирование не требуется. Кабель присоединяется к ПК с помощью тройниковых соединителей (T-connectors).

Расстояние между двумя рабочими станциями без повторителей может составлять максимум 300 м, а общее расстояние для сети на Cheapernet-кабеля - около 1000 м. Приемопередатчик Cheapernet

расположен на сетевой плате и как для гальванической развязки между адаптерами, так и для усиления внешнего сигнала.

**Оптоволоконные линии** являются наиболее дорогими. Скорость распространения информации по ним достигает нескольких гигабит в секунду. Допустимое удаление более 50 км. Внешнее воздействие помех практически отсутствует. Применяются там, где возникают электромагнитные поля помех или требуется передача информации на очень большие расстояния без использования повторителей. Они обладают противоподслушивающими свойствами, так как техника ответвлений в оптоволоконных кабелях очень сложна. Оптопроводники объединяются в ЛВС с помощью звездообразного соединения.

Существует ряд принципов построения ЛВС на основе выше рассмотренных компонентов. Такие принципы еще называют - *топологиями*.

### 5.2.3. Топологии ВС

#### **Топология типа звезда**

Концепция топологии сети в виде звезды пришла из области больших ЭВМ, в которой головная машина получает и обрабатывает все данные с периферийных устройств как активный узел обработки данных. Этот принцип применяется в системах передачи данных, например, в электронной почте RELCOM. Вся информация между двумя периферийными рабочими местами проходит через центральный узел ВС (рис. 5.1).

Пропускная способность сети определяется вычислительной мощностью узла и гарантируется для каждой рабочей станции. Коллизий (столкновений) данных не возникает.

Кабельное соединение довольно простое, так как каждая рабочая станция связана с узлом. Затраты на прокладку кабелей высокие, особенно когда центральный узел географически расположен не в центре топологии.



Рис. 5.1. Топология в виде звезды

При расширении ВС не могут быть использованы ранее выполненные кабельные связи: к новому рабочему месту необходимо прокладывать отдельный кабель из центра сети.

Топология в виде звезды является наиболее быстродействующей из всех топологий ВС, поскольку передача данных между рабочими станциями проходит через центральный узел (при его хорошей производительности) по отдельным линиям, используемым только этими рабочими станциями. Частота запросов передачи информации от одной станции к другой невысокая по сравнению с достигаемой в других топологиях.

#### **Кольцевая топология**

При кольцевой топологии сети рабочие станции связаны одна с другой по кругу, т.е. рабочая станция 1 с рабочей станцией 2, рабочая станция 3 с рабочей станцией 4 и т.д. Последняя рабочая станция связана с первой. Коммуникационная связь замыкается в кольцо (рис. 5.2).

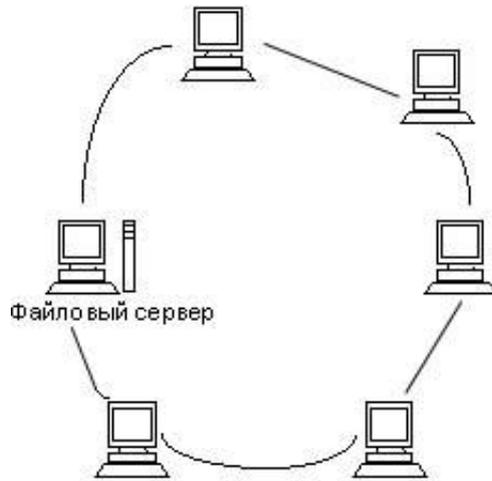


Рис. 5.2. Кольцевая топология.

Прокладка кабелей от одной рабочей станции до другой может быть довольно сложной и дорогостоящей, особенно если географически рабочие станции расположены далеко от кольца (например, в линию).

Сообщения циркулируют регулярно по кругу. Рабочая станция посылает по определенному конечному адресу информацию, предварительно получив из кольца запрос. Пересылка сообщений является очень эффективной, так как большинство сообщений можно отправлять “в дорогу” по кабельной системе одно за другим. Очень просто можно сделать кольцевой запрос на все станции. Продолжительность передачи информации увеличивается пропорционально количеству рабочих станций, входящих в ВС.

Основная проблема при кольцевой топологии заключается в том, что каждая рабочая станция должна активно участвовать в пересылке информации, и в случае выхода из строя хотя бы одной из них вся сеть парализуется. Неисправности в кабельных соединениях локализуются легко.

Подключение новой рабочей станции требует краткосрочного выключения сети, так как во время установки кольцо должно быть разомкнуто. Ограничения на протяженность ВС не существует, так как оно, в конечном счете, определяется исключительно расстоянием между двумя рабочими станциями.

### ***Шинная топология***

При шинной топологии (рис. 5.4) среда передачи информации представляется в форме коммуникационного пути, доступного для всех рабочих станций, к которому они все должны быть подключены. Все рабочие станции могут непосредственно вступать в контакт с любой рабочей станцией, имеющейся в сети. Рабочие станции в любое время, без прерывания работы всей ВС, могут быть подключены к ней или отключены. Функционирование ВС не зависит от состояния отдельной рабочей станции.



Рис. 5.4. Шинная топология

В стандартной ситуации для шинной сети Ethernet часто используют тонкий кабель или Cheapernet-кабель с тройниковым соединителем. Выключение и особенно подключение к такой сети требуют разрыва шины, что вызывает нарушение циркулирующего потока информации и зависание системы.

Новые технологии предлагают пассивные штепсельные коробки, через которые можно отключать и/или включать рабочие станции во время работы ВС.

Благодаря тому, что рабочие станции можно включать без прерывания сетевых процессов и коммуникационной среды, очень легко прослушивать информацию, т.е. отвлекать информацию из коммуникационной среды.

### ***Древовидная структура ЛВС***

Наряду с известными топологиями ВС кольцо, звезда и шина, на практике применяется и комбинированная, например, древовидная структура (рис. 5.5). Она образуется в основном в виде комбинаций вышеназванных топологий ВС. Основание дерева ВС располагается в точке (корень), в которой собираются коммуникационные линии информации (ветви дерева).

ВС с древовидной структурой применяются там, где невозможно непосредственное применение базовых сетевых структур в чистом виде. Для подключения большого числа рабочих станций соответственно адаптерным платам применяют сетевые усилители и/или коммутаторы. Коммутатор, обладающий функциями усилителя, называют активным концентратором. На практике применяют две их разновидности, обеспечивающие подключение соответственно восьми или шестнадцати линий.

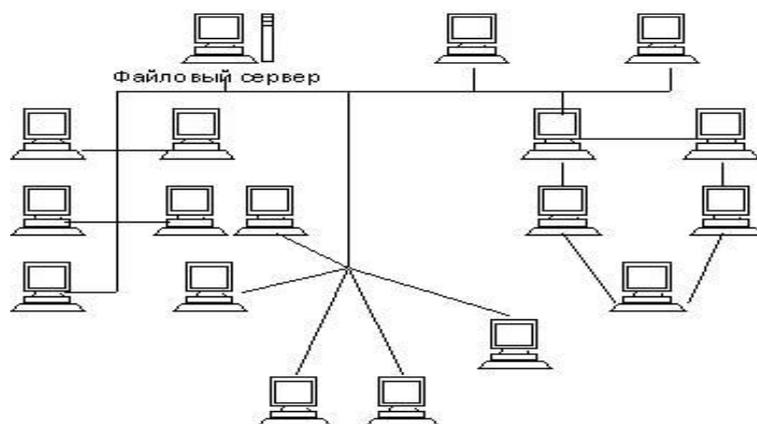


Рис. 5.5. ВС с древовидной структурой

Устройство к которому можно присоединить максимум три станции, называют пассивным концентратором, который обычно используют как разветвитель и не нуждается в усилителе. При подключении пассивного концентратора максимальное расстояние до рабочей станции не должно превышать нескольких десятков метров.

## 5.2.4. Типы построения сетей по методам передачи информации

### ***Локальная сеть Ethernet***

Спецификацию Ethernet в конце семидесятых годов предложила компания Xerox Corporation. Позднее к этому проекту присоединились компании Digital Equipment Corporation (DEC) и Intel Corporation. В 1982 году была опубликована спецификация на Ethernet версии 2.0. На базе Ethernet институтом IEEE был разработан стандарт IEEE 802.3. Различия между ними незначительные.

На логическом уровне в Ethernet применяется топология шина:

- все устройства, подключенные к сети, равноправны, т.е. любая станция может начать передачу в любой момент времени (если передающая среда свободна);
- данные, передаваемые одной станцией, доступны всем станциям сети.

### ***Сети промышленной автоматизации.***

Поскольку аппаратные средства автоматизированных систем управления технологическим процессом (АСУТП) могут быть территориально разнесены, то для их взаимодействия нужна специальная вычислительная сеть, называемая *промышленной сетью* (или ***Fieldbus***). В число узлов сети входят компьютеры, выполняющие функции NC и SCADA. Это могут быть обычные персональные компьютеры и специализированные программируемые логические контроллеры (ПЛК), называемые *промышленными компьютерами*. Специфика ПЛК - наличие нескольких аналоговых и цифровых портов, встроенный интерпретатор специализированного языка, детерминированные задержки при обработке сигналов, требующих незамедлительного реагирования. Однако, ПЛК в отличие от IBM

РС, рассчитаны на решение ограниченного круга задач в силу специализированности программного обеспечения.

В целом промышленные компьютеры имеют следующие особенности:

- 1) работа в режиме реального времени (для промышленных персональных компьютеров разработаны такие ОС реального времени, как OS-9, QNX, VRTX и др.);
- 2) конструкция, приспособленная для работы ЭВМ в цеховых условиях (повышенные вибрации, электромагнитные помехи, запыленность, перепады температур, иногда взрывоопасность);
- 3) возможность встраивания дополнительных блоков управляющей, регистрирующей, сопрягающей аппаратуры, что помимо специальных конструкторских решений обеспечивается использованием стандартных шин и увеличением числа плат расширения;
- 4) автоматический перезапуск компьютера в случае "зависания" программы;
- 5) повышенные требования к надежности функционирования. В значительной мере специализация промышленных компьютеров определяется программным обеспечением. Конструктивно промышленный компьютер представляет собой корзину (*крейт*) с несколькими гнездами (*слотами*) для встраиваемых плат. Возможно использование мостов между крейтами. В качестве стандартных шин в настоящее время преимущественно используются шины VME-bus (Versabus Module Europe-bus) и PCI (Peripheral Component Interconnect).

VME-bus - системная шина для создания распределенных систем управления на основе встраиваемого оборудования (процессоры, накопители, контроллеры ввода/вывода). Представляет собой расширение локальной шины компьютера на несколько гнезд объединительной платы (до 21 слота), возможно построение многомастерных систем, т.е. систем, в которых ведущими могут быть два или более устройств. Имеет 32-разрядные немultipлексируемые шины данных и адресов, возможно использование multipлексируемой 64-разрядной шины. Пропускная способность шины 320 Мбайт/с.

PCI - более удобная шина для однопроцессорных архитектур, получает все большее распространение. Пропускная способность до 264 Мбайт/с, разрядность шины 2x32 и/или при multipлексировании 64, архитектура с одним ведущим устройством. Имеется ряд разновидностей шины, например, шина CompaqPCI, в которой унифицирован ряд геометрических и механических параметров (так называемая Евромеханика).

Программная связь с аппаратурой нижнего уровня (датчиками, исполнительными устройствами) происходит через драйверы. Межпрограммные связи реализуются через интерфейсы, подобные OLE. Для упрощения создания систем разработан стандарт OPC (OLE for Process Control).

Обычными для промышленных сетей являются предельные расстояния между узлами (датчиками, исполнительными устройствами и контроллерами) в сотни метров, размеры сообщений - до одного килобайта (в сжатой форме). Опрос датчиков периодический. Важное требование к промышленной сети - обеспечение работы в реальном масштабе времени, поэтому для АСУТП сети типа Ethernet не подходят, поскольку в них не гарантируется ограничение задержек сверху.

Пример промышленной сети - *Profibus*, скорость 12 Мбод, пакеты до 247 байт, расстояния до 1,5 км. Имеет выход в сеть АСУТП, в качестве которой чаще всего используется сеть Ethernet. Наряду с Profibus, используют и другие протоколы, например, популярен протокол CAN. На физическом уровне в Fieldbus часто используют интерфейс RS-485 - витая пара, длина сегмента до 1,2 км, на сегменте может быть до 32 узлов.

### 5.3. Сетевой и транспортный уровни

#### 5.3.1. Транспортные и сетевые протоколы

В территориальных и корпоративных сетях со сложной разветвленной структурой для передачи сообщений от отправителя к адресату имеется много альтернативных маршрутов. Эти маршруты, как правило, включают не только конечные узлы отправителя и получателя, но и ряд промежуточных узлов и сетей.

Проблема обеспечения передачи информации между сетями, т.е. проблема обеспечения взаимодействия различных сетей в составе интегрированной сети, в англоязычной литературе носит название Internetworking. Это взаимодействие выражено функциями транспортного и сетевого уровней в семиуровневой модели ISO.

Функции *транспортного* уровня реализуются в конечных узлах. К ним относятся:

- разделение пакета на дейтаграммы, если сеть работает без установления соединения;
- сборка сообщений из дейтаграмм в узле-получателе;

- обеспечение заданного уровня услуг, включающих заказ времени доставки, типа канала связи, возможности сжатия данных с частичной потерей информации (как, например в алгоритме JPEG);
- управление сквозными соединениями в сети с помощью специальных команд запроса соединения, разъединения, передачи, приема, регистрации и др.

Назначение *сетевых* протоколов - приспособление пакетов к особенностям промежуточных сетей и выбор направления передачи пакетов (маршрутизация). В список основных функций входят:

- формирование пакетов с учетом требований промежуточных сетей (дополнение пакетов транспортного уровня обрамлением, включающим флаги, сетевые адреса получателя и отправителя, служебную информацию);
- управление потоками;
- маршрутизация;
- обнаружение неисправностей;
- ликвидация "заблудившихся" дейтаграмм и т.п.

Наиболее широко используемыми протоколами на сетевом уровне являются протоколы IP (Internet Protocol), X.25, IPX (Internet Packet Exchange) и на транспортном уровне TCP (Transmission Control Protocol) и SPX (Sequence Packet Exchange). Последние входят в систему протоколов TCP/IP и SPX/IPX соответственно. Протоколы TCP/IP первоначально были разработаны для сети ARPANET, а затем на их основе стала развиваться сеть Internet. Протоколы SPX/IPX разработаны и применяются фирмой Novell для сетей Novell Netware, объединяющих персональные ЭВМ. Протоколы X.25 разработаны ITU и включают части для физического, канального и сетевого уровней.

### 5.3.2. Управление потоками данных в сетях

Это одна из функций сетевого уровня, включающая управление нагрузками и борьбу с блокировками. Различают несколько уровней управления.

*Межузловое управление* связано с распределением буферной памяти в промежуточных узлах (выделением каждому направлению определенного числа буферов), сводящееся к ограничению длин канальных очередей.

*Управление "вход-выход"* направлено на предотвращение блокировок. Реализуется указанием в первом пакете сообщения его длины, что позволяет приемному узлу прогнозировать заполнение памяти и запрещать прием дейтаграмм определенных сообщений, если прогнозируется блокировка памяти.

*Управление внешними потоками* (доступом) реализуется путем предоставления приоритета в передаче внутренним потокам перед внешними, ограничением числа пакетов в сети (пакет принимается, если у узла есть соответствующее разрешение), посылкой предупредительных пакетов-заглушек в адрес источника, от которого идут пакеты в перегруженную линию связи.

### 5.3.3. Мостовые соединения

Серверы делятся на функциональные и на серверы доступа. Последние входят в число блоков взаимодействия сетей (подсетей).

*Блок взаимодействия* называют функциональный блок, обеспечивающий взаимодействие нескольких информационных сетей или подсетей. К блокам взаимодействия относятся повторители, мосты, маршрутизаторы, шлюзы, концентраторы, модемы и др.

*Повторитель* (repeater) - блок взаимодействия, служащий для регенерации электрических сигналов, передаваемых между двумя сегментами ЛВС. Повторители используются, если реализация ЛВС на одном сегменте кабеля (отрезке, моноканале) не допускается из-за ограничений на расстояние или на число узлов, причем при условии, что в соседних сегментах используются один и тот же метод доступа и одни и те же протоколы. Трафик в сегментах, соединенных повторителем, - общий. Повторитель может быть многопортовым. Сигнал, пришедший на один из портов, повторяется на всех остальных портах.

*Мост* (bridge) - блок взаимодействия, служащий для соединения разных подсетей, которые могут иметь неодинаковые канальные протоколы.

При малых расстояниях между подсетями связь возможна через серверы подсетей, в которых размещаются интерфейсные платы, называемые внутренними мостами, и соответствующее сетевое программное обеспечение. При этом предполагается использование подсетей с идентичными протоколами, например, IPX, иначе нужно использовать маршрутизаторы или шлюзы. В частности, высокопроизводительную ЭВМ подключают к ЛВС через выделяемую для этих целей машину-шлюз.

Возможно применение внешних мостов - специально выделяемых узлов для целей сопряжения по одному в каждой из соединяемых подсетей. Внешние мосты обходятся дороже, но обеспечивают лучшие эксплуатационные характеристики. Важная функция мостов - ограничение трафика, так как локальный трафик одной подсети замыкается в ней, не проходя в другую подсеть.

#### 5.3.4. Маршрутизация

Цель маршрутизации - доставка пакетов по назначению с максимизацией эффективности. Чаще всего эффективность выражена взвешенной суммой времен доставки сообщений при ограничении снизу на вероятность доставки. Маршрутизация сводится к определению направлений движения пакетов в маршрутизаторах. Выбор одного из возможных в маршрутизаторе направлений зависит от текущей топологии сети (она может меняться хотя бы из-за временного выхода некоторых узлов из строя), длин очередей в узлах коммутации, интенсивности входных потоков и т.п.

Алгоритмы маршрутизации включают процедуры:

- измерение и оценивание параметров сети;
- принятие решения о рассылке служебной информации;
- расчет таблиц маршрутизации (ТМ);
- реализация принятых маршрутных решений.

В зависимости от того, используется ли при выборе направления информация о состоянии только данного узла или всей сети, различают алгоритмы *изолированные* и *глобальные*. Если ТМ реагируют на изменения состояния сети, то алгоритм *адаптивный*, иначе *фиксированный* (статический), а при редких корректировках - *квазистатический*. В статических маршрутизаторах изменения в ТМ вносит администратор сети.

*Простейший алгоритм* - изолированный, статический. *Алгоритм кратчайшей очереди* в отличие от простейшего является адаптивным, пакет посылается по направлению, в котором наименьшая очередь в данном узле. *Лавинный алгоритм* - многопутевой, основан на рассылке копий пакета по всем направлениям, пакеты сбрасываются, если в данном узле другая копия уже проходила. Очевидно, что лавинный алгоритм обеспечивает надежную доставку, но порождает значительный трафик и потому используется только для отдельных пакетов большой ценности.

#### 5.3.5. Транспортный протокол TCP в стеке протоколов TCP/IP

Эти протоколы берут свое начало от одной из первых территориальных сетей ARPANET. Они получили широкое распространение благодаря реализации в ОС Unix и в сети Internet и в настоящее время оформлены в виде стандартов RFC (Requests For Comments) организацией IETF (Internet Engineering Task Force).

TCP/IP - пятиуровневые протоколы, но основными среди них, давшими название всей совокупности, являются протоколы сетевого (IP - Internet Protocol) и транспортного (TCP - Transport Control Protocol) уровней.

TCP - дуплексный транспортный протокол с установлением соединения. Его функции: упаковка и распаковка пакетов на концах транспортного соединения; установление виртуального канала путем обмена запросом и согласием на соединение; управление потоком - получатель при подтверждении правильности передачи сообщает размер окна, т.е. диапазон номеров пакетов, которые получатель готов принять; помещение срочных данных между специальными указателями, т.е. возможность управлять скоростью передачи.

В TCP имеется программа-демон, которая постоянно готова к работе и при приходе запроса генерирует свою копию для обслуживания создаваемого соединения, а сама программа-родитель ждет новых вызовов.

Структура TCP-пакета (в скобках указано число битов):

- порт отправителя (16);
- порт получателя (16);
- код позиции в сообщении, т.е. порядковый номер первого байта в поле данных (32);
- номер следующего байта (32);
- управление (16);
- размер окна (16), т.е. число байт, которое можно послать до получения подтверждения;
- контрольная сумма (16);
- дополнительные признаки, например срочность передачи (16);
- опции (24);
- заполнитель (8);

– данные.

### 5.3.6. Коммутационное оборудование

*Маршрутизатор (router)* - блок взаимодействия, служащий для выбора маршрута передачи данных в корпоративных и территориальных сетях. С помощью маршрутизаторов могут согласовываться не только каналные протоколы, как это имеет место при применении мостов, но и сетевые протоколы. Маршрутизаторы содержат таблицы и протоколы маршрутизации в отличие от других узлов (узлы, имеющие адреса и, следовательно, доступные по сети, называют хостами), которые могут содержать лишь локальные таблицы соответствия IP-адресов физическим адресам сетевых контроллеров в локальной сети.

*Шлюз (gateway - межсетевой преобразователь)* - блок взаимодействия, служащий для соединения информационных сетей различной архитектуры и с неодинаковыми протоколами. В шлюзах предусматривается согласование протоколов всех семи уровней. Примерами шлюзов могут быть устройства, соединяющие ЛВС типа Ethernet с сетью SNA, используемой для связи больших машин фирмы IBM. Часто под шлюзом понимают сервер, имеющий единственный внешний канал передачи данных.

*Концентраторы* предназначены для объединения в сеть многих узлов. Так, концентраторами являются хабы в 10Base-T или Token Ring. Хабы могут быть пассивными или активными, в последнем случае в хабе имеются усилители-формирователи подводимых сигналов. Однако такие концентраторы создают общую среду передачи данных без разделения трафика.

Коммутаторы в отличие от концентраторов предназначены для объединения в сеть многих узлов или подсетей с возможностью создания одновременно многих соединений. Они называются также переключателями (свитчами - switches). Коммутаторы используются также для связи нескольких ЛВС с территориальной сетью. Один коммутатор может объединять несколько как однотипных, так и разнотипных ЛВС.

Использование коммутаторов вместо маршрутизаторов (там, где это возможно) позволяет существенно повысить пропускную способность сети. Коммутатор работает с локальными MAC-адресами, в нем имеется таблица соответствия MAC-адресов и портов. Кроме того, между разными портами коммутатора образуется несколько соединений, по которым пакеты могут передаваться одновременно. В то же время маршрутизатор оперирует IP-адресами и таблицами маршрутизации и выполняет сложные алгоритмы маршрутизации.

Возможны коммутация "на лету" (сквозная коммутация - out-trough), когда передача пакета начинается сразу после расшифровки заголовка, и с полным получением пакета (промежуточная буферизация - store-and-forward). Первый способ применяют в небольших сетях, второй - в магистральных коммутаторах. Сквозная коммутация позволяет обойтись малым объемом буфера, но не дает возможности контролировать безошибочность передачи данных.

*Виртуальная ЛВС (ВЛВС)* - это локальная сеть, в которой узлы группируются не по территориальному, а по функциональному признаку. Для этого каждая подсеть в ВЛВС получает свой идентификатор, каждому идентификатору соответствуют определенные порты коммутаторов сети. Идентификатор указывается в заголовке кадра (структура кадра в ВЛВС задается стандартом IEEE 802.10) и поэтому коммутатор направляет кадр в нужную подсеть. Администратор сети может управлять структурой сети (перекоммутацией портов) с помощью специального ПО.

Лидером в производстве коммутаторов для ВЛВС является фирма Cisco. Ее коммутаторы семейства Catalyst допускают объединение в ВЛВС до 1024 подсетей FDDI, E, TR, ATM. Встроенные программы управления позволяют закреплять любой порт за любой подсетью.

К блокам взаимодействия относят также модемы, *мультиплексоры и демультиплексоры* - устройства для преобразования сообщений в кадры TDM (временное мультиплексирование) и обратно.

### 5.3.7. Сетевой протокол IP в стеке протоколов TCP/IP.

IP - дейтаграммный сетевой протокол без установления соединения. Его функции: фрагментация и сборка пакетов при прохождении через промежуточные сети, имеющие другие протоколы; маршрутизация; проверка контрольной суммы заголовка пакета (правильность передачи всего пакета проверяется на транспортном уровне, т.е. с помощью TCP, в оконечном узле); управление потоком - сброс дейтаграмм при превышении заданного времени жизни.

Структура дейтаграммы в IP (в скобках указано число битов):

- версия протокола IP (4);
- длина заголовка (4);
- тип сервиса (8);
- общая длина (16);

- идентификация (16) - порядковый номер дейтаграммы;
- место фрагмента в дейтаграмме (16) - указывает номер фрагмента при фрагментации дейтаграммы в промежуточных сетях;
- время жизни дейтаграммы в сети (8);
- тип протокола (8);
- контрольный код CRC заголовка (16);
- адрес источника (32);
- адрес назначения (32);
- опции (32);
- данные (не более 65356 байт).

От версии протокола зависит структура заголовка. Сделано это для возможности последующего внесения изменений. Например, предполагается вместо четырехбайтовых адресов установить в Internet в будущем шестибайтовые адреса.

### 5.3.8. Другие протоколы в стеке TCP/IP

В состав протокола IP входит ряд частных протоколов. Среди них протоколы ARP, IGP, EGP, относящиеся к маршрутизации на разных иерархических уровнях в архитектуре сети. На одном уровне с IP находится протокол управления ICMP (Internet Control Message Protocol).

Протокол ARP (Address Resolution Protocol) относится к связям "хост-хост" или "хост-шлюз" в конкретной подсети. Он использует локальные таблицы маршрутизации - ARP-таблицы, устанавливающие соответствие IP-адресов с NPA (Network Point of Attachment) адресами серверов доступа в соответствующих подсетях. В подсетях не нужно рассчитывать кратчайший путь и определять маршрут в разветвленной сети, что, естественно, ускоряет доставку. ARP-таблицы имеются в каждом узле. Если в таблице отправителя нет строки для IP-адреса получателя, то отправитель сначала посылает широковещательный запрос. Если некоторый узел имеет этот IP-адрес, он откликается своим NPA, и отправитель пополняет свою таблицу и отправляет пакет. Иначе отправка пакета произойдет на внешний порт сети.

Протокол IGP (Interior Gateway Pr.) предназначен для управления маршрутизацией в некотором домене (автономной сети - AS), т.е. он определяет маршруты между внутренними сетями домена. Другими словами, в AS имеется (или может быть получена) информация о путях ко всем сетям домена, и протокол IGP доставляет дейтаграмму в нужную подсеть в соответствии с алгоритмом маршрутизации RIP или OSPF.

Протокол EGP (Exterior Gateway Pr.) относится к корневой сети и предназначен для управления маршрутизацией между внешними шлюзами и пограничными маршрутизаторами доменов.

В TCP/IP входит также протокол UDP (User Datagram Protocol) - транспортный протокол без установления соединения, он значительно проще TCP, но используется чаще всего для сообщений, уместающихся в один пакет. После оформления UDP-пакета он передается с помощью средств IP к адресату, который по заголовку IP-пакета определяет тип протокола и передает пакет не агенту TCP, а агенту UDP. Агент определяет номер порта и ставит пакет в очередь к этому порту. В UDP служебная часть дейтаграммы короче, чем в TCP (8 байт вместо 20), не требуется предварительного установления соединения или подтверждения правильности передачи, как это делается в TCP, что и обеспечивает большую скорость за счет снижения надежности доставки.

Структура UDP-дейтаграммы (в скобках указано число битов):

- порты отправителя и получателя (по 16 битов);
- длина (16);
- контрольная сумма (16);
- данные (не более 65,5 тыс. байт).

Часто считают, что в состав TCP/IP входят также протоколы высоких уровней такие, как:

- SMTP (Simple Mail Transport Protocol) - почтовый протокол, который по классификации ISO можно было бы отнести к прикладному уровню;

- FTP (File Transfer Protocol) - протокол с функциями представительного уровня;

- Telnet - протокол с функциями сеансового уровня.

Протоколы TCP/IP являются основными протоколами сети Internet, они поддерживаются операционными системами Unix и Windows NT.

На нижних уровнях в TCP/IP используется протокол IEEE 802.X или X.25.

### 5.3.9. Адресация в TCP/IP (в Internet)

Различают два типа адресов. На канальном уровне используют адреса, называемые физическими. Это шестибайтовые адреса сетевых плат, присваиваемые изготовителем контроллеров (каждый изготовитель вместе с лицензией на изготовление получает уникальный диапазон адресов). На сетевом уровне используют сетевые адреса, иначе называемые *виртуальными*, или *логическими*. Эти адреса имеют иерархическую структуру, для них существуют цифровое и буквенное выражения.

Узлы в Internet имеют адрес и имя. Адрес - уникальная совокупность чисел: адреса сети и компьютера (хоста - узла в сети), которая указывает их местоположение. Имя характеризует пользователя. Оно составляется в соответствии с доменной системой имен. Соответствие между IP-адресом и IP-именем хоста устанавливается специальной *службой директорий*. В Internet это DNS (Domain Name Service), в ISO - стандарт X.500.

*IP-имя*, называемое также *доменным именем*, - удобное для человека название узла или сети. Имя отражает иерархическое построение глобальных сетей и потому состоит из нескольких частей (аналогично обычным почтовым адресам). Корень иерархии обозначает либо страну, либо отрасль знаний, например: *ru* - Россия, *us* - США, *de* - Германия, *uk* - Великобритания, *edu* - наука и образование, *com* - коммерческие организации, *org* - некоммерческие организации, *gov* - правительственные организации, *mil* - военные ведомства, *net* - служба поддержки Internet и т.д. Корень занимает в *IP*-имени правую позицию, левее записываются локальные части адреса и, наконец, перед символом @ указывается имя почтового ящика пользователя. Так, запись *norenkov@rk6.bmstu.ru* расшифровывается, как пользователь *norenkov* в подразделении *rk6* организации *bmstu* в стране *ru*.

*IP-адрес* - 32-битовое слово, записываемое в виде четырех частей (побайтно), разделенных точками. Каждая подсеть и узел в подсети получают свои номера, причем для сети (подсети) может использоваться от одного до трех старших байтов, а оставшиеся байты - для номера узла. Какая часть IP-адреса относится к сети, определяется ее маской, выделяющей соответствующие биты в IP-адресе. Например, для некоторой сети маска может быть 255.0.0.0, а для ее подсети - 255.255.0.0 и т.д. Тем самым описывается иерархия сетей.

Номера при включении нового хоста выдает организация, предоставляющая телекоммуникационные услуги и называемая *провайдером*. Провайдер, в частности, обеспечивает включение IP-адреса и соответствующего ему IP-имени в сервер службы адресов DNS. Это означает запись данных о хосте в DIB (Directory Information Base) локального узла DNS.

### 5.3.10. Распределенные вычисления (РВ)

Различают режимы удаленного узла и дистанционного управления (рис. 5.9).

В режиме *удаленного узла* основные процедуры приложения исполняются на терминальном узле (local node), а с удаленным узлом (remote node) связь используется для пересылки файлов. В большинстве случаев режим удаленного узла приводит к более заметной инерционности связи через телефонные каналы. Дистанционное управление обеспечивает передачу клавишных команд в прямом направлении и экранных изображений (обычно лишь изменений в них) в сжатом виде в обратном направлении, поэтому задержки меньше.

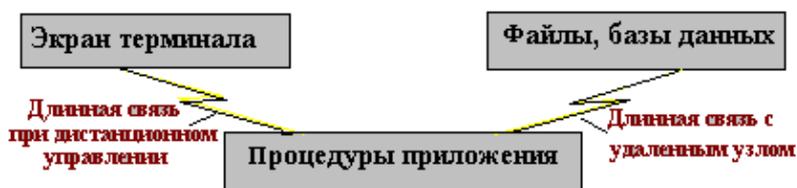


Рис. 5.9. Удаленный узел и дистанционное управление

Системы распределенных вычислений основаны на режиме *дистанционного* управления, при котором терминальный узел используется только для интерфейса с пользователем и передачи команд управления, а основные процедуры приложения исполняются на удаленном узле (сервере). Поэтому в сетях распределенных вычислений должны быть выделены серверы приложений. При организации РВ решаются вопросы размещения функций по узлам сети. В зависимости от того, между какими взаимодействующими частями РВ имеется длинная связь, различают четыре модели распределенных вычислений (рис. 5.10):

- файловый сервер (FS - File Server);
- доступ к удаленным данным (RDA - Remote Data Access);
- сервер баз данных (DBS - Data Base Server);
- сервер приложений (ApS - Application Server).

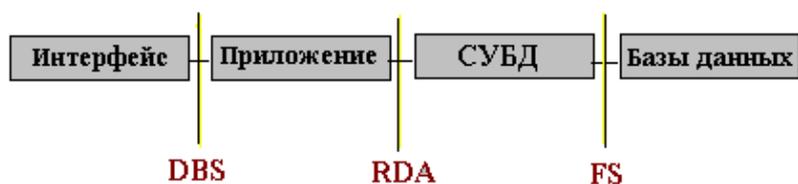


Рис. 5.10. Варианты распределенных вычислений

FS - основная модель для ЛВС на персональных ЭВМ. В случае ее использования возникает проблема корректного обновления файлов. Все процессы клиентов и серверов имеют маркеры, содержащие имя файла и маску, в которой указаны права: только чтение атрибутов файла, только чтение самого файла, открытие файла, модификация файла, стирание. Все обращения идут через менеджер маркеров, который отслеживает соблюдение ограничений и разрешает конфликты одновременного обращения для чтения и обновления файлов. Недостаток FS - перегрузка сети из-за необходимости пересылать файлы полностью.

Положительные стороны RDA: уменьшение трафика, унификация интерфейса с сервером на базе языка SQL.

### 5.3.11. Технологии распределенных вычислений

ПО организации распределенных вычислений называют ПО *промежуточного слоя* (Middleware). Новое направление организации распределенных вычислений в сетях Internet-Intranet основано на создании и использовании программных средств, которые могут работать в различных аппаратно-программных средах. Совокупность таких средств называют *многоплатформенной распределенной средой* - MPC (crossware).

Находят применение технологии распределенных вычислений RPC (Remote Procedure Call), ORB (Object Request Broker), MOM (Message-oriented Middleware), DCE (Distributed Computing Environment), мониторы транзакций, ODBC.

RPC - процедурная блокирующая синхронная технология, предложенная фирмой Sun Microsystems. Вызов удаленных программ подобен вызову функций в языке C. При пересылках на основе транспортных протоколов TCP или UDP данные представляются в едином формате обмена XDR. Синхронность и блокирование означают, что клиент, обратившись к серверу, для продолжения работы ждет ответа от сервера.

Для систем распределенных вычислений разработаны специальные языки программирования, для RPC это язык IDL (Interface Definition Language), который дает пользователю возможность оперировать различными объектами безотносительно к их расположению в сети. На этом языке можно записывать обращения к серверам приложений. Другой пример языка для систем распределенных вычислений - NewEra в среде Informix.

## 5.4. Структура и функции территориальных сетей

### 5.4.1. Структура территориальных сетей

Глобальная сеть Internet - самая крупная и единственная в своем роде сеть в мире. Среди глобальных сетей она занимает уникальное положение. Правильнее ее рассматривать как некоторую надсеть - объединение многих сетей, сохраняющих самостоятельное значение. Действительно, Internet не имеет ни четко выраженного владельца, ни национальной принадлежности. Любая сеть может иметь связь с Internet и, следовательно, рассматриваться как ее часть, если в ней используются принятые для Internet протоколы TCP/IP или имеются конверторы в протоколы TCP/IP. Практически все сети национального и регионального масштабов имеют выход в Internet.

Типичная территориальная (национальная) сеть имеет иерархическую структуру.

Верхний уровень - федеральные узлы, связанные между собой магистральными каналами связи. Магистральные каналы физически организуются на ВОЛС или на спутниковых каналах связи. Средний уровень - региональные узлы, образующие региональные сети. Они связаны с федеральными узлами и, возможно, между собой выделенными высоко- или среднескоростными каналами, такими, как каналы T1, E1, B-ISDN или радиорелейные линии. Нижний уровень - местные узлы (серверы доступа), связанные с региональными узлами преимущественно коммутируемыми или выделенными телефонными каналами связи, хотя заметна тенденция к переходу к высоко- и среднескоростным каналам. Именно к местным узлам подключаются локальные сети малых и средних предприятий, а

также компьютеры отдельных пользователей. Корпоративные сети крупных предприятий соединяются с региональными узлами выделенными высоко- или среднескоростными каналами.

Иерархическая архитектура Internet может быть представлена рисунком 5.15.

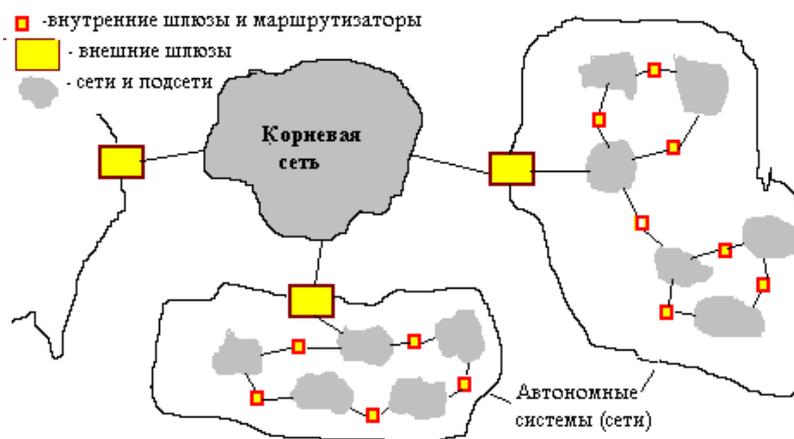


Рис. 5.15. Иерархическая структура территориальной сети

Внутри каждой автономной системы (AS) используется некоторый единый внутренний протокол маршрутизации, например IGP. Между AS маршрутизация подчиняется внешним протоколам, например EGP.

Основные услуги телекоммуникационных технологий - электронная почта, передача файлов, телеконференции, справочные службы (доски объявлений), видеоконференции, доступ к информационным ресурсам (информационным базам) сетевых серверов, мобильная сотовая связь, компьютерная телефония.

#### 5.4.2. Протоколы теледоступа

Специфика телекоммуникаций проявляется прежде всего в прикладных протоколах. Среди них наиболее известны протоколы, связанные с Internet, и протоколы ISO-IP (ISO 8473), относящиеся к семиуровневой модели открытых систем. К прикладным протоколам Internet относятся следующие:

Telnet - протокол эмуляции терминала, или, другими словами, протокол реализации дистанционного управления используется для подключения клиента к серверу при их размещении на разных компьютерах, пользователь через свой терминал имеет доступ к компьютеру-серверу;

FTP - протокол файлового обмена (реализуется режим удаленного узла), клиент может запрашивать и получать файлы с сервера, адрес которого указан в запросе;

HTTP (Hypertext Transmission Protocol) - протокол для связи WWW-серверов и WWW-клиентов;

NFS - сетевая файловая система, обеспечивающая доступ к файлам всех UNIX-машин локальной сети, т.е. файловые системы узлов выглядят для пользователя, как единая файловая система;

SMTP, IMAP, POP3 - протоколы электронной почты.

Указанные протоколы реализуются с помощью соответствующего программного обеспечения. Для Telnet, FTP, SMTP на серверной стороне выделены фиксированные номера протокольных портов.

В семиуровневой модели ISO используются аналогичные протоколы. Так, протокол VT соответствует протоколу Telnet, FTAM - FTP, MOTIS - SMTP, CMIP - SNMP, протокол RDA (Remote Database Access) предназначен для доступа к удаленным БД.

#### 5.4.3. Электронная почта

Электронная почта (E-mail) - средство обмена сообщениями по электронным коммуникациям (в режиме off-line). Можно пересылать текстовые сообщения и архивированные файлы. Функции клиента - составление, отправление, архивирование сообщений.

Разработан ряд альтернативных протоколов E-mail для прикладного уровня. Наиболее популярны среди них протоколы SMTP в стеке протоколов TCP/IP и X.400 в модели ISO. Расширение числа возможных кодировок и форматов данных по сравнению с SMTP сделано в MIME (Multipurpose Internet Mail Extensions). На их базе разработано ПО E-mail, способное работать в обоих

протоколах. Оно включает программы почтовых серверов и клиентов. Применение MIME упрощает пересылку графических и звуковых файлов, реализацию шифрования и электронной подписи.

На ЭВМ пользователя должна быть установлена программа-клиент, поддерживающая названные выше функции создания, передачи и приема сообщений. На почтовом сервере, выделяемом в корпоративной или локальной сети, организуется промежуточное хранение поступающих сообщений. Связь индивидуальных пользователей с почтовым сервером осуществляется по протоколам IMAP или POP3. Для индивидуального пользователя, общающегося с другими абонентами по телефонной сети общего пользования, такое промежуточное хранение возможно в собственном компьютере, но тогда требуется либо круглосуточное включение компьютера, либо предварительная договоренность о времени связи.

В территориальных сетях почтовые сообщения проходят через ряд промежуточных федеральных или региональных узлов. В таких узлах устанавливается программное обеспечение (так называемый агент передачи сообщений), выполняющее функции сортировки и маршрутизации сообщений.

Примерами программных систем электронной почты, выполняющих все отмеченные функции E-mail, могут служить Lotus cc: mail, Microsoft Mail или Outlook Express. Они позволяют адресовать сообщения индивидуальному пользователю; на доску объявлений; на последовательный просмотр несколькими исполнителями с возможностями коррекции сообщения; поиск сообщений, прошедших в почтовый сервер, по контексту, по адресу, по времени отправки.

#### 5.4.4. Файловый обмен

Файловый обмен - доступ к файлам, распределенным по различным компьютерам. В сети Internet на прикладном уровне используется протокол FTP. Доступ возможен в режимах off-line и on-line. В режиме off-line посылается запрос к FTP-серверу, сервер формирует и посылает ответ на запрос. В режиме on-line осуществляется интерактивный просмотр каталогов FTP-сервера, выбор и передача нужных файлов. На ЭВМ пользователя нужен FTP-клиент.

Доступ к БД удаленных ЭВМ с эмуляцией удаленной машины на ЭВМ пользователя выполняется с помощью протокола Telnet.

Протокол файлового обмена FTP. При запросе файла по протоколу FTP пользователь должен знать, где находится нужный ему файл. Для этого удобно воспользоваться другой информационной системой сети Internet, называемой Archie. Обращаясь к клиенту Archie по команде archie <имя файла>, пользователь получает в ответ адрес сервера, имя директории и размер файла. Далее можно обращаться к FTP-клиенту:

```
ftp[<параметры>][<имя сервера>]. (5.1)
```

Квадратные скобки в записи команд означают необязательные части. Параметры используются только при отладке FTP. В качестве имени сервера указывается IP-имя или IP-адрес удаленного компьютера.

В большинстве серверов Internet для входа по FTP-команде нужны предварительная регистрация пользователя и указание пароля. Однако это не требуется при обращениях к общедоступным (анонимным) серверам. Такие серверы создают и обслуживают организации, заинтересованные в распространении информации определенного вида.

#### 5.4.5. Информационная система WWW

WWW (World Wide Web - всемирная паутина) - гипертекстовая информационная система сети Internet. Другое ее краткое название - Web. Это более современная система по сравнению с Gopher и предоставляет пользователям большие возможности.

Во-первых, это гипертекст - структурированный текст с введением в него перекрестных ссылок, отражающих смысловые связи частей текста. Слова-ссылки выделяются цветом и/или подчеркиванием. Выбор ссылки вызывает на экран связанный со словом-ссылкой текст или рисунок. Можно искать нужный материал по ключевым словам.

Во-вторых, облегчено представление и получение графических изображений. К 1996 г. в мире насчитывалось около 30 тысяч WWW-серверов.

Информация, доступная по Web-технологии, хранится в Web-серверах. Сервер имеет программу Listener, постоянно отслеживающую приход на определенный порт (обычно это порт 80) запросов от клиентов. Сервер удовлетворяет запросы, посылая клиенту содержимое запрошенных Web-страниц или результаты выполнения запрошенных процедур.

Клиентские программы WWW называют браузерами (browsers). Имеются текстовые (например, Lynx) и графические (наиболее известны Netscape Navigator и MS Explorer) браузеры. Sun предлагает браузер HotJava. В браузерах имеются команды листания, перехода к предыдущему или следующему документу, печати, перехода по гипертекстовой ссылке и т.п. Из браузеров доступны различные сервисы - FTP, Gopher, USENET, E-mail. Для подготовки материалов для их включения в базу WWW разработаны специальный язык HTML (Hypertext Markup Language) и реализующие его программные редакторы, например Internet Assistant в составе редактора Word или SiteEdit, подготовка документов предусмотрена и в составе большинства браузеров.

Для связи Web-серверов и клиентов разработан протокол HTTP, работающий на базе TCP/IP. Web-сервер получает запрос от браузера, находит соответствующий запросу файл и передает его для просмотра в браузер. Популярными серверами являются Apache Digital для ОС Unix, Netscape Enterprise Server и Microsoft Internet Information Server (IIS), которые могут работать как в Unix, так и в Windows NT, и Netware Web Server, предназначенный для работы в ОС Netware. Все три сервера поддерживают язык CGI, имеют встроенный HTML-редактор. Кроме того, в первых двух из них поддерживается стандарт шифрования SSL (Secure Sockets Layer) для защиты передаваемых по сети данных от несанкционированного доступа. Опыт показывает, что для крупных серверов предпочтительнее платформа Unix, тогда как для серверов с малым числом транзакций лучше подходит ОС Windows NT.

На базе HTML создан язык виртуальной реальности VRML (Virtual Reality Modeling Language)- в нем дополнительно можно использовать 3D графику.

## **Тема 6. Микроконтроллеры**

### **6.1. Определение микроконтроллера**

В микропроцессорной технике выделяется самый большой класс больших интегральных схем (БИС) - однокристалльные микроЭВМ (ОМЭВМ). Их архитектура является результатом эволюции архитектуры микропроцессоров и микропроцессорных систем, обусловленной стремлением существенно снизить их аппаратные затраты и стоимость.

ОМЭВМ представляют собой приборы, конструктивно выполненные в виде одной БИС и включающие в себя все устройства, необходимые для реализации цифровых систем управления минимальной конфигурации: процессор, запоминающее устройство данных, запоминающее устройство команд, внутренний генератор тактовых сигналов, а также программируемую интегральную схему для связи с внешней средой.

В некоторых публикациях ОМЭВМ называют "микроконтроллерами". Обосновано это тем обстоятельством, что такие микросхемы имеют незначительную емкость памяти, физическое и логическое разделение памяти программ ПЗУ и памяти данных ОЗУ, упрощенную и ориентированную на задачи управления систему команд и данных, т.е. такие ОМЭВМ предназначены не для решения задач обработки данных, а для решения задач управления и регулирования в приборах, устройствах и системах автоматизации.

Микроконтроллеры - это первая ступень на пути, который прошли информационные технологии до современных суперкомпьютеров, но и эти маленькие кристаллы могут многое. К микроконтроллерам и однокристалльным микро-ЭВМ относят сегодня те первые процессоры, на основе которых создавались первые персональные компьютеры. Уменьшившись с течением времени в размерах и приобретя функциональное многообразие, эти процессоры в новом качестве служат целям обработки информации и управления.

Использование однокристалльных микроконтроллеров позволяет разработчику устройства воплотить найденное решение конкретной, специфической задачи при минимальных затратах по стоимости и размерам. Типичной для микроконтроллеров является ситуация, когда на обслуживание микросхемы отведена ничтожная доля выводов, тогда как основная масса используется непосредственно для нужд задачи - ввода, сбора информации и вывода - управления.

Быстрое развитие и распространение однокристалльных микроконтроллеров обусловлено их универсальностью и низкой стоимостью. Современные микроконтроллеры обладают всеми ресурсами для решения задач управления, при этом производятся в таком разнообразии, что могут удовлетворить любые потребности. Сейчас они являются средством реализации блочного подхода к построению устройств электронной техники.

Вот краткий перечень изделий, построенных на базе микроконтроллеров:

Микро-АТС, автоответчики, АОНЫ, мобильные телефоны, зарядные устройства, факсы, модемы, пейджеры, таймеры, системы сигнализации, измерительные приборы, счетчики воды, газа и электроэнергии, дозиметры, приборы авто сигнализации, системы управления зажиганием и впрыском топлива, приборные панели и радарные детекторы, интеллектуальные датчики, системы управ-

ления электродвигателями, промышленные роботы, регуляторы температуры, влажности, давления и т. д., схемы управления принтерами и плоттерами, сетевые контроллеры, сканеры, схемы управления аудиосистемами, системы синтеза речевых сообщений, видеоигры, системы дистанционного управления, кассовые аппараты и т. д.

## 6.2. Классификация микроконтроллеров

Номенклатура предлагаемых на сегодняшнем мировом рынке микроконтроллеров (МК) чрезвычайно обширна и в своем составе содержит определенное множество семейств или крупных групп МК, выпускаемых различными производителями. Семейства между собой различаются:

- по архитектуре процессорного ядра;
- набору интегрируемых на кристалл МК модулей памяти и периферийных устройств;
- разрядности обрабатываемых данных (4, 8, 16, 32, 64 бит);
- быстродействию;
- «возрасту» разработки;
- энергопотреблению и ряду других характеристик.

В пределах семейства, как правило, микроконтроллеры подразделяются:

- на подгруппы или серии.

Различие между МК, входящих в состав подгруппы заключается, главным образом, в вариациях встроенной памяти. Серии же между собой различаются, в основном, составом и параметрами встроенных периферийных устройств, определяющих специфику применения МК.

В число периферийных устройств обычно входят параллельные порты ввода/вывода, каналы последовательного интерфейса, таймеры и таймеры/счетчики, контроллер прерываний и т.п. Кроме этого, многие микроконтроллеры содержат дополнительные запоминающие устройства, многофункциональные таймерные модули, многоканальные АЦП и ЦАП, широтно-импульсные модуляторы, цифровые компараторы, устройства для подключения жидкокристаллических и электролюминесцентных индикаторов и другие устройства.

Наиболее крупными и известными на мировом рынке производителями микроконтроллеров являются такие фирмы как Intel, Motorola, Microchip, Atmel, Zilog, AMD, Philips, Dallas Semiconductor, Texas Instruments, Hitachi, Mitsubishi, Fujitsu, NEC, Toshiba, Holtek, Uicom (Scenix), Infineon (Siemens), Temic, STMicroelectronics, Samsung, Triscend и ряд других американских, европейских, южно-азиатских и японских фирм. Отечественная промышленность выпускала весьма ограниченный спектр микроконтроллеров, большая часть которых относилась к Intel-совместимым (серии K1816, K1830, K1835 – аналоги семейств MCS-48 и MCS-51).

По статистике, в области встраиваемых систем управления наибольшую долю рынка занимают восьмиразрядные микроконтроллеры. Но, вместе с тем, для решения многих современных задач вычислительной мощности 8-разрядных микроконтроллеров уже довольно часто оказывается недостаточно, поэтому в настоящее время начинает активно развиваться и рынок 16- и 32-разрядных микроконтроллеров.

## 6.3. Тенденции в развитии современных микроконтроллеров

Следует отметить несколько тенденций в развитии современных микроконтроллеров:

1. В настоящее время происходит «сближение» структур обычных микроконтроллеров и цифровых сигнальных процессоров (DSP – Digital Signal Processors). Так, в обычные микроконтроллеры интегрируют модули быстрых АЦП, модули скоростной арифметики и т.п., а DSP оснащают дополнительными возможностями, характерными для управляющих микроконтроллеров, например, расширяют периферию, добавляют «битовые» команды для работы с портами и т.п.

Лидером в производстве DSP является фирма Texas Instruments, несколько отстают от нее Analog Devices, Motorola и AT&T.

2. Совсем недавно о себе заявило новое направление – «системы на кристалле» (System on Chip – SoC). Изделия данного класса именуют также FPSLIC (Field Programmable System Level Integration Circuits), что примерно можно перевести как программируемые пользователем микросхемы системного уровня интеграции. Идея данного направления и основное его достоинство базируются на том, что технологически стало возможным разместить на одном кристалле фиксированное сложное микропроцессорное ядро, включая оперативную память для хранения программ и данных, а также достаточный объем программируемой логики для специализации кристалла пользователем под конкретную задачу.

Ведущими фирмами, развивающими это направление, являются Atmel, Triscend, Lucent Technologies и LSI Logic.

3. Заслуживает внимания ряд МК, который анонсируется как «новый промышленный стандарт», в рамки которого можно свести микроконтроллеры с резидентной Flash-памятью программ со свойством программируемости в системе (In-System Programming – ISP). Такого рода микроконтроллеры можно найти в продукции фирм Motorola, Atmel, Philips и SST.

Одним из немаловажных параметров микроконтроллера является тип и размер внутренней памяти программ.

Разные фирмы предлагают МК с различными вариантами исполнения данной памяти. Так, например, фирмы Motorola и Microchip выпускают значительную долю МК с однократным программированием (OTPROM – One-Time Programmable ROM), а для целей отладки предлагают кристаллы ППЗУ с ультрафиолетовым стиранием (EPROM). Другие фирмы, например, Atmel, Philips, SST взяли курс на широкое внедрение технологии Flash и SuperFlash при реализации памяти программ в производстве микроконтроллеров.

Сейчас появился новый тип энергонезависимой памяти – Ferroelectric RAM. Данный вид памяти характеризуется малым временем обращения, по сравнению с традиционной Flash, и очень большим количеством циклов перезаписи (1012). Необходимо заметить, что из-за особенностей технологии FRAM различий между циклом записи и чтения нет, поэтому количество циклов чтения также ограничено. Тем не менее, внедрение данной технологии позволит расширить области использования и более свободного применения энергонезависимой памяти.

#### 6.4. Оценка быстродействия микроконтроллеров

Одной из важных характеристик МК является их быстродействие. Кратко приведем некоторые замечания по данному поводу:

1. При оценке быстродействия необходимо проводить анализ всевозможных вариантов построения системы синхронизации МК, поскольку некоторые современные МК имеют «на борту» схему синтезатора частот на основе системы импульсно-фазовой автоподстройки частоты (ФАПЧ или PLL – Phase-Locked Loop), с помощью которой можно формировать внутреннюю тактовую частоту гораздо выше внешней.

Яркой иллюстрацией могут служить МК семейства HC08 фирмы Motorola. Например, при работе с внешним «часовым» кварцевым резонатором (32,768 кГц) внутреннюю частоту тактирования центрального процессора и внутренних магистралей ( $f_{bus}$ ) можно программным путем варьировать от 2 до 8 МГц. Еще один пример – микроконтроллеры семейства F<sup>2</sup>MC – 16xx фирмы Fujitsu. При внешней частоте синхронизации 4 МГц, внутреннюю частоту можно программно менять от 2 до 16 МГц.

2. Необходимо учитывать и архитектуру самого микропроцессорного ядра. Так, наличие кэша приводит к увеличению быстродействия без увеличения частоты синхронизации. Микроконтроллер с RISC-архитектурой также будет работать быстрее, нежели с архитектурой CISC, на той же самой частоте в силу того, что его инструкции исполняются за меньшее количество тактов.

В качестве примера самых скоростных МК (есть и другие) можно упомянуть семейство SX фирмы Ubicom (Scenix), которые имеют полностью статическую КМОП реализацию, обеспечивающую работу при тактовых частотах от 0 до 50 МГц, причем их тестирование показывает, что большинство кристаллов при гарантированной частоте 50 МГц работает на частотах до 150 МГц. Высокое быстродействие МК позволило этой фирме реализовать идею так называемой «виртуальной периферии». Различные аппаратные периферийные устройства в этом случае реализуются в виде соответствующих программных модулей, библиотека которых доступна на сайте фирмы. Такой подход обеспечивает уменьшение числа компонентов, времени разработки, увеличивает гибкость проектов, что, в конечном счете, дает возможность разработчику достаточно быстро построить нужное периферийное устройство.

#### 6.5. Энергосберегающие микроконтроллеры ведущих производителей

Отметим еще одно важное свойство современных МК – это наличие возможности «опционального» программного управления работой («включением/выключением») внутренних ресурсов микроконтроллера и, как следствие, наличие нескольких режимов пониженного энергопотребления. В активном режиме (Run mode) все ресурсы МК находятся в рабочем состоянии. В энергосберегающих режимах производится либо отключение ЦП с сохранением работоспособности периферийных устройств и ОЗУ, либо полностью запрещается тактирование ЦП и периферийных модулей при сохранении данных в ОЗУ, либо осуществляется программное «опциональное» отключение определенного набора периферийных модулей.

Примером компаний, которые делают особый акцент на максимально возможное снижение энергопотребления для выпускаемых микроконтроллеров, могут служить такие фирмы как Texas Instruments, Microchip, Toshiba и Hitachi. Например, микроконтроллеры семейства H8 фирмы Hitachi поддерживают до семи различных режимов пониженного энергопотребления.

В заключении можно отметить, что для облегчения и интенсификации труда разработчиков микропроцессорных систем в настоящее время различными фирмами предлагается множество инструментальных средств, а именно, внутрисхемные эмуляторы, программные отладчики-симуляторы, мониторы отладки, эмуляторы ПЗУ, оценочные платы, платы развития и т.п. Базовое программное обеспечение для разработки программ микроконтроллеров обычно свободно распространяется и доступно на web-серверах фирм-производителей. Стандартный набор такого программного обеспечения представляет собой интегрированную среду проектирования, включающую в себя текстовый редактор, компилятор языка Ассемблер и программный отладчик-симулятор.

Таким образом, современный разработчик имеет возможность под каждое конкретное применение подобрать по совокупности необходимых параметров подходящий микроконтроллер и сформировать максимально возможный набор профессиональных инструментальных средств разработки и отладки.

### 4.3. Лабораторные работы

<i>№ п/п</i>	<i>Номер раздела дисципли- ны</i>	<i>Наименование лабораторной работы</i>	<i>Объем (час.)</i>	<i>Вид занятия в инте- рактивной, актив- ной, инновационной формах, (час.)</i>
1	2.	Программирование вычислений функций	6	разбор конкретных ситуаций 2 час.)
2	2.	Программирование циклов	6	разбор конкретных ситуаций 2 час.)
3	2.	Программирование ввода-вывода	6	разбор конкретных ситуаций 2 час.)
4	2.	Программирование функций	6	разбор конкретных ситуаций 2 час.)
5	2.	Программирование вычислений с условиями	6	разбор конкретных ситуаций 2 час.)
6	2.	Программирование прерываний	6	разбор конкретных ситуаций 2 час.)
<b>ИТОГО</b>			<b>36</b>	<b>12</b>

### 4.4. Практические занятия

Учебным планом не предусмотрено.

### 4.5. Контрольные мероприятия: курсовая работа

Цель: Познакомиться с основными понятиями алгебры логики; изучить основные операции булевой алгебры; получить практические навыки в построении таблиц истинности и булевых выражений. Изучить основные законы и соотношения булевой алгебры; получить практические навыки по преобразованию и упрощению булевых выражений методами непосредственных преобразований и карт Карно.

Структура: Каждое индивидуальное задание предполагает выполнение студентом следующих разделов:

- построение таблицы истинности булевой функции нескольких переменных,
- минимизация логической функции,
- переход от логической функции к логической схеме,
- синтез логических устройств в заданном базисе.

Основная тематика: Синтез логических функции и схем.

Рекомендуемый объем: Пояснительная записка объемом 15 - 20 страниц должна содержать титульный лист, задание, описание выполняемых действий по каждому разделу и полученные результаты.

Выдача задания, прием и защита проводится в соответствии с календарным учебным графиком

<b>Оценка</b>	<b>Критерии оценки</b>
отлично	Курсовая работа сдана в первую неделю защит. В курсовой работе полностью раскрыта тема работы. Минимизация функции выполнена самым оптимальным способом. Логическая схема нарисована аккуратно и четко. Синтез логических устройств произведен правильно.
хорошо	Курсовая работа сдана в срок со второй по четвертую неделю защит или курсовая работа содержит незначительные ошибки.
удовлетворительно	Курсовая работа сдана в срок с пятой недели по седьмую неделю защит или содержит значительное количество ошибок, или ошибка подразумевает полную переработку всей курсовой работы.
неудовлетворительно	Курсовая работа не сдана в установленный срок.

**5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

<i>№, наименование разделов дисциплины</i>	<i>Кол-во часов</i>	<i>Компетенции</i>		<i>Σ ко мп.</i>	<i>t<sub>ср</sub>, час</i>	<i>Вид учебных занятий</i>	<i>Оценка результатов</i>
		<i>ОК</i>	<i>ОПК</i>				
		<i>7</i>	<i>7</i>				
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>1.</b> Теоретические основы построения узлов ЭВМ	12	+	+	2	6	Лк, СР	Зачет, КР
<b>2.</b> Процессоры и микропроцессоры	48	+	+	2	24	Лк, ЛР, СР	Зачет
<b>3.</b> Организация ввода/вывода в ЭВМ	12	+	+	2	6	Лк, СР	Зачет
<b>4.</b> Вычислительные системы	12	+	+	2	6	Лк, СР	Зачет
<b>5.</b> Сети компьютеров	12	+	+	2	6	Лк, СР	Зачет
<b>6.</b> Микроконтроллеры	12	+	+	2	6	Лк, СР	Зачет
<i>всего часов</i>	<b>108</b>	<b>54</b>	<b>54</b>	<b>2</b>	<b>54</b>		

## 6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации : учебное пособие / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - М. : Кнорус, 2013. - 376 с., стр.50-320.

## 7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№	Наименование издания	Вид занятия	Количество экземпляров в библиотеке, шт.	Обеспеченность, (экз./ чел.)
1	2	3	4	5
<b>Основная литература</b>				
1.	Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации : учебное пособие / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - М. : Кнорус, 2013. - 376 с.	Лк, ЛЗ	10	1
<b>Дополнительная литература</b>				
2.	Колтыгин, Д. С. Булева алгебра и логические элементы : методические указания к курсовой работе / Д. С. Колтыгин. - Братск : БрГУ, 2012. - 40 с.	КР	50	5

## 8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО - ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронный каталог библиотеки БрГУ  
[http://irbis.brstu.ru/CGI/irbis64r\\_15/cgiirbis\\_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=](http://irbis.brstu.ru/CGI/irbis64r_15/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=).
2. Электронная библиотека БрГУ  
<http://ecat.brstu.ru/catalog> .
3. Электронно-библиотечная система «Университетская библиотека online»  
<http://biblioclub.ru> .
4. Электронно-библиотечная система «Издательство «Лань»  
<http://e.lanbook.com> .
5. Информационная система "Единое окно доступа к образовательным ресурсам"  
<http://window.edu.ru> .
6. Научная электронная библиотека eLIBRARY.RU <http://elibrary.ru> .
7. Университетская информационная система РОССИЯ (УИС РОССИЯ)  
<https://uisrussia.msu.ru/> .
8. Национальная электронная библиотека НЭБ  
<http://xn--90ax2c.xn--p1ai/how-to-search/> .

## 9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

### 9.1. Методические указания для обучающихся по выполнению лабораторных работ

#### Лабораторная работа № 1

#### Программирование вычислений функций

#### Цель работы:

Изучить программирование вычислений различных функций.

Вид занятия в интерактивной, активной форме: выполнить задание и ознакомиться с составом и характеристиками.

Задание:

1. Изучить необходимые команды на языке Assembler.
2. Написать программу по заданию преподавателя.

Порядок выполнения:

На основании представленного задания определить состав операторов и команд. Написать программу по вычислению функции по заданию преподавателя.

Форма отчетности:

Отчет по лабораторной работе, скрепленный титульным листом. Отчет должен содержать название работы, цель, задание и результат выполнения задания.

Задания для самостоятельной работы:

Изучить структуру регистров процессора, команды для написания конкретного задания.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным во втором разделе данной дисциплины и учебном курсе (дополнительная литература).

Основная литература

1. Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации : учебное пособие / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - М. : Кнорус, 2013. - 376 с.

Дополнительная литература

1. Колтыгин, Д. С. Булева алгебра и логические элементы : методические указания к курсовой работе / Братск : БрГУ, 2012. - 40 с.

Контрольные вопросы для самопроверки

1. Что такое регистры пользовательские?
2. Что такое регистры системные?
3. Какие команды ассемблера необходимы для вашей программы?

## Лабораторная работа № 2

### Программирование циклов

Цель работы:

Изучить программирование циклических вычислений.

Вид занятия в интерактивной, активной форме: выполнить задание и ознакомиться с составом и характеристиками.

Задание:

1. Изучить необходимые команды на языке Assembler.
2. Написать программу по заданию преподавателя.

Порядок выполнения:

На основании представленного задания определить состав операторов и команд. Написать программу по вычислению функции по заданию преподавателя.

Форма отчетности:

Отчет по лабораторной работе, скрепленный титульным листом. Отчет должен содержать название работы, цель, задание и результат выполнения задания.

Задания для самостоятельной работы:

Изучить структуру регистров процессора, команды для написания конкретного задания.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным во втором разделе данной дисциплины и учебном курсе (дополнительная литература).

Основная литература

1. Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации : учебное пособие / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - М. : Кнорус, 2013. - 376 с.

Дополнительная литература

1. Колтыгин, Д. С. Булева алгебра и логические элементы : методические указания к курсовой работе / Братск : БрГУ, 2012. - 40 с.

Контрольные вопросы для самопроверки

1. Что такое цикл?
2. Что такое регистры задействованы при организации циклов?
3. Какие команды ассемблера необходимы для вашей программы?

**Лабораторная работа № 3**

Программирование ввода-вывода

Цель работы:

Изучить программирование ввода-вывода.

Вид занятия в интерактивной, активной форме: выполнить задание и ознакомиться с составом и характеристиками.

Задание:

1. Изучить необходимые команды на языке Assembler.
2. Написать программу по заданию преподавателя.

Порядок выполнения:

На основании представленного задания определить состав операторов и команд. Написать программу по вводу-выводу по заданию преподавателя.

Форма отчетности:

Отчет по лабораторной работе, скрепленный титульным листом. Отчет должен содержать название работы, цель, задание и результат выполнения задания.

Задания для самостоятельной работы:

Изучить структуру регистров процессора, команды для написания конкретного задания.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным во втором разделе данной дисциплины и учебном курсе (дополнительная литература).

Основная литература

1. Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации : учебное пособие / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - М. : Кнорус, 2013. - 376 с.

Дополнительная литература

1. Колтыгин, Д. С. Булева алгебра и логические элементы : методические указания к курсовой работе / Братск : БрГУ, 2012. - 40 с.

#### Контрольные вопросы для самопроверки

1. Что такое прерывание?
2. Что такое регистры задействованы при операции ввода-вывода?
3. Какие команды ассемблера необходимы для вашей программы?

### **Лабораторная работа № 4**

#### Программирование функций

#### Цель работы:

Изучить программирование функций.

Вид занятия в интерактивной, активной форме: выполнить задание и ознакомиться с составом и характеристиками.

#### Задание:

1. Изучить необходимые команды на языке Assembler.
2. Написать программу по заданию преподавателя.

#### Порядок выполнения:

На основании представленного задания определить состав операторов и команд. Написать программу по организации функции по заданию преподавателя.

#### Форма отчетности:

Отчет по лабораторной работе, скрепленный титульным листом. Отчет должен содержать название работы, цель, задание и результат выполнения задания.

#### Задания для самостоятельной работы:

Изучить структуру регистров процессора, команды для написания конкретного задания.

#### Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным во втором разделе данной дисциплины и учебном курсе (дополнительная литература).

Основная литература

1. Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации : учебное пособие / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - М. : Кнорус, 2013. - 376 с.

Дополнительная литература

1. Колтыгин, Д. С. Булева алгебра и логические элементы : методические указания к курсовой работе / Братск : БрГУ, 2012. - 40 с.

#### Контрольные вопросы для самопроверки

1. Как организовать работу с функциями?
2. Что такое стек?
3. Какие команды ассемблера необходимы для вашей программы?

### **Лабораторная работа № 5**

#### Программирование вычислений с условиями

Цель работы:

Изучить программирование вычислений различных функций с условиями.

Вид занятия в интерактивной, активной форме: выполнить задание и ознакомиться с составом и характеристиками.

Задание:

1. Изучить необходимые команды на языке Assembler.
2. Написать программу по заданию преподавателя.

Порядок выполнения:

На основании представленного задания определить состав операторов и команд. Написать программу по вычислению функции с условиями по заданию преподавателя.

Форма отчетности:

Отчет по лабораторной работе, скрепленный титульным листом. Отчет должен содержать название работы, цель, задание и результат выполнения задания.

Задания для самостоятельной работы:

Изучить структуру регистров процессора, команды для написания конкретного задания.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным во втором разделе данной дисциплины и учебном курсе (дополнительная литература).

Основная литература

1. Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации : учебное пособие / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - М. : Кнорус, 2013. - 376 с.

Дополнительная литература

1. Колтыгин, Д. С. Булева алгебра и логические элементы : методические указания к курсовой работе / Братск : БрГУ, 2012. - 40 с.

Контрольные вопросы для самопроверки

1. Что такое метка?
2. Какие команды условий вы знаете?
3. Какие команды ассемблера необходимы для вашей программы?

**Лабораторная работа № 6**

**Программирование прерываний**

Цель работы:

Изучить программирование прерываний.

Вид занятия в интерактивной, активной форме: выполнить задание и ознакомиться с составом и характеристиками.

Задание:

1. Изучить необходимые команды на языке Assembler.
2. Написать программу по заданию преподавателя.

Порядок выполнения:

На основании представленного задания определить состав операторов и команд. Написать программу по прерыванию по заданию преподавателя.

#### Форма отчетности:

Отчет по лабораторной работе, скрепленный титульным листом. Отчет должен содержать название работы, цель, задание и результат выполнения задания.

#### Задания для самостоятельной работы:

Изучить структуру регистров процессора, команды для написания конкретного задания.

#### Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным во втором разделе данной дисциплины и учебном курсе (дополнительная литература).

#### Основная литература

1. Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации : учебное пособие / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - М. : Кнорус, 2013. - 376 с.

#### Дополнительная литература

1. Колтыгин, Д. С. Булева алгебра и логические элементы : методические указания к курсовой работе / Братск : БрГУ, 2012. - 40 с.

#### Контрольные вопросы для самопроверки

1. Назовите команды работы со стекком.
2. Какие прерывания вы знаете?
3. Какие команды ассемблера необходимы для вашей программы?

## **9.2. Методические указания по выполнению курсовой работы**

В процессе выполнения курсовой работы студенты должны закрепить теоретические знания в области алгебры логики, основ булевой алгебры, получить практические навыки по преобразованию и упрощению булевых выражений методами непосредственных преобразований и карт Карно.

Проведение курсовой работы должно начинаться с вводного занятия, на котором преподаватель разбирает общую методику проведения работ, их цели и задачи.

Результаты работы должны быть оформлены в точном соответствии с разделом "Порядок оформления отчета по курсовой работе" методических указаний к курсовой работе, указанной в п.9.1. в дополнительной литературе.

Курсовая работа выполняется индивидуально. Отчет о выполненной работе представляется каждым студентом. При сдаче отчета преподаватель опрашивает студентов в объеме материала данной работы и контрольных вопросов.

Далее рассматриваются примеры для каждого пункта курсового проектирования.

## 10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Информационно-коммуникационные технологии (ИКТ) – преподаватель использует для:

- получения информации при подготовке к занятиям,
- создания презентационного сопровождения лекций;
- интерактивного общения;
- ОС Windows 7 Professional;
- Microsoft Office 2007 Russian Academic OPEN NO Level;
- Антивирусное программное обеспечение Kaspersky Security;
- ПО “Антиплагиат”

## 11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

<i>Вид занятия</i>	<i>Наименование аудитории</i>	<i>Перечень основного оборудования</i>	<i>№ ЛР или ПЗ</i>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
ЛР	Дисплейные классы	Персональные компьютеры	ЛЗ № 1-6
СР	ЧЗЗ	-	-

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ  
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

**1. Описание фонда оценочных средств (паспорт)**

№ компетенции	Элемент компетенции	Раздел	Тема	ФОС
ОК-7	способность к самоорганизации и самообразованию	1. Теоретические основы построения узлов ЭВМ	1.1 Физические формы представления информации 1.2 Математические модели схем ЭВМ	Вопросы к зачету
		2. Процессоры и микропроцессоры	2.1 Основные принципы построения устройств обработки цифровой информации 2.2. Принципы организации арифметико-логических устройств 2.3. Структура и формат команд. Кодирование команд 2.4. Способы адресации 2.5. Запоминающие устройства ЭВМ	Вопросы к зачету
		3. Организация ввода/вывода в ЭВМ	3.1. BIOS 3.2. Системные и локальные шины	Вопросы к зачету
		4. Вычислительные системы	4.1. Классификация вычислительных систем 4.2. Многомашинные вычислительные системы	Вопросы к зачету
		5. Сети компьютеров	5.1. Основные понятия 5.2. Локальные вычислительные сети 5.3. Сетевой и транспортный уровни	Вопросы к зачету
		6. Микроконтроллеры	6.1. Определение микроконтроллера 6.2. Классификация микроконтроллеров 6.3. Тенденции в развитии современных микроконтроллеров 6.4. Оценка быстродействия микроконтроллеров	Вопросы к зачету
ОПК-7	способность учитывать современные тен-	1. Теоретические основы построения узлов ЭВМ	1.3 Элементы и узлы ЭВМ	Вопросы к зачету

	денции развития электроники, измерительной и вычислительной техники, информационных технологий в своей профессиональной деятельности	2. Процессоры и микропроцессоры	2.6. Принципы организации систем обработки прерывания программ	Вопросы к зачету
		3. Организация ввода/вывода в ЭВМ	3.3. Шины ввода/вывода	Вопросы к зачету
		4. Вычислительные системы	4.3. Многопроцессорные вычислительные системы	Вопросы к зачету
		5. Сети компьютеров	5.4. Структура и функции территориальных сетей	Вопросы к зачету
		6. Микроконтроллеры	6.5. Энергосберегающие микроконтроллеры ведущих производителей	Вопросы к зачету

## 2. Вопросы к зачету

№ п/п	Компетенции		ВОПРОСЫ К ЗАЧЕТУ	№ и наименование раздела
	Код	Определение		
1	2	3	4	5
1.	ОК-7	способность к самоорганизации и самообразованию	1. Физические формы представления информации 2. Математические модели схем ЭВМ	1. Теоретические основы построения узлов ЭВМ
			3. Основные принципы построения устройств обработки цифровой информации 4. Принципы организации арифметико-логических устройств 5. Структура и формат команд. Кодирование команд 6. Способы адресации 7. Запоминающие устройства ЭВМ	2. Процессоры и микропроцессоры
			8. BIOS 9. Системные и локальные шины	3. Организация ввода/вывода в ЭВМ
			10. Классификация вычислительных систем 11. Многомашинные вычислительные системы	4. Вычислительные системы
			12. Основные понятия 13. Локальные вычислительные сети 14. Сетевой и транспортный уровни	5. Сети компьютеров
			15. Определение микроконтроллера 16. Классификация	6. Микроконтроллеры

			микроконтроллеров 17. Тенденции в развитии современных микроконтроллеров 18. Оценка быстродействия микроконтроллеров	
2.	ОПК-7	способность учитывать современные тенденции развития электроники, измерительной и вычислительной техники, информационных технологий в своей профессиональной деятельности	1. Элементы и узлы ЭВМ	1. Теоретические основы построения узлов ЭВМ
			2. Принципы организации систем обработки прерывания программ	2. Процессоры и микропроцессоры
			3. Шины ввода/вывода	3. Организация ввода/вывода в ЭВМ
			4. Многопроцессорные вычислительные системы	4. Вычислительные системы
			5. Структура и функции территориальных сетей	5. Сети компьютеров
			6. Энергосберегающие микроконтроллеры ведущих производителей	6. Микроконтроллеры

### 3. Описание показателей и критериев оценивания компетенций

Показатели	Оценка	Критерии
<p><b>Знать</b> (ОК-7):</p> <ul style="list-style-type: none"> <li>– основные этапы развития вычислительных систем;</li> </ul> <p>(ОПК-7):</p> <ul style="list-style-type: none"> <li>- достоинства и недостатки основных типов вычислительных машин и систем</li> </ul> <p><b>Уметь</b> (ОК-7):</p> <ul style="list-style-type: none"> <li>- организовать взаимодействие двух персональных компьютеров, персонального компьютера и периферийного устройства;</li> </ul> <p>(ОКП-7):</p> <ul style="list-style-type: none"> <li>- решать стандартные задачи по проектированию, настройке и обслуживанию распределённых вычислительных систем</li> </ul> <p><b>Владеть</b> (ОК-7):</p> <ul style="list-style-type: none"> <li>- навыками конфигурирования адресов стека TCP/IP</li> </ul> <p>(ОПК-7):</p> <ul style="list-style-type: none"> <li>– методами настройки и обслуживания коммуникационных устройств логической структуризации сети.</li> </ul>	<b>зачтено</b>	Оценка «зачтено» выставляется в случае, если студент демонстрирует: <ul style="list-style-type: none"> <li>– всестороннее систематическое знание программного материала;</li> <li>– правильное выполнение практических заданий, направленных на применение программного материала;</li> <li>– правильное применение основных положений программного материала.</li> </ul>
	<b>незачтено</b>	Оценка «незачтено» выставляется в случае, если студент демонстрирует: <ul style="list-style-type: none"> <li>– существенные пробелы в знании программного материала;</li> <li>– принципиальные ошибки при выполнении практических заданий, направленных на применение программного материала;</li> <li>– невозможность применения основных положений программного материала.</li> </ul>

#### **4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности**

Дисциплина Вычислительные машины, системы и сети направлена на изучение основ и состава вычислительных машин и систем, компьютерных сетей, методов контроля, обработки, анализа теоретических и экспериментальных исследований в сфере профессиональной деятельности.

Изучение дисциплины предусматривает:

- лекции,
- лабораторные занятия,
- самостоятельную работу,
- курсовую работу,
- зачет

В ходе освоения раздела 1 «Теоретические основы построения узлов ЭВМ» обучающиеся должны изучить основные понятия, определения, термины, существующие в вычислительной технике, изучить основы построения узлов ЭВМ.

В ходе освоения раздела 2 «Процессоры и микропроцессоры» обучающиеся должны знать принципы построения процессоров.

В ходе освоения раздела 3 «Организация ввода/вывода в ЭВМ» обучающиеся должны знать основные принципы организации ввода-вывода в ЭВМ.

В ходе освоения раздела 4 «Вычислительные системы» обучающиеся должны изучить классификацию, устройства вычислительных систем.

В ходе освоения раздела 5 «Сети компьютеров» обучающиеся должны знать, как устроены различные сети и телекоммуникации.

В ходе освоения раздела 6 «Микроконтроллеры» обучающиеся должны знать современные микроконтроллеры, их особенности, преимущества и недостатки.

В процессе выполнения лабораторных работ происходит изучение программирование процессоров и микроконтроллеров.

В процессе выполнения курсовой работы происходит изучение основ построения вычислительных систем, логические элементы, методы синтеза и анализа логических схем.

Работа с литературой является важнейшим элементом в получении знаний по дисциплине. Прежде всего, необходимо воспользоваться списком рекомендуемой по данной дисциплине литературой. Дополнительные сведения по изучаемым темам можно найти в периодической печати и Интернете.

К зачету допускаются студенты, которые выполнили и оформили все лабораторные работы и курсовую работу.

Оценка знаний, умений, навыков осуществляется в процессе промежуточной аттестации обучающихся по дисциплине, которая осуществляется в виде зачета. Для оценивания знаний, умений, навыков используются ФОС по дисциплине, содержащий вопросы к зачету.

**АННОТАЦИЯ**  
**рабочей программы дисциплины**  
**Вычислительные машины, системы и сети**

**1. Цель и задачи дисциплины**

Целью изучения дисциплины является: приобретение умений и навыков исследования проблем в своей предметной области, выбора методов и средств их решения, анализа результатов теоретических и экспериментальных исследований.

Задачей изучения дисциплины является: формирование способностей анализа результатов исследований, выбора методов и средств решения проблем в своей предметной области.

**2. Структура дисциплины**

2.1 Распределение трудоемкости по отдельным видам учебных занятий, включая самостоятельную работу: Лекции- 18ч., лабораторные занятия-36ч., самостоятельная работа-54ч. Курсовая работа.

Общая трудоемкость дисциплины составляет 108 часа, 3 зачетных единицы

2.2 Основные разделы дисциплины:

- 1 – Теоретические основы построения узлов ЭВМ
- 2 – Процессоры и микропроцессоры
- 3 – Организация ввода/вывода в ЭВМ
- 4 – Вычислительные системы
- 5 – Сети компьютеров
- 6 – Микроконтроллеры

**3. Планируемые результаты обучения (перечень компетенций)**

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- ОК-7 способность к самоорганизации и самообразованию;
- ОПК-7 способность учитывать современные тенденции развития электроники, измерительной и вычислительной техники, информационных технологий в своей профессиональной деятельности.

**4. Вид промежуточной аттестации: зачет**

*Протокол о дополнениях и изменениях в рабочей программе  
на 20\_\_-20\_\_ учебный год*

1. В рабочую программу по дисциплине вносятся следующие дополнения:

---

---

2. В рабочую программу по дисциплине вносятся следующие изменения:

---

---

---

Протокол заседания кафедры № \_\_\_\_\_ от «\_\_» \_\_\_\_\_ 20\_\_ г.,  
(разработчик)

Заведующий кафедрой \_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(Ф.И.О.)

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 27.03.04 Управление в технических системах от «20» октября 2015 г. №1171 и учебным планом ФГБОУ ВО «БрГУ» для очной, заочной формы обучения от «6» марта 2017 г. №125 и учебным планом ФГБОУ ВО «БрГУ» для ускоренной формы обучения от «4» апреля 2017 г. №203 и учебным планом ФГБОУ ВО «БрГУ» для очной, заочной формы обучения от «12» марта 2018 г. №130.

**Программу составил:**

Колтыгин Д.С., к.т.н, доцент \_\_\_\_\_

Рабочая программа рассмотрена и утверждена на заседании кафедры УТС

от «28» декабря 2018 г., протокол № 6

Заведующий кафедрой \_\_\_\_\_ И.В. Игнатъев

**СОГЛАСОВАНО:**

Заведующий выпускающей кафедрой \_\_\_\_\_ И.В. Игнатъев

Директор библиотеки \_\_\_\_\_ Т.Ф. Сотник

Рабочая программа одобрена методической комиссией факультета ЭиА

от «28» декабря 2018 г., протокол № 5

Председатель методической комиссии факультета \_\_\_\_\_

**СОГЛАСОВАНО:**

Начальник  
учебно-методического управления \_\_\_\_\_ Г.П. Нежевец

Регистрационный № \_\_\_\_\_