

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Страданченко Сергей Георгиевич

Должность: директор

Дата подписания: 18.11.2021 18:14:12

Уникальный идентификатор:  
fab83d7432c6481398711018a37134004b6775228bd796b69ac37a9044e06ade

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Институт сферы обслуживания и предпринимательства (филиал)  
федерального государственного бюджетного образовательного  
учреждения высшего образования «Донской государственный  
технический университет» в г. Шахты Ростовской области  
(ИСОиП (филиал) ДГТУ в г. Шахты)

КОЛЛЕДЖ ЭКОНОМИКИ И СЕРВИСА

На правах рукописи

# ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНЫХ СИСТЕМ

## УЧЕБНОЕ ПОСОБИЕ

для подготовки обучающихся специальности  
09.02.03 Программирование в компьютерных системах

Рассмотрены и рекомендованы для  
использования в учебном процессе на  
заседании педагогического совета  
Протокол № 1от «31» сентября 2018 г

Шахты  
ИСОиП (филиал) ДГТУ в г. Шахты  
2018

**Составитель:**  
Преподаватель высшей категории  
КЭС ФГБОУ ИСОиП (филиала) ДГТУ в г. Шахты  
*Л.В. Завгородняя*

**Рецензенты:**

Преподаватель высшей категории ГБПОУ РО «Дон-Текс»  
*Н.О. Бабаджанян*  
Преподаватель высшей категории КЭС ИСОиП (филиал) ДГТУ в г. Шахты  
*И.Ю. Бабенко*

**Техническое обеспечение компьютерных систем:** учебное пособие для подгот. обучающ. спец. 09.02.03 Программирование в компьютерных системах оч. и и заоч. форм обучения / сост. Л.В. Завгородняя, преп. высшей категории: Шахты: ИСОиП (филиал) ДГТУ в г. Шахты, 2019. – 98 с.

Настоящее учебное пособие рассматривает такие вопросы как: вычислительные приборы и устройства, алгоритмы и вычисления, архитектура и структура вычислительных машин и систем, классы архитектур вычислительных систем, персональные компьютеры.

Предназначено для обучающихся специальности 09.02.03 Программирование в компьютерных системах.

Режим доступа к электронной копии печатного издания:  
<http://www.libdb.sssu.ru>

© ИСОиП (филиал) ДГТУ, 2019

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
Раздел 1. Введение	7
1.1 Современное состояние рынка средств вычислительной техники	7
1.2 Услуги, предоставляемые поставщикам и потребителям информации	9
1.3 Микроэлектроника, вычислительная техника и информационная индустрия, их влияние на эффективность средств труда и технологических систем во всех сферах деятельности	12
Контрольные вопросы к Разделу 1	15
Раздел 2. Вычислительные приборы и устройства. Алгоритмы и вычисления	16
2.1 Информация, кодирование, обработка в ЭВМ. Системы счисления	16
2.2 Кодирование символьной информации. Кодирование и обработка чисел	17
2.3 Типы и структуры данных	20
2.4 Физическое представление обрабатываемой информации	22
2.5 Классы вычислительных машин. Логические основы ЭВМ, элементы и узлы	25
2.6 Алгоритмы и программы	27
Контрольные вопросы к Разделу 2	33
Раздел 3. Архитектура и структура вычислительных машин и систем	33
3.1 Базовые представления об архитектуре ЭВМ. Процессор, структура и функционирование	33
3.2 Технологии повышения производительности процессоров	39
3.3 Организация оперативной памяти	44
3.4 Дисковые массивы и уровни RAID. Внешние устройства	47
3.5 Создание конфигурации массивов RAID	52
Контрольные вопросы к Разделу 3	55
Раздел 4. Вычислительные системы	55
4.1 Основные определения. Классы архитектур вычислительных систем	55
4.2 Примеры некоторых архитектур вычислительных систем	60
4.3 Обобщенные представления об архитектуре вычислительных машин и систем	64
4.4 Перспективные типы процессоров ЭВМ	67
4.5 Коммуникационные среды	70
4.6 Кластерные и массивно-параллельные системы	75
Контрольные вопросы к Разделу 4	77
Раздел 5. Персональные компьютеры	78
5.1 Устройство ПК на процессорах Intel, AMD	78

5.2	Операционные системы ПК (DOS, Windows)	80
5.3	Установка ОС Windows. Настройка ОС Windows	85
5.4	Установка прикладных программ	88
5.5	Основы эксплуатации ЭВМ	90
	Контрольные вопросы к Разделу 5	94
	ЗАКЛЮЧЕНИЕ	95
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	96

## ВВЕДЕНИЕ

Учебное пособие по МДК 02.03 Техническое обеспечение компьютерных систем для обучающихся специальности 09.02.03 Программирование в компьютерных системах разработано в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по профессии 09.02.03 Программирование в компьютерных системах, утвержденного приказом Министерства образования и науки РФ от 28 июля 2014 г. N 804 «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах».

Целью изучения МДК 02.03 Техническое обеспечение компьютерных систем является ознакомление обучающихся с основными направлениями развития вычислительной техники; формирование информационной культуры в области теории и практики современного технического обеспечения ПК; приобретение навыков решения вопросов эффективности применения технических средств для решения практических задач.

В результате освоения дисциплины обучающиеся должны

### **Знать:**

- базовые понятия и основные принципы построения архитектур вычислительных систем (ОК 1-9, ПК 2.1-2.4);
- типы вычислительных систем и их архитектурные особенности (ОК 1-9, ПК 2.1-2.4);
- организацию и принцип работы основных логических блоков компьютерных систем (ОК 1-9, ПК 2.1-2.4);
- процессы обработки информации на всех уровнях компьютерных архитектур (ОК 1-9, ПК 2.1-2.4);
- основные компоненты программного обеспечения компьютерных систем (ОК 1-9, ПК 2.1-2.4);
- основные принципы управления ресурсами и организации доступа к этим ресурсам (ОК 1-9, ПК 2.1-2.4);
- характеристики ЭВМ, систем и сетей (ОК 1-9, ПК 2.1-2.4);

### **Уметь:**

- получать информацию о параметрах компьютерной системы (ОК 1-9, ПК 2.1-2.4);
- подключать дополнительное оборудование и настраивать связь между элементами компьютерной системы (ОК 1-9, ПК 2.1-2.4);
- производить установку и настройку программного обеспечения компьютерных систем (ОК 1-9, ПК 2.1-2.4);

- использовать методы и средства оценки характеристик вычислительных систем и сетей ЭВМ для решения задач числовой, символьной и распределительной обработки данных (ОК 1-9, ПК 2.1-2.4).

Учебное пособие предназначено для более полного усвоения теоретического материала следующих разделов рабочей программы:

Раздел 1. Введение;

Раздел 2. Вычислительные приборы и устройства. Алгоритмы и вычисления;

Раздел 3. Архитектура и структура вычислительных машин и систем;

Раздел 4. Вычислительные системы;

Раздел 5. Персональные компьютеры.

Учебное пособие способствует формированию следующих общих (ОК) и профессиональных (ПК) компетенций:

ОК-1: Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес

ОК-2: Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК-3: Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК-4: Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК-5 Использовать информационно-коммуникационные технологии в профессиональной деятельности

ОК-6 Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК-7 Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК-8 Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК-9: Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

ПК-2.1: Разрабатывать объекты базы данных.

ПК-2.2: Реализовывать базу данных в конкретной системе управления базами данных (СУБД).

ПК-2.3: Решать вопросы администрирования базы данных.

ПК-2.4: Реализовывать методы и технологии защиты информации в базах данных.

## Раздел 1. Введение

### 1.1 Современное состояние рынка средств вычислительной техники

Компьютеры являются фактически неотъемлемым элементом нынешней жизни. Рынок компьютерной техники в РФ в настоящее время считается более сформированным розничным направлением и конкурентоспособным. В последние годы высокоинтенсивный рост сегмента домашней электроники обусловлен потребительской активностью населения, связанной с улучшением общего благосостояния. С другой — непрерывное возникновение на рынке принципиально новых образцов техники, вследствие бурного формирования инновационного, научно-технического прогресса способствует понижению жизненного цикла потребительских товаров, повышая частоту их обновления.

Сорок процентов объема розничных продаж товаров в настоящее время приходится на информационно-коммуникационные технологии.

Общепринятыми фаворитами в изготовлении комплектующих для персональных компьютеров считаются государства азиатского региона, а кроме того Соединенные Штаты Америки, где высокотехнологичным отраслям экономики уделяется особый интерес. В Российской Федерации данное направление плохо сформировано, и фактически отсутствуют изготовители комплектующих для персональных компьютеров.

Основными поставщиками на российский рынок компьютерной техники являются: Acer; ASUS; HP; Samsung; Lenovo.

В соответствии с информацией с апреля по июнь 2018 года аналитической компании Gartner по итогам второго квартала мировой рынок персональных компьютеров (ПК) впервые вырос с начала 2012 года. Отсутствие роста объясняется тем, что многие пользователи начали отказываться от ПК и ноутбуков в пользу планшетов и смартфонов.

#### **В мире**

По оценке Gartner, во втором квартале было поставлено 62,1 млн персональных компьютеров – рост в 1,4% к аналогичному периоду прошлого года. Мировыми лидерами рынка ПК стали: Lenovo – 21,9% от всех поставленных за квартал ПК, HP Inc. – 21,9%, Dell – 16,8%, Apple – 7,1%, Acer – 6,4%. Стоит отметить, что компания IDC оценила мировой рынок ПК в 62,3 млн устройств, что на 2,7% больше значения прошлого года. Отличие в результатах связано с тем, что Gartner считает планшеты-трансформеры персональными компьютерами и учитывает их при оценке, но не включает в выборку «хромбуки» (Chromebooks) и планшеты с отличными от Windows операционными системами, а IDC, наоборот, – ис-

ключает из выборки планшеты-трансформеры, работающие на Windows, но учитывает «хромбуки».

### **Россия**

По данным исследования российского рынка персональных компьютеров IDC PC Quarterly Tracker на 31 мая 2018 года, за первый квартал в Россию было поставлено около 1,2 млн настольных и портативных ПК, что на 27,3% больше показателя аналогичного периода 2017 года.

При этом лидером российского рынка вопреки общемировым тенденциям стал HP. В пятерку лидеров вошли: HP Inc. – 27,6% от всех поставленных за квартал ПК, Lenovo – 17,6%, Acer – 12,6%, ASUS – 11,5% и Dell – 6,4%.

Рынок ноутбуков в первом квартале 2018 года вырос ещё больше – на 44,9% по сравнению с показателями аналогичного периода прошлого года. Объем поставок составил 863 тыс. штук. Пятёрка лидеров та же: HP Inc. Lenovo, ASUS, Acer и Dell (см. таблицу 1). Увы, но российских производителей ПК на отечественном рынке практически не видно.

Таблица 1 - Лидеры мирового рынка ПК, 2 квартал 2018

<b>Компания</b>	<b>2 кв 18 г. Поставки</b>	<b>2 кв 18 г. Доля рынка</b>	<b>2 кв 17 г. Поставка</b>	<b>2 кв 17 г. Доля рынка</b>	<b>2 кв 18 г./ 2 кв 17 г. Рост</b>
<b>1. HP Inc</b>	14,864	23,9%	13,812	22,8%	7,6%
<b>2. Lenovo</b>	13,755	22,1%	12,362	20,4%	11,3%
<b>3. Dell Inc</b>	11,255	18,1%	10,329	17,0%	9,0%
<b>4. Apple</b>	4,308	6,9%	4,303	7,1%	0,1%
<b>5. Acer Group</b>	4,197	6,7%	4,144	6,8%	1,3%
Другие	13,89	22,3%	14,991	24,7%	-7,3%
Всего	62,269	100,0%	60,624	100,0%	2,7%

Проведенный анализ рынка компьютерной техники в РФ свидетельствует о перспективном развитии в будущие периоды.

Таким образом, рынок компьютерной техники в России в дальнейшем будет развиваться по пути, проложенному западными странами, с той лишь разницей, что производство в данной области будет осуществляться не российскими фирмами, а западными и восточными партнерами, имеющими высокотехнологичные линии производства микросхем и других научно-, капиталоемких и трудоемких элементов современных персональных средств обработки данных. Укрепление позиций российских компьютерных фирм возможно в отношении последующей сборки компьютеров, а также интеграции компьютерной техники в общую систему взаимосвязей на предприятии. Возможно также наращивание позиций российских про-



изводителей программного обеспечения, которые уже сейчас завоевали авторитет среди пользователей.

## **1.2 Услуги, предоставляемые поставщикам и потребителям информации**

**Рынок информационных продуктов и услуг (информационный рынок)** - система экономических, правовых и организационных отношений при торговле продуктами интеллектуального труда на коммерческой основе.

Информационный рынок характеризуется определенной номенклатурой продуктов и услуг, условиями и механизмами их предоставления, ценами. В отличие от торговли обычными товарами, имеющими материально-вещественную форму, здесь в качестве предмета продажи или обмена выступают информационные системы, информационные технологии, лицензии, патенты, товарные знаки, ноу-хау, инженерно-технические услуги, различного рода информация и прочие виды информационных ресурсов.

Основными **источниками информации** для информационного обслуживания являются базы данных. Они интегрируют в себе поставщиков и потребителей информационных услуг, связи и отношения между ними, порядок и условия продажи и покупки информационных услуг.

**Поставщиками информационных продуктов и услуг** могут быть:

- центры, где создаются и хранятся базы данных, а также производится постоянное накопление и редактирование в них информации;
- центры, распределяющие информацию на основе разных баз данных;
- службы телекоммуникации и передачи данных;
- специальные службы, куда стекается информация по конкретной сфере деятельности для ее анализа, обобщения, прогнозирования, например банки, биржи;
- коммерческие фирмы;
- информационные брокеры.

Потребителями информационных продуктов и услуг могут быть различные юридические и физические лица.

*Пример. Об участниках и состоянии информационного рынка России можно узнать:*

- из справочников «*Вся компьютерная Москва*», «*Кто есть кто на компьютерном рынке России*», «*Информационные технологии Петербурга*»;

- из 1-го тома «Российская энциклопедия информации и телекоммуникаций»;

- из сети INTERNET.

**Потребители рынка информационных услуг** различаются задачами, решаемыми с использованием информационных услуг. По **степени коммерциализации задач** их можно условно разбить на следующие группы:

- ученые, исследователи, ИТР;
- представители делового мира;
- население.

Вторая группа потребителей отличается высокой покупательной способностью и повышенными требованиями к аналитической, готовой к употреблению информации. В настоящее время более 70% продаж информационного товара приходится на деловой мир, маркетинговые службы транснациональных корпораций, финансовые компании.

Потребителей, **с точки зрения использования баз данных**, можно разделить на две категории:

- а) потребители, непосредственно использующие информацию для решения маркетинговых задач фирмы;
- б) потребители, использующие покупные базы данных для организации коммерческого информационного обслуживания.

### **Инфраструктура информационного рынка**

Инфраструктура информационного рынка - совокупность секторов, каждый из которых объединяет группу людей или организаций, предлагающих однородные информационные продукты и услуги.

1. **Деловая информация** . Состоит из следующих частей:

- биржевая и финансовая информация - котировки ценных бумаг, валютные курсы, учетные ставки, рынок товаров и капиталов, инвестиции, цены. Поставщиками являются специальные службы биржевой и финансовой информации, брокерские компании, банки;

- статистическая информация - ряды динамики, прогнозные модели и оценки по экономической, социальной и демографическим областям. Поставщиками информации являются государственные службы, компании.

- коммерческая информация по компаниям, фирмам, корпорациям, направлениям работы и их продукции, ценам; о финансовом состоянии, связях, сделках, руководителях, деловых новостях в области экономики и бизнеса. Поставщиками являются специальные информационные службы.

2. **Информация для специалистов**. Содержит следующие части:

- профессиональная информация - специальные данные и информация для юристов, врачей, фармацевтов, преподавателей, инженеров и т.д.;

- *научно-техническая информация* - документальная, библиографическая, реферативная, справочная информация в области естественных, технических, общественных наук, по отраслям производства и сферам человеческой деятельности;

- *доступ к первоисточникам* - организация доступа к источникам информации через библиотеки и специальные службы, возможности приобретения первоисточников, их получения по межбиблиотечному абонементу в различных формах.

3. **Потребительская информация.** Состоит из следующих частей:

- *новости и литература* - информация служб новостей и агентств прессы, электронные журналы, справочники, энциклопедии;

- *потребительская информация* - расписания транспорта, резервирование билетов и мест в гостиницах, заказ товаров и услуг, банковские операции и т.п.;

- *развлекательная информация* - игры, телетекст, видеотекст.

4. **Услуги образования.** Включают все формы и ступени образования: дошкольное, школьное, специальное, средне профессиональное, высшее, повышение квалификации и переподготовку. Информационная продукция может быть представлена в компьютерном или некомпьютерном виде: учебники, методические разработки, практикумы, развивающие компьютерные игры, компьютерные обучающие и контролирующие системы и пр.

5. **Обеспечивающие информационные системы и средства.** Состоит из следующих частей:

- *программные продукты* - программные комплексы с разной ориентацией - от профессионала до неопытного пользователя компьютера: системное программное обучение, программы общей ориентации, прикладное программное обеспечение по реализации функций в конкретной области принадлежности, по решению задач типовыми математическими методами и др.;

- *технические средства* - компьютеры, телекоммуникационное оборудование, сопутствующие материалы и комплектующие;

- *разработка и сопровождение* информационных систем и *технологий* обследование организации в целях выявления информационных потоков, разработка концептуальных моделей, разработка структуры программного комплекса, создание и сопровождение баз данных;

- *консультирование по различным аспектам информационной индустрии* - какую приобретать информационную технику, какое программное обеспечение необходимо для реализации профессиональной деятельности, нужна ли информационная система и какая, на базе какой информационной технологии лучше организовать свою деятельность и т.д.;

- *подготовка источников информации* - создание баз данных по заданной теме, области, явлению и т.п.

В каждом секторе может быть организован любой вид доступа:

- непосредственный к хранилищу информации на бумажных носителях;

- дистанционный к удаленным или находящимся в данном помещении компьютерным базам данных.

### **1.3 Микроэлектроника, вычислительная техника и информационная индустрия, их влияние на эффективность средств труда и технологических систем во всех сферах деятельности**

#### **Информатика, вычислительная техника, микроэлектроника**

Сегодня, в век информатизации и компьютеризации, информация является таким же ресурсом, как трудовые, материальные и энергетические. **Информация** (informatio) - разъяснение, осведомленность, изложение. **Информация** - ценнейший ресурс наряду с такими традиционными видами ресурсов, как нефть, газ, полезные ископаемые и др., а значит, процесс ее переработки по аналогии с процессами переработки материальных ресурсов можно воспринимать как технологию.

Исторически первым носителем человеческих информации, знаний была речь, представлявшая изначально кодированные звуки для координации действий в человеческом сообществе. Затем появилось наскальное письмо каменного века, далее пиктограммы (“иконы”) бронзового века, иероглифическое письмо (сохраненное до сих пор, например, в Китае) и письмо “обычное” - конкатенация букв алфавита в слоги (“слоговое письмо”) и т.д.

Объединение систем, процессов, связанных с понятиями “информация”, “управление” привело к появлению **информатики**.

Подлинная информационная революция связана, прежде всего, с созданием электронно-вычислительных машин в конце 40-х годов, и с этого же времени исчисляется эра развития информационной технологии, материальное ядро которой образует микроэлектроника.

**Микроэлектроника** формирует элементную базу всех современных средств приема, передачи и обработки информации, систем управления и связи. Сама микроэлектроника возникла первоначально именно как технология: в едином кристаллическом устройстве оказалось возможным сформировать все основные элементы электронных схем.

#### **Современные информационные технологии**

**Информационные технологии (ИТ)** — это процессы, использующие совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта). Информационная технология является процессом, состоящим из четко регламентированных правил выполнения операций, действий, этапов разной степени сложности над данными, хранящимися в компьютерах.

**Область применения ИТ** - индустрия обработки информации, важнейший сектор общественного производства, оказывающий глубокое воздействие на темпы и характер развития современного общества.

В современном обществе основным техническим средством технологии переработки информации служит ПК, который существенно повлиял как на концепцию построения и использования технологических процессов, так и на качество резульатной информации.

Внедрение ПК в информационную сферу и применение телекоммуникационных средств связи определили современный этап развития ИТ. В понятие современной ИТ включены также коммуникационные технологии, которые обеспечивают передачу информации разными средствами, а именно — телефон, телеграф, телекоммуникации, факс и др.

### **Информационная индустрия**

**Информационная индустрия** — новейшая отрасль хозяйства отдельных государств и мира. Она, возникнув во втором периоде НТР, все еще остается формирующейся отраслью, несмотря на большой объем уже выполняемой ею информационной работы и очень высокую оценку ее роли в мировой экономике. До сих пор не сложилось единого, общепринятого названия этой молодой отрасли и одинакового подхода к составу ее производств. Всем этим объясняется отсутствие в официальных справочниках большинства государств данных по отрасли.

Информация — сведения, знания, обладание которыми обогащает представление об окружающем мире, а в практической деятельности людей влияет на процессы принятия решений и управление. Ее полезность состоит в предоставлении потребителю информации свободы действия. Она помогает из нескольких и даже многих вариантов решения задачи или проблемы выбрать оптимальный, оценить последствия других отвергнутых вариантов. Важнейшее свойство информации — она не является физическим объектом. Поэтому ее получение, хранение, передача, обработка осуществляются при очень малом потреблении энергии и вещества.

В отличие от материальных и финансовых ресурсов информация обладает **особыми свойствами**:

- количество информации как ресурса непрерывно возрастает, т.е. оно неисчерпаемо;

- информация не уменьшается и не исчезает даже при ее одновременном использовании многими потребителями;
- спрос на информацию в отличие от большинства других ресурсов может продолжаться непрерывно;
- производство информационного продукта может обходиться дороже, чем его копирование или тиражирование;
- информация, как и средства техники и технология, подвержена моральному износу, особенно при поступлении новых данных, но это происходит, как правило, гораздо быстрее (в течение дней и даже часов).

Информационная индустрия занимает особое место в сферах хозяйственной деятельности. Во-первых, она — одна из самых технически оснащенных отраслей экономики. Ее службы располагают сложнейшей дорогостоящей электронной аппаратурой. Затраты в расчете на одного занятого в отрасли выше, чем в большинстве других подразделений народного хозяйства. Функционирование многих служб осуществляется автоматически с помощью электронной техники и технологии. Моральный износ оборудования идет очень быстро, так как электронная промышленность непрерывно разрабатывает и поставляет все более совершенные его типы. Потребители продукции информационной индустрии также вынуждены обновлять свои технику (компьютеры и др.) и технологию (внедрять новые организационные виды программного обеспечения).

Во-вторых, информационная индустрия выполняет задачи предоставления услуг в виде самой разнообразной информации, обеспечения ею широкого и постоянно растущего контингента пользователей во всех сферах хозяйства и общественной жизни.

В-третьих, информационная индустрия — высокотехнологичная отрасль, использующая достижения не только промышленных техники и технологий, но и информатики как направления научной деятельности.

### **Влияние вычислительной техники и информационной индустрии на эффективность средств труда и технологических систем**

Использование информационных технологий для управления предприятием делает любую компанию более конкурентоспособной за счет повышения ее управляемости и адаптируемости к изменениям рыночной конъюнктуры. Подобная автоматизация позволяет:

- Повысить эффективность управления компанией за счет обеспечения руководителей и специалистов максимально полной, оперативной и достоверной информацией на основе единого банка данных.
- Улучшить делопроизводство при помощи оптимизации и стандартизации документооборота, автоматизации наиболее трудоемких его процедур.

- Снизить расходы на ведение дел за счет автоматизации процессов обработки информации, регламентации и упрощения доступа сотрудников компании к нужной информации. Изменить характер труда сотрудников, избавляя их от выполнения рутинной работы и давая возможность сосредоточиться на профессионально важных обязанностях.

- Обеспечить надежный учет и контроль поступлений и расходования денежных средств на всех уровнях управления.

- Руководителям среднего и нижнего звеньев анализировать деятельность своих подразделений и оперативно готовить сводные и аналитические отчеты для руководства и смежных отделов.

- Повысить эффективность обмена данными между отдельными подразделениями, филиалами и центральным аппаратом.

- Гарантировать полную безопасность и целостность данных на всех этапах обработки информации. И многое другое.

### **Контрольные вопросы к Разделу 1**

1. Дайте определение информации, информатизации и информационным технологиям (ИТ).

2. На чем основываются информационные технологии в экономике и управлении?

3. Раскройте содержание понятия «информационные технологии в сфере экономики и управления».

4. Какие фирмы являются основными поставщиками компьютерной техники на российский рынок?

5. Что такое информационный рынок?

6. Кто является Поставщиком информационных продуктов и услуг?

7. Как можно классифицировать потребителей рынка информационных услуг?

8. Что такое инфраструктура информационного рынка?

9. Из каких секторов состоит инфраструктура информационного рынка?

10. Что такое микроэлектроника?

11. Какова область применения ИТ?

12. Что такое информационная индустрия?

13. Каковы свойства информации как ресурса?

14. Какое влияние оказывает вычислительная техника и информационная индустрия на эффективность средств труда и технологических систем?

## Раздел 2. Вычислительные приборы и устройства. Алгоритмы и вычисления

### 2.1 Информация, кодирование, обработка в ЭВМ. Системы счисления

Информация - это сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специализированным устройством, например ЭВМ, для обеспечения целенаправленной деятельности.

Информация бывает:

- Текстовая.
- Числовая.
- Графическая.
- Звуковая.

Информацию можно хранить, обрабатывать, передавать.

Любая информация, с которой работает современная вычислительная техника, преобразуется в числа в двоичной системе счисления.

Единицы измерения информации.

1 байт = 8 бит

1 Кбайт = 1024 байтам

1 Мбайт = 1024 Кбайтам

1 Гбайт = 1024 Мбайтам

1 Тбайт = 1024 Гбайтам

**Кодирование информации** — процесс преобразования сигнала из формы, удобной для непосредственного использования информации, в форму, удобную для передачи, хранения или автоматической переработки.

Для кодирования текстов используются различные таблицы перекодировки. Важно, чтобы при кодировании и декодировании одного и того же текста использовалась одна и та же таблица.

Таблица перекодировки - таблица, содержащая упорядоченный некоторым образом перечень кодируемых символов, в соответствии с которой происходит преобразование символа в его двоичный код и обратно.

Наиболее популярные таблицы перекодировки: ДКОИ-8, ASCII, CP1251, Unicode.

Кодирование числовой информации – это запись числа в двоичном коде так что число 5 в десятичной системе счисления будет 0 000 0101. Отрицательная (-5) будет 1 111 10101.

#### **Кодирование графической информации**

Изображение является непрерывным сигналом. Чтобы перевести его в двоичный код необходимо разбить изображение на отдельные точки.



Каждая точка имеет свой цвет, который можно закодировать двоичным кодом. Записав код каждой точки, мы получим код всего изображения.

### **Кодирование звуковой информации**

Звук представляет собой гармонические колебания в определенном диапазоне частот, распространяющихся в определенной среде. Сигналы, несущие звук, являются аналоговыми (непрерывными). Поэтому для представления звука двоичным кодом нужно преобразовать аналоговый сигнал в дискретный (двоичный). Выполняют это преобразование специальные устройства.

**Обработка информации** - получение одних информационных объектов из других информационных объектов путем выполнения некоторых алгоритмов.

Компьютер может вводить информацию, обрабатывать, выводить, а также накапливать.

### **Системы счисления**

**Система счисления** – это совокупность приёмов и правил для записи чисел цифрами и знаками.

Система счисления делится на два вида *позиционная* и *не позиционная*.

**Позиционная система счисления** – это система счисления, в которой значение цифры напрямую зависит от её положения в числе. Например, цифры: 1, 2, 3 будут давать числа: 123, 132, 213, 231, 312, 321.

**Не позиционная система счисления** – это система счисления, в которой величина, которую обозначает цифра, не зависит от положения в числе. При этом система может накладывать ограничения на положение цифр, например, чтобы они были расположены в порядке убывания. Например, цифры: I-1, V-4, X-10 будут давать число: XVI-15, XXXVVI-39 и т.п.

Числа в разных системах можно переводить в другие системы, складывать, вычитать, умножать, делить.

## **2.2 Кодирование символьной информации. Кодирование и обработка чисел**

### **Кодирование информации в компьютере**

Вся информация, которую обрабатывает компьютер, должна быть представлена двоичным кодом с помощью двух цифр — 0 и 1. Эти два символа принято называть двоичными цифрами, или битами. С помощью двух цифр 1 и 0 можно закодировать любое сообщение. Это явилось причиной того, что в компьютере обязательно должно быть организовано два важных процесса:

- **кодирование**, которое обеспечивается устройствами ввода при преобразовании входной информации в форму, воспринимаемую компьютером, то есть в двоичный код;
- **декодирование**, которое обеспечивается устройствами вывода при преобразовании данных из двоичного кода в форму, понятную человеку.

С точки зрения технической реализации использование двоичной системы счисления для кодирования информации оказалось намного более простым, чем применение других способов. Действительно, удобно кодировать информацию в виде последовательности нулей и единиц, если представить эти значения как два возможных устойчивых состояния электронного элемента:

- 0 — отсутствие электрического сигнала или сигнал имеет низкий уровень;
- 1 — наличие сигнала или сигнал имеет высокий уровень.

Эти состояния легко различать. Недостаток двоичного кодирования — длинные коды. Но в технике легче иметь дело с большим числом простых элементов, чем с небольшим количеством сложных.

Вам и в быту ежедневно приходится сталкиваться с устройством, которое может находиться только в двух устойчивых состояниях: включено/выключено. Конечно же, это хорошо знакомый всем выключатель. А вот придумать выключатель, который мог бы устойчиво и быстро переключаться в любое из 10 состояний, оказалось невозможным. В результате после ряда неудачных попыток разработчики пришли к выводу о невозможности построения компьютера на основе десятичной системы счисления. И в основу представления чисел в компьютере была положена именно двоичная система счисления.

В настоящее время существуют разные способы двоичного кодирования и декодирования информации в компьютере. В первую очередь это зависит от вида информации, а именно, что должно кодироваться: текст, числа, графические изображения или звук. Кроме того, при кодировании чисел важную роль играет то, как они будут использоваться: в тексте, в расчетах или в процессе ввода-вывода. Накладываются также и особенности технической реализации.

### **Кодирование чисел**

Система счисления — совокупность приемов и правил записи чисел с помощью определенного набора символов.

Для записи чисел могут использоваться не только цифры, но и буквы (например, запись римских цифр — XXI). Одно и то же число может быть по-разному представлено в различных системах счисления.

В зависимости от способа изображения чисел системы счисления делятся на позиционные и непозиционные.

В позиционной системе счисления количественное значение каждой цифры числа зависит от того, в каком месте (позиции или разряде) записана та или иная цифра этого числа. Например, меняя позицию цифры 2 в десятичной системе счисления, можно записать разные по величине десятичные числа, например 2; 20; 2000; 0,02 и т. д.

В непозиционной системе счисления цифры не изменяют своего количественного значения при изменении их расположения (позиции) в числе. Примером непозиционной системы может служить римская система, в которой независимо от местоположения одинаковый символ имеет неизменное значение (например, символ X в числе XXV).

Количество различных символов, используемых для изображения числа в позиционной системе счисления, называется основанием системы счисления.

В компьютере наиболее подходящей и надежной оказалась двоичная система счисления, в которой для представления чисел используются последовательности цифр 0 и 1.

Кроме того, для работы с памятью компьютера оказалось удобным использовать представление информации с помощью еще двух систем счисления:

- *восьмеричной* (любое число представляется с помощью восьми цифр — 0, 1, 2... 7);
- *шестнадцатеричной* (используемые символы-цифры — 0, 1, 2... 9 и буквы — A, B, C, D, E, F, заменяющие числа 10, 11, 12, 13, 14, 15 соответственно).

### **Кодирование символьной информации**

Нажатие алфавитно-цифровой клавиши на клавиатуре приводит к тому, что в компьютер посылается сигнал в виде двоичного числа, представляющего собой одно из значений кодовой таблицы. Кодовая таблица - это внутреннее представление символов в компьютере. Во всем мире в качестве стандарта принята таблица ASCII (American Standart Code for Informational Interchange - американский стандартный код информационного обмена).

Для хранения двоичного кода одного символа выделен 1 байт = 8 бит. Учитывая, что каждый бит принимает значение 1 или 0, количество возможных сочетаний единиц и нулей равно  $2^8 = 256$ .

Значит, с помощью 1 байта можно получить 256 разных двоичных кодовых комбинаций и отобразить с их помощью 256 различных символов. Эти коды и составляют таблицу ASCII.

Например, при нажатии клавиши с буквой S в память компьютера записывается код 01010011. При выводе буквы S на экран компьютер выполняет декодирование — на основании этого двоичного кода строится изображение символа.

SUN (СОЛНЦЕ) - 01010011 01010101 01001110

Стандарт ASCII кодирует первые 128 символов от 0 до 127: цифры, буквы латинского алфавита, управляющие символы. Первые 32 символа являются управляющими и предназначены в основном для передачи команд управления. Их назначение может варьироваться в зависимости от программных и аппаратных средств. Вторая половина кодовой таблицы (от 128 до 255) американским стандартом не определена и предназначена для символов национальных алфавитов, псевдографических и некоторых математических символов. В разных странах могут использоваться различные варианты второй половины кодовой таблицы.

**Обратите внимание!** Цифры кодируются по стандарту ASCII в двух случаях - при вводе-выводе и когда они встречаются в тексте. Если цифры участвуют в вычислениях, то осуществляется их преобразование в другой двоичный код.

Для сравнения рассмотрим число 45 для двух вариантов кодирования.

При использовании в тексте это число потребует для своего представления 2 байта, поскольку каждая цифра будет представлена своим кодом в соответствии с таблицей ASCII. В двоичной системе - 00110100 00110101.

При использовании в вычислениях код этого числа будет получен по специальным правилам перевода и представлен в виде 8-разрядного двоичного числа 00101101, на что потребуется 1 байт.

## 2.3 Типы и структуры данных

### Типы данных

Под типом данных (*data type*) понимается множество величин, объединенных определенными признаками и совокупностью допустимых преобразований.

Так, если в качестве объединяющего признака используется **вид данных**, то данные можно разделить на следующие типы:

- символьные;
- текстовые;
- звуковые;
- графические.

Например, в *MS Excel* выделяются три основных типа данных: число, текст и формула.

Если в качестве объединяющего признака использовать **способ представления данных** в вычислительных системах (алгоритмических языках программирования), то данные можно разделить на следующие типы:

- целочисленный – используется для представления целых чисел;
- вещественный – используется для представления чисел с плавающей запятой (точкой);
- строковый – используется для представления строки символов;
- логический – используется для представления логических значений «ложь» или «истина»;
- указательный – используется для ссылки на другой объект и т. д.

Например, в объектно-ориентированном языке программирования Visual Basic вводятся следующие типы данных: целочисленный тип обозначается Byte (целые числа от 0 до 255); Long (целые числа двойной длины); Integer (целые числа от -32768 до 32767); Single (вещественные числа одинарной точности с плавающей точкой); Double (вещественные числа двойной точности с плавающей точкой); String (текстовая строка); Boolean (логические значения, true – «истина» и false – «ложь»); Object (ссылка на другой объект).

Если в качестве объединяющего признака выбрать **пригодность данных к решению задач управления**, то данные можно разделить на данные, которые:

- используются при принятии решений;
- обычно непосредственно не используются при принятии решений, но накапливаются для возможного использования в определенных ситуациях;
- не используются при принятии решений и их использование не предусматривается, такие данные избыточные.

Таким образом, в зависимости от принятого классификационного признака данные можно разделять на типы.

### **Структуры данных**

Работа с большими наборами данных автоматизируется проще, когда данные упорядочены, т.е. образуют заданную структуру.

Под структурой данных (*data structure*) понимается множество элементов данных, которые определенным образом объединены и упорядочены.

Для объединения данных применяют линейные, табличные, иерархические и сетевые структуры.

**Линейная структура** данных, называемая также списком, – это упорядоченная структура, в которой адрес элемента данных однозначно определяется его индексом (номером). Примером линейной структуры может быть список сотрудников коммерческой фирмы и т. д. В списках обычно новый элемент начинается с новой строки. Если элементы располагаются в строку, вводят разделительные знаки между элементами.

**Табличная структура** данных – это упорядоченная структура, в которой адрес элемента данных однозначно определяется двумя числами – номером строки и номером столбца, на пересечении которых находится ячейка с исходным элементом. Характерным примером такой организации данных являются данные, записанные в соответствующие ячейки программы *MS Excel*.

**Иерархическая структура** данных – это упорядоченная структура, в которой адрес каждого элемента определяется путем (маршрутом доступа), идущим от вершины структуры к данному элементу. В иерархической структуре элементы распределены по уровням. Каждый элемент более высокого уровня может состоять из элементов нижнего уровня, а элемент нижнего уровня может входить в состав только одного элемента более высокого уровня. Примером такой структуры является левая часть окна утилиты «Проводник» операционной системы *Windows* (рис. 6.1) или почтовые адреса.

**Сетевая структура** – структура, в которой элементы связаны между собой произвольным образом, например сетевая база данных.

Кроме приведенных выше существуют структуры данных, определяемые той или иной **предметной областью**. Например, в системах управления базами данных (СУБД) данные могут быть объединены и упорядочены в следующие структуры:

- реляционная;
- иерархическая;
- сетевая.

В математических и других дисциплинах могут использоваться такие структуры данных, как **массив** (структурированный тип данных, состоящий из некоторого числа элементов одного типа), **запись** (совокупность элементов данных разного типа), **множество** (тип данных, состоящий из однотипных неповторяющихся элементов) и т. д.

## 2.4 Физическое представление обрабатываемой информации

### Физическое представление информации

В вычислительных системах информация представляется в двоичном алфавите. Физическими аналогами знаков этого алфавита служат физиче-

ские сигналы, способные принимать два хорошо различимых значения, например электрическое напряжение (потенциал) высокого и низкого уровня, отсутствие и наличие импульса тока, противоположные по знаку значения напряженности магнитного поля и т. п.

Непременным требованием к физическим аналогам двоичного алфавита является возможность надежного распознавания двух различных значений сигнала, которые при описании законов функционирования схем обозначаются символами 0 (нуль) и 1 (единица).

В схемах цифровых устройств переменные и соответствующие им сигналы изменяются и воспринимаются не непрерывно, а лишь в дискретные моменты времени — по тактовым импульсам.

В цифровых устройствах применяют три способа физического представления информации: *потенциальный, импульсный и динамический*. Слово может быть представлено последовательным или параллельным способом (кодом). Устройства последовательного действия работают медленнее, чем параллельного. Однако устройство параллельного действия требует большего объема аппаратуры. В вычислительной технике применяются оба способа в зависимости от требований, предъявляемых к конкретному изделию.

### Классификация ЭВМ по физическому представлению данных

По физическому представлению данных ЭВМ делятся на **аналоговые** (АВМ - непрерывного действия), **цифровые** (ЦВМ - дискретного действия) и **гибридные** (ГВМ - на разных этапах обработки используются разные способы физического представления данных). Классификация ЭВМ по физическому представлению данных приведена на рисунке 1.

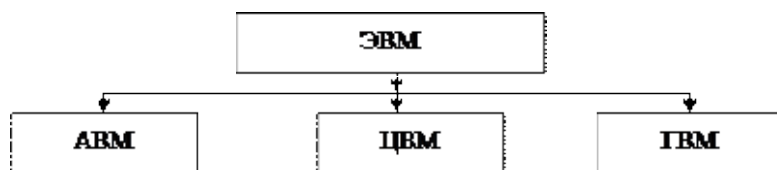


Рисунок 1 - Классификация ЭВМ по физическому представлению данных

**АВМ** – работают с информацией, представленной в аналоговой форме, т.е. в виде непрерывного ряда значений некоторой физической величины, например, электрического напряжения. Они просты и удобны в использовании, возможны высокие скорости вычислений, но низкая точность вычислений. Обычно используются для решения математических задач, например, дифференциальных уравнений. Пример аналоговой формы представления информации в ЭВМ показан на рисунке 2.

**ЦВМ** - работают с информацией, представленной в дискретной, а точнее цифровой форме. Это наиболее универсальное средство обработки

данных. Пример цифровой формы представления информации в ЭВМ показан на рисунке 3.

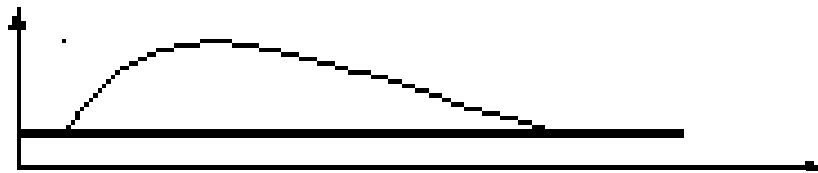


Рисунок 2 - Аналоговая форма представления информации в ЭВМ

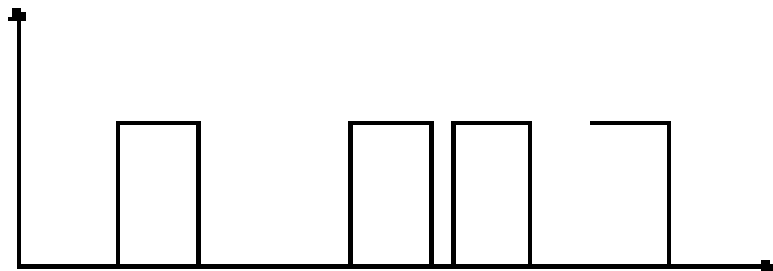


Рисунок 3 - Цифровая форма представления информации в ЭВМ

**ГВМ** – работают с информацией, представленной и в цифровой, и в аналоговой форме, и совмещают преимущества АВМ и ЦВМ. ГВМ целесообразно использовать для решения задач управления сложными быстродействующими техническими комплексами.

В экономике, науке и технике преимущественно используются ЦВМ.

### **Процессы обработки информации**

Информация в вычислительной системе подвергается различным процессам: вводу, хранению, обработке и выводу.

Ввод информации в компьютерную систему осуществляется с устройств ввода, имеющих большое разнообразие, начиная с клавиатуры, манипулятора «мышь» и заканчивая современными цифровыми фото и видеокамерами, а также с различных дисков, флэш-карт и т. д.

Хранение информации производится на запоминающих устройствах: кратковременное — в оперативной памяти и в различных регистрах памяти, выполненных на полупроводниковых приборах, магнитных и оптических элементах; долговременное — во внешних запоминающих устройствах, выполненных на магнитных лентах (стримерах), дисках (жестких типа «винчестер», мягких — (практически уже не применяемых с появлением стандарта USB)), и триумфально шествующих разнообразных лазерных и флэш-технологий.

Обработка информации в вычислительной системе производится в соответствии с принятой системой команд, алгоритмами, определяемыми программным обеспечением, и командами, поступающими с внешних устройств управления.



Вывод информации производится на внешние устройства связи и регистрации информации без ее визуального отображения (на указанные выше запоминающие устройства) и устройства с отображением: печатающие устройства, индикаторы, табло и другие устройства индивидуального и коллективного отображения. Выбор метода обработки информации определяется характером решаемых задач, особенностями используемой информации, а также параметрами технических средств автоматизации и возможностями программного и аппаратного обеспечения компьютерных систем.

## 2.5 Классы вычислительных машин. Логические основы ЭВМ, элементы и узлы

### Назначение ЭВМ

Существуют различные способы классификации ЭВМ. Классификацию ЭВМ по физическому представлению данных мы рассмотрели в предыдущей лекции. Теперь рассмотрим классификацию по назначению, показанную на рисунке 4. По этому параметру ЭВМ можно разделить на три группы.

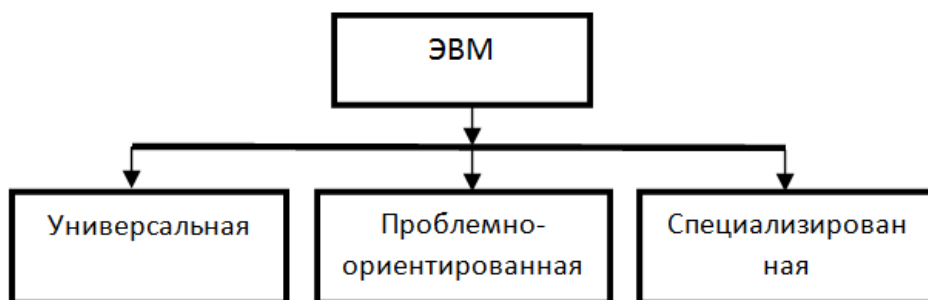


Рисунок 4 - Классификация ЭВМ по назначению

*Универсальные* ЭВМ (общего назначения) – предназначены для решение различных видов математических, экономических, инженерно-технических и др. задач, характеризующихся сложными алгоритмами и большим объёмом обрабатываемых данных. Они обладают высокой производительностью, разнообразием форм обрабатываемых данных, большим набором операций, большой ёмкостью памяти, развитой системой ввода/вывода.

*Проблемно-ориентированные* ЭВМ – предназначены для решения узкого круга задач, например, управление технологическими объектами, обработка небольших объёмов данных. Их ресурсы ограничены.

*Специализированные ЭВМ* - предназначены для решения определённого узкого круга задач или реализации строго определённой группы функций. Узкая ориентация позволяет специализировать их структуру и получить высокую производительность при небольшой сложности. Примеры: программируемые МП спецназначения, адаптеры и контроллеры для логического управления отдельными устройствами и процессами, устройства согласования работы отдельных узлов ВС.

### **Поколения ЭВМ**

Такой вид классификации связан с тем, что за время своего развития ЭВМ проделали большую эволюцию, как с точки зрения элементной базы, так и в плане изменения их структуры, появления новых возможностей, расширения области применения и характера использования. Но это деление достаточно условно. Поколения ЭВМ и их характеристики представлены в таблице 2.

Каждое следующее поколение имеет значительно лучшие характеристики по производительности и емкости памяти.

Таблица 2 – Классификация ЭВМ по поколениям

<b>Поколение</b>	<b>Период</b>	<b>Элементная база</b>
<b>1</b>	50-е годы	Электронные вакуумные лампы
<b>2</b>	60-е годы	Транзисторы (дискретные полупроводниковые приборы)
<b>3</b>	70-е годы	Полупроводниковые интегральные схемы <sup>1</sup> с малой и средней степенью интеграции (100-1000 транзисторов в 1 корпусе)
<b>4</b>	80-90-е годы	Большие и сверхбольшие ИС <sup>2</sup> (основная из них МП – дес.тыс.-млн. активных элементов на 1 кристалле)
<b>5</b>	Настоящее время	Многие десятки параллельно работающих МП для построения эффективных систем обработки знаний; сверхсложные МП с параллельно-векторной структурой для одновременного выполнения десятков последовательных инструкций программы
<b>6</b>	Следующие поколения	Оптоэлектронные компьютеры с массовым параллелизмом и нейронной структурой, с распределённой сетью множества несложных МП, моделирующих архитектуру нейронных биологических систем.

### **Размеры и вычислительная мощность**

Классификация ЭВМ по размерам и вычислительной мощности представлена на рисунке 5.

Наиболее важные характеристики этих классов производительности приведены в таблице 3.

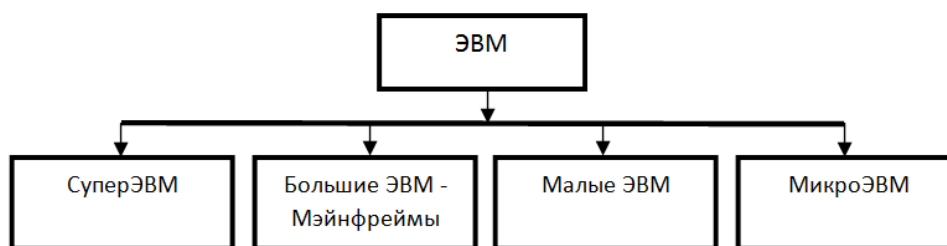


Рисунок 5 - Классификация ЭВМ по размерам и вычислительной мощности

Таблица 3 – Классы ЭВМ и их характеристики

Параметр	Класс ЭВМ			
	СуперЭВМ	Большие ЭВМ	Малые ЭВМ	МикроЭВМ
Производительность млн/сек (MIPS)	1000 - 1000000	100 - 10000	10 - 1000	10 - 100
Емкость ОП Мбайт	2000 - 100000	512 - 10000	128 - 2048	32 - 512
Ёмкость ВЗУ Гбайт	500 - 50000	100 - 10000	20 - 500	10 - 50
Разрядность Бит	64 - 256	64 - 128	32 - 128	32 - 128
Пример	ILLIAC IV, Cray	IBM/370, Comparex	PDP-11, VAX	IBM PC, Apple Macintosh,

## 2.6 Алгоритмы и программы

### Понятие алгоритма

Написанию программы всегда предшествует разработка некоторого плана (*алгоритма*) решения задачи.

Понятие алгоритма является одним из основных понятий современной математики. Еще на самых ранних ступенях развития математики (Древний Египет, Вавилон, Греция) в ней стали возникать различные вычислительные процессы чисто механического характера. С их помощью искомые величины ряда задач вычислялись последовательно из исходных величин по определенным правилам и инструкциям. Со временем все такие процессы в математике получили название алгоритмов. Термин *алгоритм* происходит от имени средневекового узбекского математика Аль-Хорезми, который еще в IX в. (825 г.) дал правила выполнения четырех арифметических действий в десятичной системе счисления. Процесс выполнения арифметических действий был назван *алгоритмом*. С 1747 г. вместо слова *алгоритм* стали употреблять *алгорисмус*, смысл которого со-

стоял в комбинировании четырех операций арифметического исчисления — сложения, вычитания, умножения, деления. К 1950 г. *алгорисмус* стал *алгоритмом*. Смысл алгоритма чаще всего связывался с алгоритмами Евклида — процессами нахождения наибольшего общего делителя двух натуральных чисел, наибольшей общей меры двух отрезков и т.п.

*Под алгоритмом понимали конечную последовательность точно сформулированных правил, которые позволяют решать те или иные классы задач. Такое определение алгоритма не является строго математическим, так как в нем не содержится точной характеристики того, что следует понимать под классом задач и под правилами их решения.*

Запишем следующее определение понятия **алгоритм**.

**Алгоритм** - это однозначно определенная последовательность действий, записанная на понятном исполнителю *алгоритмическом языке* и определяющая процесс перехода от исходных данных к результату.

### **Свойства алгоритма**

В приведенном определении уже указаны основные **свойства алгоритма**.

- *Дискретность* - алгоритм состоит из конечного набора инструкций или *шагов*.

- *Однозначность* - каждый шаг понимается исполнителем единственным образом, что позволяет гарантированно получить решение для некоторого набора входных данных.

- *Массовость* - алгоритм работает при меняющихся в некоторых пределах входных данных. Он всегда сводится к некоторому преобразованию исходных данных в результат или результаты. В этом смысле формулы для решения квадратного уравнения или даже четко составленную инструкцию по варке кофе можно считать алгоритмами, выполнимыми исполнителем-человеком. Для машины, разумеется, требуется более четкая формализация задачи, чем для человека, понимать естественный язык компьютеры пока неспособны, отсюда необходимость учета при составлении алгоритма ограниченного набора инструкций ЭВМ.

- *Результативность* - за конечное число шагов достигается некоторый результат.

### **Формы записи алгоритма**

Алгоритмы можно записывать по-разному. Форма записи, состав и количество операций алгоритма зависят от того, кто будет исполнителем этого алгоритма. Если задача решается с помощью ЭВМ, алгоритм решения задачи должен быть записан в понятной для машины форме, т. е. в виде программы. Всякий алгоритм может быть:

- записан на естественном языке (словесное представление);
- изображен в графической форме (в виде блок-схемы);
- записан в текстовой форме (псевдокод);
- записан на алгоритмическом языке.

Первоначально для записи алгоритмов пользовались средствами обычного языка (*словесное представление алгоритмов*). Уточним понятие словесного представления алгоритма на примере нахождения произведения  $n$  натуральных чисел — факториал числа  $n$  ( $c = n!$ ), т.е. вычисления по формуле  $c = 1*2*3*4*...*n$ . Этот процесс может быть записан в виде следующей системы последовательных указаний (пунктов):

1. Полагаем  $c$  равным единице и переходим к следующему пункту.
2. Полагаем  $i$  равным единице и переходим к следующему пункту.
3. Полагаем  $c = i * c$  и переходим к следующему пункту.
4. Проверяем, равно ли  $i$  числу  $n$ . Если  $i = n$ , то вычисления прекращаем. Если  $i < n$ , то увеличиваем на единицу  $i$  и переходим к пункту 3.

В данном алгоритме в качестве элементарных операций используются простейшие арифметические операции умножения, которые могли бы быть разложены на еще более элементарные операции. Мы такого разбиения не делаем в силу простоты и привычности арифметических правил.

Алгоритмы, в соответствии с которыми решение поставленных задач сводится к арифметическим действиям, называются **численными алгоритмами**.

Алгоритмы, в соответствии с которыми решение поставленных задач сводится к логическим действиям, называются **логическими алгоритмами**.

Примерами логических алгоритмов могут служить алгоритмы поиска минимального числа, поиска пути на графе, поиска пути в лабиринте и др.

**Графическая** форма записи (*блок-схема*) характерна тем, что отдельные шаги алгоритма изображаются геометрическими фигурами, а последовательность выполнения шагов - связями между этими фигурами. На рисунке 6 указаны основные элементы блок-схем.

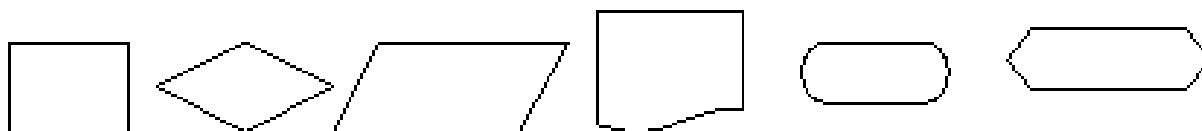


Рисунок 6 – Основные элементы блок-схем

Указанные на рисунке геометрические фигуры интерпретируются так:

*Прямоугольник* - любая последовательность действий; внутри прямоугольника записываются формулы или словесное описание выполняемых действий;

*Ромб* - блок проверки условия; так как любое условие может быть только истинно или ложно, у блока 1 вход и 2 выхода, соответствующие действиям, выполняемым в случаях, когда условие истинно и когда оно ложно. Выходы подписываются символами "+" и "-", "да" и "нет", "1" и "0" и т. п.

*Параллелограмм* - блок ввода исходных данных. Внутри фигуры обычно пишется, какие именно данные должны быть введены;

*Лист с разрывом* - блок вывода данных. Внутри блока указывается, какие данные или сообщения программа выводит для представления пользователю;

*Закругленные прямоугольники* - необязательные блоки начала и конца программы, внутри блоков обычно указываются ключевые слова "нач" и "кон" соответственно;

*Шестигранник* служит для изображения циклов, как правило, у нее 2 входа (первый и повторный вход в цикл) и 1 выход, соответствующий завершению циклического процесса.

На рисунке 7 приведен пример блок-схемы, иллюстрирующей известный процесс решения квадратного уравнения.

Язык блок-схем довольно громоздок, как правило, он не применяется профессионалами, однако, на начальном этапе обучения программированию планирование несложных программ в виде блок-схем может оказаться весьма полезным.

*Текстовая* форма записи алгоритма (*псевдокод*) характерна тем, что шаги алгоритма и последовательность их выполнения задаются с помощью набора специальных *ключевых слов*. Эта форма ближе к реальным языкам программирования. Существует много различных вариантов псевдокода, например, в русскоязычной литературе по программированию распространен следующий вариант псевдокода:

- нач** - начало программы;
- кон** - конец программы;
- если ... то ... иначе** - проверка условия;
- ввод** - ввод данных;
- вывод** - вывод данных;
- для ... от .. до ... нц ... кц** - цикл со счетчиком (нц - начало цикла, кц - конец);
- пока ... нц ...кц** - цикл с предусловием;
- нц ... кц ... пока** - цикл с постусловием.

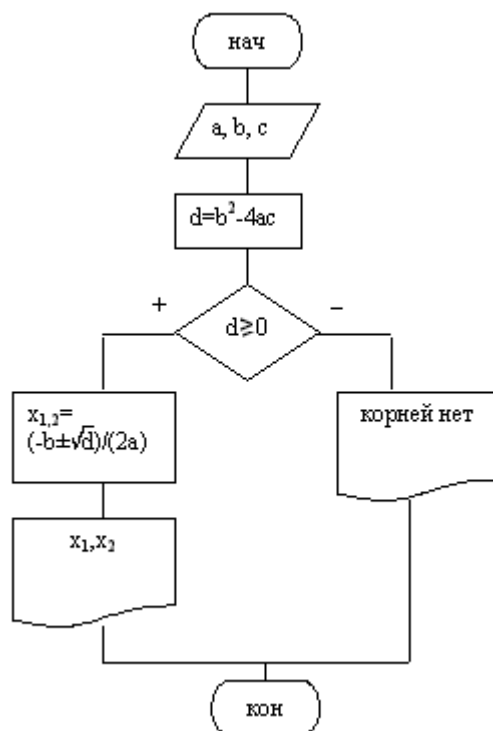


Рисунок 7 – Блок-схема решения квадратного уравнения

### Программа и программное обеспечение

Программа - это *реализация* алгоритма на конкретном языке программирования. Совокупность существующих программ образует *программное обеспечение* (ПО). ПО принято делить на 2 класса.

*Системное ПО* обеспечивает работу компьютера и внешних устройств, а также поддержку прикладных программ. Оно разрабатывается квалифицированными специалистами на машинно-ориентированных языках программирования, дающих доступ к аппаратуре компьютера. Примерами системного ПО могут служить операционная система Windows (или любая другая), *драйверы* внешних устройств компьютера, *утилиты* для его технического обслуживания, *системы программирования*, применяемые для разработки собственных приложений. Иногда *системы программирования* выделяют в отдельный класс: *инструментарий технологии программирования*.

*Прикладное ПО* предназначено для решения конкретных задач пользователя. Оно разрабатывается на языках высокого уровня, облегчающих процесс программирования за счет множества готовых решений. К одному из таких языков относится и Паскаль, который мы будем изучать.

### Этапы разработки программы

Разработка любой программы, от несложной учебной задачи до профессионального приложения, может быть разбита на ряд этапов. Кратко опишем и охарактеризуем их.

1. Определение входных и выходных данных, требований к программе.

На первом этапе определяются входные и выходные данные программы, способ ее взаимодействия (*интерфейса*) с пользователем, язык и среда программирования, в которой она будет разрабатываться, а также требования к аппаратному и системному программному обеспечению компьютеров, на которых будет работать приложение.

2. Разработка алгоритма.

На этом шаге производится определение последовательности действий, ведущих к решению задачи и запись их в одной из существующих форм.

3. Кодирование (программирование).

Третий этап - это перевод алгоритма на язык программирования и создание *исходного текста программы* в одной из систем программирования. Программа на любом языке состоит из *операторов* - так называются отдельные действия, разрешенные в языке. Число операторов в любом языке ограничено и правила их написания жестко заданы.

4. Компиляция и отладка.

Исходный текст на языке алгоритмическом программирования не будет непосредственно исполняться компьютером - для работы программы ее требуется *откомпилировать*, т. е., перевести в машинный код. Эту работу выполняет специальная программа-*компилятор* или *оболочка языка*. В результате преобразования компилятором исходного текста программы в машинный код получается *исполняемый файл* с расширением *exe*, который можно запустить (*выполнить*) в той операционной системе (ОС), для которой разработан компилятор.

Программа, которую удалось откомпилировать, не обязательно работает правильно. Она может содержать ошибки, для выявления которых предназначен этап *отладки* - поиска ошибок в программе. Как правило, компиляция и отладка выполняются программистом в тесной взаимосвязи.

Возможны программные ошибки трех видов:

- *синтаксические* (ошибки в правилах языка);

- *алгоритмические* (ошибки в логике программы);

- *ошибки времени исполнения*, возникающие в процессе работы запущенной программы.

Компилятор способен найти только синтаксические ошибки, для выявления же алгоритмических ошибок служит этап *тестирования* программы. Ошибки времени исполнения возникают как результат некорректных действий пользователя, недопустимых операций над данными (например, попытки извлечь квадратный корень из отрицательного числа, поделить на ноль) или ошибок программного и аппаратного обеспечения ЭВМ.

5. Тестирование.



*Тестированием* называют проверку правильности работы программы на наборах "пробных" (*тестовых*) данных с заранее известным результатом. Конечно же, тестирование всей программы сразу возможно лишь для несложных учебных задач. Реальные программы, как правило, тестируются "по частям" - отдельными функциями и модулями.

#### 6. Документирование и поддержка.

Этот этап включает в себя создание справочной системы и документации к программе, возможно, расширение ее функциональности, выпуск новых версий, исправление ошибок, которые практически неизбежны в любой сложной программной системе. В наших учебных задачах этап поддержки будет отсутствовать.

### **Контрольные вопросы к Разделу 2**

1. Дайте определение понятию «информация».
2. Что такое кодирование и декодирование?
3. Что такое системы счисления? Виды систем счисления.
4. Что такое типы данных? Перечислите их.
5. Что такое структуры данных? Какими они бывают?
6. Назовите основные процессы обработки информации.
7. Как классифицируются ЭВМ по физическому представлению данных?
8. Дайте характеристику классов ЭВМ: АВМ, ЦВМ, ГВМ.
9. Назовите основные характеристики ЭВМ по поколениям.
10. Приведите классификацию ЭВМ по назначению.
11. Назовите области использования малых ЭВМ, микроЭВМ, суперЭВМ.
12. Что такое алгоритм?
13. Назовите основные свойства алгоритмов.
14. Назовите способы записи алгоритмов.
15. Что такое программа и программное обеспечение?
16. На какие власы делится программное обеспечение?
17. Перечислите этапы разработки программного обеспечения.

### **Раздел 3. Архитектура и структура вычислительных машин и систем**

#### **3.1 Базовые представления об архитектуре ЭВМ. Процессор, структура и функционирование**

##### **Базовые представления об архитектуре ЭВМ**

С середины 60-х годов существенно изменился подход к созданию вычислительных машин. Вместо независимой разработки аппаратуры и некоторых средств математического обеспечения стала проектироваться система, состоящая из совокупности **аппаратных** (*hardware*) и **программных** (*software*) средств. При этом на первый план выдвинулась концепция их взаимодействия. Так возникло принципиально новое понятие - **архитектура** ЭВМ.

Архитектурой компьютера считается его представление на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т. д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного запоминающего устройства (ОЗУ, ОП), внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Следует различать понятия: архитектура и структура ЭВМ. Архитектура ЭВМ включает в себя как структуру, отражающую состав ПК, так и программно - математическое обеспечение.

**Структура компьютера** — это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства — от основных логических узлов компьютера до простейших схем. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации

### **Принципы (архитектура) фон Неймана**

В основу построения большинства компьютеров положены следующие общие принципы, сформулированные в 1945 г. американским ученым Джоном фон Нейманом.

**1. Принцип программного управления.** Из него следует, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

Выборка программы из памяти осуществляется с помощью счетчика команд. Этот регистр процессора последовательно увеличивает хранимый в нем адрес очередной команды на длину команды. Так как команды программы расположены в памяти друг за другом, то тем самым организуется выборка цепочки команд из последовательно расположенных ячеек памяти.

Если после выполнения команды следует перейти не к следующей, а к какой-то другой, используются команды условного или безусловного переходов (ветвления), которые заносят в счетчик ячейки памяти, содержа-

щей следующую команду. Выборка команд из памяти прекращается после достижения и выполнения команды «стоп».

Таким образом, процессор исполняет программу автоматически, без вмешательства человека.

**2. Принцип однородности памяти.** Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными. Это открывает целый ряд возможностей. Например, программа в процессе своего выполнения также может подвергаться переработке, что позволяет задавать в самой программе правила получения некоторых ее частей (так в программе организуется выполнение циклов и подпрограмм). Более того, команды одной программы могут быть получены как результаты исполнения другой программы. На этом принципе основаны методы трансляции — перевода текста программы с языка программирования высокого уровня на язык конкретной машины.

**3. Принцип адресности.** Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти, так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

Компьютеры, построенные на этих принципах, относятся к типу фон-неймановских. Существуют и другие классы компьютеров, принципиально отличающиеся от фон-неймановских. Здесь, например, может не выполняться принцип программного управления, т. е. они могут работать без счетчика (регистра адреса) команд, указывающего на выполняемую команду программы. Для обращения к какой-либо переменной, хранящейся в памяти, этим компьютерам не обязательно давать ей имя. Такие компьютеры называются не-фон-неймановскими.

### **Логические узлы (агрегаты) ЭВМ**

**Центральное устройство.** ЦУ представляет основную компоненту ЭВМ и, в свою очередь, включает ЦП — центральный процессор (central processing unit - CPU) и ОП - оперативную (главную) память (main storage, core storage, random access memory - RAM).

**Процессор** непосредственно реализует операции обработки информации и управления вычислительным процессом, осуществляя выборку машинных команд и данных из оперативной памяти и запись в ОП, включение и отключение ВУ. Основными блоками процессора являются:

– устройство управления (УУ) с интерфейсом процессора (системой сопряжения и связи процессора с другими узлами машины);

- арифметико-логическое устройство (АЛУ);
- процессорная память (регистры, внутренний кэш).

**Оперативная память** предназначена для временного хранения данных и программ в процессе выполнения вычислительных и логических операций.

ЦУ описывается следующими характеристиками:

- длина машинного слова (разрядность, адресность);
- система команд;
- объем ОП;
- быстродействие (такты частота процессора, цикл записи/считывания ОП).

**Внешние устройства (ВУ).** ВУ обеспечивают эффективное взаимодействие компьютера с окружающей средой — пользователями, объектами управления, другими машинами. ВУ разделяются на следующие группы: интерактивные устройства (ввода/вывода); устройства хранения (массовые накопители); устройства массового ввода информации, устройства массового вывода информации.

В специализированных управляющих ЭВМ (технологические процессы, связь, ракеты и пр.) внешними устройствами ввода являются датчики (температуры, давления, расстояния и пр.), вывода — манипуляторы (гидро-, пневмо-, сервоприводы рулей, клапанов и др.).

В универсальных ЭВМ (человеко-машинная обработка информации) в качестве ВУ выступают терминалы, принтеры и др. устройства.

**Каналы связи** (внутримашинный интерфейс) служат для сопряжения центральных узлов машины с ее внешними устройствами.

Однотипные ЦУ и устройства хранения данных могут использоваться в различных типах машин. Известны примеры того, как фирмы, начавшие свою деятельность с производства управляющих машин, совершенствуя свою продукцию, перешли к выпуску систем, которые в зависимости от конфигурации ВУ могут исполнять как роль универсальных, так и управляющих машин (Hewlett-Packard и Digital Equipment Corporation).

### **Простейшие типы архитектур**

Если абстрагироваться от подробностей, то основные классические типы архитектур можно определить как следующие: «звезда», иерархическая, магистральная.

**Архитектура «звезда».** Здесь ЦУ соединено непосредственно с ВУ и управляет их работой (ранние модели машин).

**Классическая архитектура (фон Неймана)** - одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд — программа. Это однопроцессорный компьютер.

Вычислительная машина включает пять базовых компонент и состоит из следующих типов устройств:

- центральный процессор (ЦП), включающий АЛУ и УУ;
- запоминающие устройства — память, в том числе оперативная (ОП) и внешние ЗУ;
- устройства ввода и устройства вывода информации — внешние (периферийные) устройства (ВУ).

**Иерархическая архитектура** — ЦУ соединено с периферийными процессорами (вспомогательными процессорами, каналами и пр.), управляющими в свою очередь контроллерами, к которым подключены группы ВУ (системы IBM 360—375);

**Магистральная структура** (общая шина - unibas) — процессор (процессоры) и блоки памяти (ОП) взаимодействуют между собой и с ВУ (контроллерами ВУ) через внутренний канал, общий для всех устройств (машины DEC, ПЭВМ IBM PC-совместимые).

К этому типу архитектуры относится также архитектура персонального компьютера: функциональные блоки здесь связаны между собой общей шиной, называемой также системной магистралью.

Физически магистраль представляет собой многопроводную линию с гнездами для подключения электронных схем. Совокупность проводов магистралей разделяется на отдельные группы: шину адреса, шину данных и шину управления.

Периферийные устройства (принтер и др.) подключаются к аппаратуре компьютера через специальные контроллеры — устройства управления периферийными устройствами.

**Контроллер** — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

### **Процессор, структура и функционирование**

**Центральный процессор (ЦП)** - функционально-законченное программно-управляемое устройство, выполненное на одной или нескольких СБИС, предназначенное для выполнения арифметической и логической обработки информации программного управления работой устройств.

Упрощенная схема процессора, отражающая основные особенности архитектуры микроуровня, приведена на рисунке 8. Наиболее сложным функциональным устройством процессора является устройство управления выполнением команд. Оно содержит:

- **буфер команд**, который хранит одну или несколько очередных команд программы; читает следующие команды из запоминающего устрой-

ства, пока выполняется очередная команда, уменьшая время ее выборки из памяти;

- *дешифратор команд* расшифровывает код операции очередной команды и преобразует его в адрес начала микропрограммы, которая реализует исполнение команды;

- *управление выборкой очередной микрокоманды* представляет собой небольшой процессор, работающий по принципу фон Неймана, имеет свой счетчик микрокоманд, который автоматически выбирает очередную микрокоманду из ПЗУ микрокоманд;

- *постоянное запоминающее устройство (ПЗУ) микрокоманд* - это запоминающее устройство, в которое информация записывается однократно и затем может только считываться; отличительной особенностью ПЗУ является то, что записанная в него информация сохраняется сколь угодно долго и не требует постоянного питающего напряжения.

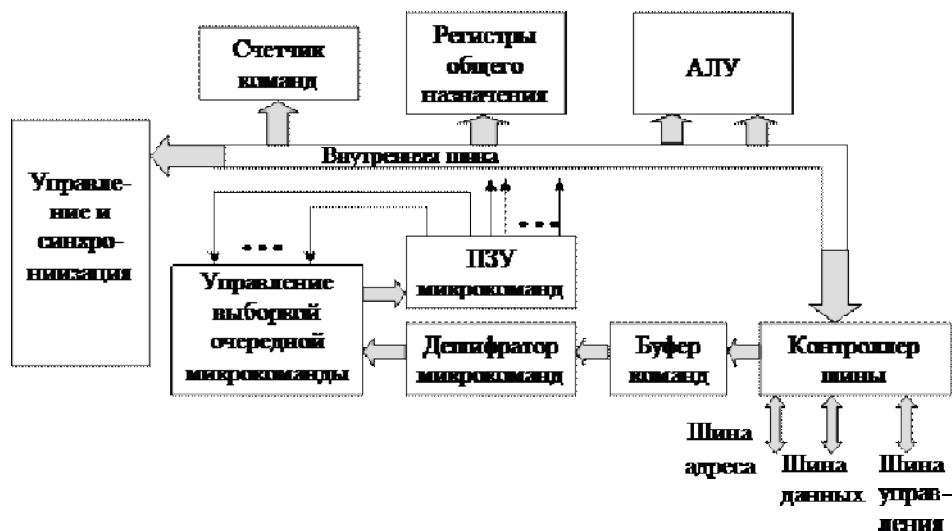


Рисунок 8 – Архитектура процессора

Поступивший от дешифратора команд адрес записывается в *счетчик микрокоманд устройства выборки*, и начинается процесс обработки последовательности микрокоманд. Каждый разряд микрокоманды связан с одним управляющим входом какого-либо функционального устройства.

Выборка очередной микрокоманды осуществляется через определенный интервал времени, который, в свою очередь, зависит от времени выполнения предыдущей микрокоманды. Частота, с которой осуществляется выборка микрокоманд, называется *тактовой частотой процессора*. Тактовая частота является важной характеристикой процессора, так как определяет скорость выполнения процессором команд, и, в конечном итоге, *быстродействие процессора*. Другими словами *тактовая частота показывает количество элементарных операций, выполняемых центральным*

процессором в секунду. Под элементарной операцией понимается любая простейшая операция типа сложения, пересылки, сравнения и т.д.

**Арифметико-логическое устройство** (АЛУ) предназначено для выполнения арифметических и логических операций преобразования информации. Функционально АЛУ состоит из нескольких специальных регистров, полноразрядного сумматора и схем местного управления.

**Регистры общего назначения** (РОН) используются для временного хранения операндов исполняемой команды и результатов вычислений, а также хранят адреса ячеек памяти или портов ввода-вывода для команд, обращающихся к памяти и внешним устройствам. Необходимо отметить, что если операнды команды хранятся в РОН, то время выполнения команды значительно сокращается.

Основными параметрами процессоров являются: *рабочее напряжение, разрядность, рабочая тактовая частота, коэффициент внутреннего умножения тактовой частоты и размер кэш-памяти.*

*Рабочее напряжение* процессора обеспечивает материнская плата, поэтому разным маркам процессоров соответствуют разные материнские платы (их надо выбирать совместно).

*Разрядность процессора* показывает, сколько бит данных он может принять и обработать в своих регистрах за один раз (за один такт) и *определяется разрядностью этих регистров.*

Обмен данными внутри процессора происходит в несколько раз быстрее, чем обмен с другими устройствами, например, с оперативной памятью. Для того чтобы уменьшить количество обращений к оперативной памяти, внутри процессора создают буферную область - так называемую *кэш-память*. Это как бы «сверхоперативная память». Когда процессору нужны данные, он сначала обращается в кэш-память, и только если там нужных данных нет, происходит его обращение в оперативную память. Принимая блок данных из оперативной памяти, процессор заносит его одновременно и в кэш-память. Высокопроизводительные процессоры имеют повышенный объем кэш-памяти.

### **3.2 Технологии повышения производительности процессоров**

В последние несколько лет появились технологии и методы повышения производительности процессора. Разработчики потратили много усилий на развитие усовершенствований, которыми мы пользуемся каждый раз, включая компьютер. Сейчас мы поговорим о функциях, которые увеличивают производительность современных процессоров.

#### **Суперскалярная архитектура**

Программные инструкции (команды) обрабатываются электронными схемами, называемыми операционными блоками или исполнительными устройствами. Термин *суперскалярная архитектура* означает одновременное использование нескольких операционных блоков, что позволяет центральному процессору выполнять несколько инструкций за один машинный такт (цикл). Например, в процессоре Pentium Pro используются два операционных конвейера (их называют «U» и «V»). Это является формой многопроцессорности внутри самого центрального процессора, поскольку несколько часто выполняемых операций выполняются параллельно несколькими устройствами. Большинство современных процессоров являются суперскалярными на том или ином уровне. За счет сочетания конвейерной обработки команд с несколькими операционными блоками в суперскалярной архитектуре центрального процессора удается достигать чрезвычайно эффективного использования каждого машинного такта.

### **Конвейерная обработка**

Центральный процессор обрабатывает команды и генерирует результат их выполнения посредством сложных серий переключений транзисторов внутри самого кристалла процессора. Первые процессоры выполняли последовательно одну команду за другой. Каждая команда выбиралась из памяти и полностью выполнялась, затем выбиралась следующая команда. Обработка могла занимать несколько машинных тактов (в зависимости от команды). Простые команды могли выполняться за 2 или 3 такта, а сложные команды требовали для своего выполнения от 2 до 7 тактов.

Конвейерная обработка (командный конвейер) позволяет начинать обработку следующей команды еще до окончания выполнения текущей команды. Таким образом, центральный процессор за один машинный такт может обрабатывать несколько команд.

Иными словами, в каждый такт в конвейере могут находиться несколько команд. Конвейерный метод обработки обеспечивает загрузку операционных блоков, не занятых обработкой текущей команды. В то же время центральный процессор может выдавать в каждом такте результат обработки только одной команды.

### **Суперконвейер**

Как было сказано выше, команды обрабатываются в конвейере, каждая часть которого выполняет определенную операцию над командой. Если сделать конвейер более длинным (увеличить количество шагов обработки команды), то на каждом шаге обработки будет выполняться меньше работы (а значит и за меньший промежуток времени) и можно будет увеличить тактовую частоту процессора. Такая технология известна под названием *суперконвейера* и является усовершенствованием простого конвейера. Длина суперконвейера составляет до 10 шагов. Процессоры Intel Pentium 4 используют *гиперконвейер* длиной в 20 шагов. Увеличение дли-



ны конвейера накладывает ограничение на программу — чтобы исполняться наиболее эффективным образом, компиляторы должны учитывать особенности конвейерной архитектуры процессоров.

### **Спекулятивное выполнение и предсказание переходов**

Некоторые процессоры обладают способностью одновременного выполнения нескольких команд. В ряде случаев не все результаты обработки этих команд будут использоваться, поскольку ветвление программы может привести к тому, что часть уже загруженных в конвейер команд не должна была исполняться. Такое часто наблюдается на участках программ вблизи команд условных переходов — где проверяется некоторое условие, и дальнейшее выполнение программы зависит от проверки выполнения этого условия (условный оператор в любом языке программирования). Ветвление программы представляет реальную проблему для конвейера команд, поскольку нет гарантии в том, что программа будет далее выполняться линейно (т.е. не будет выполнена команда перехода на другую часть программы). Менее «интеллектуальные» процессоры останавливают конвейер до того момента, когда будет известен результат проверки условия ветвления программы, что приводит к падению производительности. Более совершенные процессоры будут продолжать обрабатывать конвейер команд в предположении, что выполнение программы продолжится без ветвления.

Еще более совершенные процессоры обладают способностью предсказывать ветвление программы (с достаточно хорошей точностью) на основе анализа предыдущей истории выполнения данного участка программы. Механизм предсказания программных переходов улучшает обработку ветвлений программы. При этом используется специальная небольшая кэш-память, называемая целевым буфером ветвлений. Когда процессор обрабатывает команду перехода, то он запоминает информацию о ней в этой памяти. Если процессор в следующий раз встретит эту команду перехода, то он может уже «догадаться» (на основе записанной информации) о направлении ветвления программы в этом месте. Это позволяет не останавливать конвейер и повышает производительность процессора.

### **Динамическое выполнение команд**

Даже самый быстрый процессор выполняет команды в том порядке, в котором они располагаются в конкретной программе. Это означает, что неправильно или неэффективно написанная программа будет снижать производительность центрального процессора. Во многих случаях даже хорошо написанная программа ухудшается в процессе ее трансляции в машинные команды. Метод динамического исполнения позволяет процессору оценивать последовательность команд программы и «выбирать» лучшую последовательность обработки команд. Например, команда 2 может быть выполнена раньше окончания обработки команды 1. Результаты же

выполнения команд располагаются в первоначальном порядке для обеспечения правильного выполнения программы. При неграмотном написании программы такое выборочное переупорядочивание команд позволяет процессору лучше использовать свои ресурсы, что повышает его производительность.

### **Переименование регистров и буфера записи**

Методика переименования регистров используется для организации нескольких процессов обработки команд различными операционными блоками, пытающимися использовать одни и те же регистры. Вместо того чтобы довольствоваться единственным набором регистров, используется несколько наборов регистров. Это позволяет различным операционным блокам работать одновременно, без ненужных приостановок в работе конвейера. Буфера записи используются для хранения результатов выполнения команд до тех пор, пока эти результаты не будут опять переписаны в регистры или в память. Чем больше буферов записи, тем больше команд могут выполняться без остановки конвейеров.

### **Многопроцессорность**

Многопроцессорность — это методика организации работы нескольких процессоров в одной системе. Идея состоит в удвоении производительности системы при использовании двух процессоров вместо одного, или повышении производительности в 4 раза при использовании 4-х процессоров и т.д. На практике дело обстоит не так просто, но в определенных условиях многопроцессорность улучшает производительность системы. Для эффективного использования многопроцессорности главный компьютер должен удовлетворять следующим требованиям:

- *Поддержка со стороны системной платы.* Системная плата должна располагать дополнительными процессорными разъемами для установки нескольких процессоров, а комплект микросхем должен обеспечивать управление многопроцессорной конфигурацией.

- *Поддержка со стороны процессора.* Процессоры должны быть приспособлены для работы в многопроцессорных системах. Для подбора соответствующих процессоров необходимо обратиться за помощью к документации на системную плату.

- *Поддержка со стороны операционной системы.* Многопроцессорные системы обслуживают такие операционные системы как Windows NT/2000/XP или UNIX. Windows 98 не поддерживает многопроцессорность.

Многопроцессорные компьютеры хороши для выполнения на них специального прикладного программного обеспечения. Многопроцессорный компьютер работает под управлением операционной системы, которая распределяет различные задачи по разным процессорам компьютера. Прикладные программы, написанные для многопроцессорного компьютера,

должны состоять из отдельных потоков, которые могли бы выполняться независимо друг от друга. Это дает возможность операционной системе запускать их на разных процессорах одновременно и за счет этого увеличивать производительность компьютера в целом. Если же прикладное программное обеспечение не отвечает требованиям многопроцессорной системы, то такая система не даст преимущества (хотя операционная система сможет использовать дополнительные процессоры в том случае, когда необходимо выполнять несколько приложений одновременно).

Многопроцессорные системы могут быть *асимметричными* или *симметричными*. Эти термины характеризуют то, как операционная система распределяет задачи между процессорами компьютера. В асимметричных системах некоторые процессоры заняты выполнением только системных задач, а другие процессоры выполняют только прикладные программы. При жестком распределении процессоров по типам задач наблюдается снижение производительности в те периоды, когда компьютеру необходимо выполнять больше системных задач, чем прикладных или наоборот. Симметричная многопроцессорность (SMP — symmetric multiprocessing) позволяет на любом процессоре выполнять любые задачи — системные или прикладные. Это более гибкий подход построения многопроцессорных систем, и он позволяет достичь большей производительности. Большинство многопроцессорных системных плат для персональных компьютеров предназначены для построения симметричных многопроцессорных систем.

*Для того чтобы процессор мог работать в многопроцессорном компьютере в режиме SMP, он должен поддерживать многопроцессорный протокол, который определяет способ общения процессоров друг с другом и с системным комплектом микросхем. Процессоры Intel используют протокол SMP под названием «APIC», а комплекты микросхем Intel, которые поддерживают многопроцессорность, разработаны для реализации этого протокола. Протокол APIC является патентованным стандартом компании Intel. Поэтому хотя процессоры AMD и Cyrix и являются совместимыми с процессорами Intel, они не могут использовать этот протокол в SMP-конфигурациях. Компании AMD и Cyrix разработали свой собственный SMP-протокол под названием «OpenPIC».*

### **Hyper-Threading**

Intel объединила технологии гиперконвейера и многопроцессорности в одной из своих последних разработок — технологии **Hyper-Threading** (HT). Процессоры с использованием HT ведут себя как два независимых процессора. При этом они могут параллельно выполнять два потока команд за счет большой длины конвейера. Конечно, при этом производительность процессора не удваивается, но за счет сокращения простоя отдельных блоков процессора удается поднять производительность не менее

чем на 30-40%, что дает существенный прирост в многозадачной среде. Следует заметить, что использование НТ может и снизить производительность, если оба виртуальных процессора будут все время конкурировать за исполнительные устройства единственного физического процессора. В частности, в некоторых случаях использование НТ может снизить производительность Microsoft SQL Server.

### **Мультимедийные расширения**

С увеличением количества мультимедийных программ (графических приложений, презентаций и т.п.) для проведения интенсивных вычислений стало не хватать пропускной способности процессора. Возникла потребность в увеличении скорости выполнения некоторых вычислительных операций, необходимых для выполнения мультимедийных и коммуникационных приложений. В то время как эти операции составляют не более 10% объема программы, их выполнение занимало до 90% времени. Компании Intel и AMD стали состязаться в создании лучших «мультимедийных расширений» для своих процессоров.

### **MMX**

В 1996 году компания Intel ввела в процессоры семейства Pentium (назвав их «Pentium MMX») реализацию 57 новых команд, назвав их мультимедийными расширениями (MMX — multimedia extensions). MMX-команды обрабатывают несколько элементов целочисленных данных параллельно, используя метод под названием «одна команда — много данных» (SIMD — Single Instruction Multiple Data). С помощью этой технологии процессор может обрабатывать одновременно большое количество данных, за счет чего уменьшается время обработки видео и звуковой информации, присутствующей в мультимедийных приложениях. Следующие модели процессоров Intel (Pentium I I/I I/I V и Celeron) также поддерживают обработку набора MMX-команд. Команды MMX наиболее эффективны при обработке 2-х мерных изображений и звука.

## **3.3 Организация оперативной памяти**

Как уже говорилось ранее, оперативная память предназначена для временного хранения данных и программ в процессе выполнения вычислительных и логических операций.

Необходимо отметить, что все распространенные операционные системы, если для работы нужно больше памяти, чем физически присутствует в компьютере, не прекращают работу, а сбрасывают неиспользуемое в данный момент содержимое памяти в дисковый файл (называемый свопом - swap) и затем по мере необходимости «перегоняют» данные между ОП и

свопом. Это гораздо медленнее, чем доступ системы к самой ОП. Поэтому от количества оперативной памяти напрямую зависит скорость системы.

### ***Адресация данных***

Команды, исполняемые ЭВМ при выполнении программы, равно как и числовые и символьные операнды, хранятся в памяти компьютера. Память состоит из миллионов ячеек, в каждой из которых содержится один бит информации (значения 0 или 1). Биты редко обрабатываются поодиночке, а, как правило, группами фиксированного размера. Для этого память организуется таким образом, что группы по  $n$  бит могут записываться и считываться за одну операцию. Группа  $n$  бит называется словом, а значение  $n$  - длиной слова. Схематически память компьютера можно представить в виде массива слов.

Обычно длина машинного слова компьютеров составляет от 16 до 64 бит. Если длина слова равна 32 битам, в одном слове может храниться 32-разрядное число в дополнительном коде или четыре символа ASCII, занимающих 8 бит каждый. Восемь идущих подряд битов являются байтом. Для представления машинной команды требуется одно или несколько слов.

### ***Байтовая адресация***

Отдельные биты, как правило, не адресуются и чаще всего адреса назначаются байтам памяти. Память, в которой каждый байт имеет отдельный адрес, называется памятью с байтовой адресацией. Последовательные байты имеют адреса 0, 1, 2 и так далее. Таким образом, при использовании слов длиной 32 бита последовательные слова имеют адреса 1, 4, 8, ..., и каждое слово состоит из 4 байт.

### ***Прямой и обратный порядок байтов***

Существует два способа адресации байтов в словах:

- *в прямом порядке*. Прямым порядком байтов (little-endian) называется система адресации, при которой байты адресуются справа налево, так что наименьший адрес имеет самый младший байт слова (расположенный с правого края);

- *в обратном порядке*. Обратным порядком байтов (big-endian) называется система адресации, при которой байты адресуются слева направо, так что самый старший байт слова (расположенный с левого края) имеет наименьший адрес.

### ***Адресное пространство***

Для доступа к памяти необходимы имена или адреса, определяющие расположение данных в памяти. В качестве адресов традиционно используются числа из диапазона от 0 до  $2^k - 1$  со значением  $k$ , достаточным для адресации всей памяти компьютера. Все  $2^k$  адресов составляют адресное пространство компьютера. Следовательно, память состоит из  $2^k$  адресуемых элементов.

Адресное пространство ЭВМ графически может быть изображено прямоугольником, одна из сторон которого представляет разрядность адресуемой ячейки (слова) процессора, а другая сторона - весь диапазон доступных адресов для этого же процессора. Диапазон доступных адресов процессора определяется разрядностью шины адреса системной шины. При этом минимальный номер ячейки памяти (адрес) будет равен 0, а максимальный определяется из формулы  $M = 2^n - 1$ .

### ***Иерархическая организация памяти***

Компромиссом между производительностью и объемами памяти является решение использовать иерархию запоминающих устройств, то есть применять иерархическую модель памяти.

Применение иерархических систем памяти оправдывает себя вследствие двух важных факторов - принципа локальности обращений и низкого (экономически выгодного) соотношения стоимость/производительность. Принцип локальности обращений состоит в том, что большинство программ обычно не выполняют обращений ко всем своим командам и данным равновероятно, а в каждый момент времени оказывают предпочтение некоторой части своего адресного пространства.

Иерархия памяти обычно состоит из многих уровней, но в каждый момент времени взаимодействуют только два близлежащих уровня. Минимальная единица информации, которая может присутствовать либо отсутствовать в двухуровневой иерархии, называется блоком или строкой.

### ***Кэш-память***

Кэш-память или cache memory - компонент иерархической памяти - представляет собой буферное ЗУ, работающее со скоростью, обеспечивающей функционирование ЦП без режимов ожидания.

Необходимость создания кэш-памяти возникла потому, что появились процессоры с высоким быстродействием. Между тем для выполнения сложных прикладных процессов нужна большая память. Использование же большой сверхскоростной памяти экономически невыгодно. Поэтому между ОП и процессором стали устанавливать меньшую по размерам высокоскоростную буферную память, или кэш-память. В дальнейшем она была разделена на два уровня - встроенная в процессор (on-die) и внешняя (on-motherboard).

Существует три основных способа размещения блоков (строк) основной памяти в кэше:

- кэш-память с прямым отображением (direct-mapped cache). В этом случае каждый блок основной памяти имеет только одно фиксированное место, на котором он может появиться в кэш-памяти. Все блоки основной памяти, имеющие одинаковые младшие разряды в своем адресе, попадают в один блок кэш-памяти. Этот тип памяти наиболее прост, но и наименее эффективен, так как данные из разных областей памяти могут конфликто-

вать из-за единственной строки кэша, где они только и могут быть размещены;

- полностью ассоциативная кэш-память (fully associative cache). Может отображать содержимое любой области памяти в любую область кэша, но при этом крайне сложна в схемотехнике;

частично ассоциативная (или множественно ассоциативная, partial associative, set-associative cache) кэш-память. Является наиболее распространенным в данный момент среди процессорных архитектур. Характеризуется тем или иным количеством  $n$  «каналов» (степенью ассоциативности, « $n$ -way») и может отображать содержимое данной строки памяти на каждую из  $n$  своих строк. Этот вариант является разумным компромиссом между полностью ассоциативным и кэшем «прямого отображения».

В современных процессорах, как правило, используется либо кэш-память с прямым отображением, либо двух- (четырёх-) канальная множественно ассоциативная кэш-память.

### **3.4 Дисковые массивы и уровни RAID. Внешние устройства**

#### **Дисковые массивы**

Впервые термин «RAID-массив» появился в 1987 году, когда американские исследователи Паттерсон, Гибсон и Катц из Калифорнийского университета Беркли в своей статье «Избыточный массив недорогих дисков» (“A Case for Redundant Arrays of Inexpensive Discs, RAID”) описали, каким образом можно объединить несколько дешевых жестких дисков в одно логическое устройство так, чтобы в результате повышались емкость и быстродействие системы, а отказ отдельных дисков не приводил к отказу всей системы.

Итак, RAID — это избыточный массив независимых дисков (Redundant Arrays of Independent Discs), на который возлагается задача обеспечения отказоустойчивости и повышения производительности. Отказоустойчивость достигается за счет избыточности. То есть часть емкости дискового пространства отводится для служебных целей, становясь недоступной для пользователя.

Повышение производительности дисковой подсистемы обеспечивается одновременной работой нескольких дисков, и в этом смысле чем больше дисков в массиве (до определенного предела), тем лучше.

Совместную работу дисков в массиве можно организовать с помощью либо параллельного, либо независимого доступа. При параллельном доступе дисковое пространство разбивается на блоки (полоски) для записи данных. Аналогично информация, подлежащая записи на диск, разбивается на такие же блоки. При записи отдельные блоки записываются на раз-

ные диски, причем запись нескольких блоков на различные диски происходит одновременно, что и приводит к увеличению производительности в операциях записи. Нужная информация также считывается отдельными блоками одновременно с нескольких дисков, что тоже способствует росту производительности пропорционально количеству дисков в массиве.

Следует отметить, что модель с параллельным доступом реализуется только при условии, что размер запроса на запись данных больше размера самого блока. В противном случае осуществлять параллельную запись нескольких блоков практически невозможно.

Если же размер записываемых данных меньше размера блока, то реализуется принципиально иная модель — независимый доступ. Более того, эта модель может использоваться и в том случае, когда размер записываемых данных больше размера одного блока. При независимом доступе все данные отдельного запроса записываются на отдельный диск, то есть ситуация идентична работе с одним диском. Преимущество модели с независимым доступом в том, что при одновременном поступлении нескольких запросов на запись (чтение) все они будут выполняться на отдельных дисках независимо друг от друга. Подобная ситуация типична, например, для серверов.

В соответствии с различными типами доступа существуют и разные типы RAID-массивов, которые принято характеризовать уровнями RAID. Кроме типа доступа, уровни RAID различаются способом размещения и формирования избыточной информации. Избыточная информация может либо размещаться на специально выделенном диске, либо распределяться между всеми дисками. Способов формирования этой информации достаточно много. Простейший из них — это полное дублирование (100-процентная избыточность), или зеркалирование. Кроме того, используются коды с коррекцией ошибок, а также вычисление четности.

### **Уровни RAID-массивов**

В настоящее время существует несколько RAID-уровней, которые можно считать стандартизованными, — это RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5 и RAID 6.

Применяются также различные комбинации RAID-уровней, что позволяет объединить их достоинства. Обычно это комбинация какого-либо отказоустойчивого уровня и нулевого уровня, применяемого для повышения производительности (RAID 1+0, RAID 0+1, RAID 50).

### **RAID 0**

RAID уровня 0, строго говоря, не является избыточным массивом и соответственно не обеспечивает надежности хранения данных. Тем не менее, данный уровень активно применяется в случаях, когда необходимо обеспечить высокую производительность дисковой подсистемы. При со-



здании RAID-массива уровня 0 информация разбивается на блоки (иногда эти блоки называют *страйпами* (stripe)), которые записываются на отдельные диски, то есть создается система с параллельным доступом (если, конечно, это позволяет размер блока). Благодаря возможности одновременного ввода-вывода с нескольких дисков, RAID 0 обеспечивает максимальную скорость передачи данных и максимальную эффективность использования дискового пространства, поскольку не требуется места для хранения контрольных сумм. Реализация этого уровня очень проста. В основном RAID 0 применяется в тех областях, где требуется быстрая передача большого объема данных.

### **RAID 1: Зеркальные диски.**

Зеркальные диски представляют традиционный способ повышения надежности магнитных дисков. Это наиболее дорогостоящий из рассматриваемых способов, так как все диски дублируются и при каждой записи информация записывается также и на проверочный диск. Таким образом, приходится идти на некоторые жертвы в пропускной способности ввода/вывода и емкости памяти ради получения более высокой надежности. Зеркальные диски широко применяются многими фирмами.

Дублирование всех дисков может означать удвоение стоимости всей системы или, иначе, использование лишь 50% емкости диска для хранения данных. Повышение емкости, на которое приходится идти, составляет 100%. Такая низкая экономичность привела к появлению следующего уровня RAID.

### **RAID 2: матрица с поразрядным расслоением**

Один из путей достижения надежности при снижении потерь емкости памяти может быть подсказан организацией основной памяти, в которой для исправления одиночных и обнаружения двойных ошибок используются избыточные контрольные разряды. Такое решение можно повторить путем поразрядного расслоения данных и записи их на диски группы, дополненной достаточным количеством контрольных дисков для обнаружения и исправления одиночных ошибок. Один диск контроля четности позволяет обнаружить одиночную ошибку, но для ее исправления требуется больше дисков.

Такая организация обеспечивает лишь один поток ввода/вывода для каждой группы независимо от ее размера. Группы большого размера приводят к снижению избыточной емкости, идущей на обеспечение отказоустойчивости, тогда как при организации меньшего числа групп наблюдается снижение операций ввода/вывода, которые могут выполняться матрицей параллельно.

При записи больших массивов данных системы уровня 2 имеют такую же производительность, что и системы уровня 1, хотя в них использу-

ется меньше контрольных дисков и, таким образом, по этому показателю они превосходят системы уровня 1.

Таким образом, RAID уровня 2 предпочтительны для суперкомпьютеров, но не подходят для обработки транзакций.

### **RAID 3: аппаратное обнаружение ошибок и четность**

Большинство контрольных дисков, используемых в RAID уровня 2, нужны для определения положения неисправного разряда. Эти диски становятся полностью избыточными, так как большинство контроллеров в состоянии определить, когда диск отказал, при помощи специальных сигналов, поддерживаемых дисковым интерфейсом, либо при помощи дополнительного кодирования информации, записанной на диск и используемой для исправления случайных сбоев. По существу, если контроллер может определить положение ошибочного разряда, то для восстановления данных требуется лишь один бит четности. Уменьшение числа контрольных дисков до одного на группу снижает избыточность емкости до вполне разумных размеров. Часто количество дисков в группе равно 5 (4 диска данных плюс 1 контрольный). Каждое из таких устройств воспринимается машиной как отдельный логический диск с учетверенной пропускной способностью, учетверенной емкостью и значительно более высокой надежностью.

### **RAID 4: внутригрупповой параллелизм**

RAID уровня 4 повышает производительность передачи небольших объемов данных за счет параллелизма, давая возможность выполнять более одного обращения по вводу/выводу к группе в единицу времени. Логические блоки передачи в данном случае не распределяются между отдельными дисками, вместо этого каждый индивидуальный блок попадает на отдельный диск.

Достоинство поразрядного расслоения состоит в простоте вычисления кода Хэмминга, что необходимо для обнаружения и исправления ошибок в системах уровня 2. В RAID уровня 3 обнаружение ошибок диска с точностью до сектора осуществляется дисковым контроллером. Следовательно, если записывать отдельный блок передачи в отдельный сектор, то можно обнаружить ошибки отдельного считывания без доступа к дополнительным дискам. Главное отличие между системами уровня 3 и 4 состоит в том, что в последних расслоение выполняется на уровне сектора, а не на уровне битов или байтов.

В системах уровня 4 обновление контрольной информации реализовано достаточно просто. Для вычисления нового значения четности требуются лишь старый блок данных, старый блок четности и новый блок данных:

новая четность = (старые данные хог новые данные) хог старая четность

В системах уровня 4 для записи небольших массивов данных используются два диска, которые выполняют четыре выборки (чтение данных плюс четности, запись данных плюс четности). Производительность групповых операций записи и считывания остается прежней, но при небольших (на один диск) записях и считываниях производительность существенно улучшается. К сожалению, улучшение производительности оказывается недостаточной для того, чтобы этот метод мог занять место системы уровня 1.

#### **RAID 5: четность вращения для распараллеливания записей**

RAID уровня 4 позволяли добиться параллелизма при считывании отдельных дисков, но запись по-прежнему ограничена возможностью выполнения одной операции на группу, так как при каждой операции должны выполняться запись и чтение контрольного диска. Система уровня 5 улучшает возможности системы уровня 4 посредством распределения контрольной информации между всеми дисками группы.

Это небольшое изменение оказывает огромное влияние на производительность записи небольших массивов информации. Если операции записи могут быть спланированы так, чтобы обращаться за данными и соответствующими им блоками четности к разным дискам, появляется возможность параллельного выполнения  $N/2$  записей, где  $N$  - число дисков в группе. Данная организация имеет одинаково высокую производительность при записи и при считывании как небольших, так и больших объемов информации, что делает ее наиболее привлекательной в случаях смешанных применений.

#### **RAID 6: Двумерная четность для обеспечения большей надежности**

Этот пункт можно рассмотреть в контексте соотношения отказоустойчивость/пропускная способность. RAID 5 предлагают, по существу, лишь одно измерение дисковой матрицы, вторым измерением которой являются секторы. Теперь рассмотрим объединение дисков в двумерный массив таким образом, чтобы секторы являлись третьим измерением. Мы можем иметь контроль четности по строкам, как в системах уровня 5, а также по столбцам, которые, в свою очередь, могут расслаиваться для обеспечения возможности параллельной записи. При такой организации можно преодолеть любые отказы двух дисков и многие отказы трех дисков. Однако при выполнении логической записи реально происходит шесть обращений к диску: за старыми данными, за четностью по строкам и по столбцам, а также для записи новых данных и новых значений четности. Для некоторых применений с очень высокими требованиями к отказоустойчивости такая избыточность может оказаться приемлемой, однако для традиционных суперкомпьютеров и для обработки транзакций данный метод не подойдет.

## **Выводы**

В общем случае, если доминируют короткие записи и считывания и стоимость емкости памяти не является определяющей, наилучшую производительность демонстрируют системы RAID уровня 1. Однако если стоимость емкости памяти существенна, либо если можно снизить вероятность появления коротких записей (например, при высоком коэффициенте отношения числа считываний к числу записей, при эффективной буферизации последовательностей считывания-модификации-записи, либо при приведении коротких записей к длинным с использованием стратегии кэширования файлов), RAID уровня 5 могут обеспечить очень высокую производительность, особенно в терминах отношения стоимость/ производительность.

## **3.5 Создание конфигурации массивов RAID**

Все современные материнские платы оснащены интегрированным RAID-контроллером, предоставляющим пользователю возможность создания RAID-массива из нескольких дисков.

### **Создание RAID-массива на базе контроллера GIGABYTE SATA2**

Рассмотрим создание RAID-массива на базе двух дисков с использованием интегрированного на плате RAID-контроллера GIGABYTE SATA2.

Доступ в меню настройки контроллера возможен на этапе загрузки системы, для чего нужно нажать комбинацию клавиш Ctrl+G, когда появится соответствующая надпись на экране. Естественно, прежде в настройках BIOS нужно определить режим работы двух SATA-портов, относящихся к контроллеру GIGABYTE SATA2, как RAID (в противном случае доступ в меню конфигуратора RAID-массива будет невозможен).

Меню настройки RAID-контроллера GIGABYTE SATA2 довольно простое. Контроллер является двухпортовым и позволяет создавать RAID-массивы уровня 0 или 1. Через меню настройки контроллера можно удалить или создать RAID-массив. При создании RAID-массива имеется возможность указать его название, выбрать уровень массива (0 или 1), задать размер страйпа для RAID 0 (128, 84, 32, 16, 8 или 4К), а также определить размер массива.

Если массив создан, то какие-либо изменения в нем уже невозможны. То есть нельзя впоследствии для созданного массива изменить, например, его уровень или размер страйпа. Для этого прежде нужно удалить массив (с потерей данных), а потом создать его заново. Собственно, это свойственно не только контроллеру GIGABYTE SATA2. Невозможность изменения параметров созданных RAID-массивов — особенность всех

контроллеров, которая вытекает из самого принципа реализации RAID-массива.

После того как массив на базе контроллера GIGABYTE SATA2 создан, текущую информацию о нем можно просмотреть, используя утилиту GIGABYTE RAID Configurer, которая устанавливается автоматически вместе с драйвером.

### **Создание RAID-массива на базе контроллера Marvell 9128**

Конфигурирование RAID-контроллера Marvell 9128 возможно только через настройки BIOS платы Gigabyte GA-EX58A-UD7.

Несмотря на то что этот контроллер поддерживает работу с дисками с интерфейсом SATA III, он также полностью совместим с дисками с интерфейсом SATA II.

Контроллер Marvell 9128 позволяет создать RAID-массив уровней 0 и 1 на базе двух дисков. Для массива уровня 0 можно задать размер страйпа 32 или 64 Кбайт, а также указать имя массива. Кроме того, имеется и такая опция, как Gigabyte Rounding. Функция Gigabyte Rounding определена только для RAID-массива уровня 1. Она позволяет использовать для создания RAID-массива уровня 1 два диска (например, различных производителей или разные модели), емкость которых немного отличается друг от друга. Функция Gigabyte Rounding как раз и задает разницу в размерах двух дисков, применяемых для создания RAID-массива уровня 1. В контроллере Marvell 9128 функция Gigabyte Rounding позволяет установить разницу в размерах дисков 1 или 10 Гбайт.

### **Создание RAID-массива на базе контроллера, интегрированного в ICH10R**

RAID-контроллер, интегрированный в южный мост ICH10R, является самым распространенным. Данный RAID-контроллер 6-портовый и поддерживает не только создание массивов RAID 0 и RAID 1, но также RAID 5 и RAID 10.

Доступ в меню настройки контроллера возможен на этапе загрузки системы, для чего нужно нажать комбинацию клавиш Ctrl+I, когда появится соответствующая надпись на экране. Естественно, прежде в настройках BIOS следует определить режим работы этого контроллера как RAID (в противном случае доступ в меню конфигуратора RAID-массива будет невозможен).

Меню настройки RAID-контроллера достаточно простое. Через меню настройки контроллера можно удалить или создать RAID-массив. При создании RAID-массива можно указать его название, выбрать уровень массива (0, 1, 5 или 10), задать размер страйпа для RAID 0 (128, 84, 32, 16, 8 или 4К), а также определить размер массива.

### **Тестирование RAID-массивов**

Созданные RAID-массивы можно протестировать с помощью утилиты Iometer (Input/Output meter), которая является своеобразным отраслевым стандартом для измерения производительности дисковых систем. Данная утилита бесплатна, и ее можно скачать с ресурса <http://www.iometer.org>.

Утилита Iometer является синтетическим тестом и позволяет работать с неразбитыми на логические разделы жесткими дисками, благодаря чему можно тестировать диски независимо от файловой структуры и свежести к нулю влияние операционной системы.

При тестировании возможно создание специфической модели доступа, или «паттерна», которая позволяет конкретизировать выполнение жестким диском специфических операций. В случае создания конкретной модели доступа разрешается менять следующие параметры:

- размер запроса на передачу данных;
- случайное/последовательное распределение (в %);
- распределение операций чтения/записи (в %);
- количество отдельных операций ввода-вывода, работающих параллельно.

Утилита Iometer не требует инсталляции на компьютер и состоит из двух частей: собственно Iometer и Dynamo.

Iometer — это контролирующая часть программы с пользовательским графическим интерфейсом, позволяющим производить все необходимые настройки. Dynamo — это генератор нагрузки, который не имеет интерфейса. Каждый раз при запуске файла Iometer.exe автоматически запускается и генератор нагрузки Dynamo.exe.

Утилита Iometer позволяет производить тестирование не только локальных дисковых систем (DAS), но и сетевых накопителей (NAS). К примеру, с ее помощью можно протестировать производительность дисковой подсистемы сервера (файл-сервера), используя для этого несколько сетевых клиентов.

Для тестирования RAID-массивов с помощью утилиты Iometer необходимо сначала создать сценарии нагрузки последовательного чтения, последовательной записи, выборочного чтения и выборочной записи.

Затем можно приступить непосредственно к тестированию, нажав на кнопку с изображением флажка. При этом будет предложено указать название файла с результатами тестирования и выбрать его местоположение. Результаты тестирования сохраняются в CSV-файле, который потом легко экспортировать в Excel и, установив фильтр по первому столбцу, выбрать нужные данные с результатами тестирования.

## Контрольные вопросы к Разделу 3

1. Что такое архитектура ЭВМ?
2. Что такое структура компьютера?
3. Перечислите принципы фон Неймана.
4. Назовите основные логические узлы ЭВМ.
5. Назовите основные классические типы архитектур.
6. Что такое центральный процессор?
7. Что входит в архитектуру процессора?
8. Что является основными параметрами процессоров?
9. Перечислите технологии и методы повышения производительности процессора.
10. Как выполняется адресация данных?
11. Что такое адресное пространство?
12. Что такое кэш-память?
13. Перечислите способы размещения блоков основной памяти в кэше.
14. Что такое дисковые массивы?
15. Какие уровни RAID существуют?
16. С помощью какой утилиты выполняется тестирование RAID-массивов?

## Раздел 4. Вычислительные системы

### 4.1 Основные определения. Классы архитектур вычислительных систем

#### Основные определения

**Система** (от греческого *systema* — целое, составленное из частей соединение) — это совокупность элементов, взаимодействующих друг с другом, образующих определенную целостность, единство.

**Вычислительная система** — это совокупность одного или нескольких компьютеров или процессоров, программного обеспечения и периферийного оборудования, организованная для совместного выполнения информационно-вычислительных процессов.

**Вычислительная машина, счётная машина** — механизм, электро-механическое или электронное устройство, предназначенное для автоматического выполнения математических операций.

**Электронная вычислительная машина (ЭВМ), компьютер** — комплекс технических средств, предназначенных для автоматической обра-

ботки информации в процессе решения вычислительных и информационных задач.

### **Отличительные особенности вычислительных систем**

Из курса информатики вы знаете, что первые компьютеры (автоматические электронные вычислительные машины с программным управлением) были созданы в конце 40-х годов XX века и представляли собой гигантские вычислительные монстры, использовавшиеся только для вычислительной обработки информации. По мере развития компьютеры существенно уменьшились в размерах, но обросли дополнительным оборудованием, необходимым для их эффективного использования. В 70-х годах компьютеры из вычислительных машин сначала превратились в *вычислительные системы*, а затем в *информационно-вычислительные системы*.

Основным направлением совершенствования ЭВМ является неуклонный рост производительности (быстродействия) и интеллектуальности вычислительных средств.

Наиболее перспективным и динамичным направлением увеличения скорости решения прикладных задач является широкое внедрение идей параллелизма в работу вычислительных систем (ВС).

Дальнейшее поступательное развитие вычислительной техники напрямую связано с переходом к параллельным вычислениям, с идеями построения многопроцессорных систем и сетей, объединяющих большое количество отдельных процессоров и ЭВМ.

Создание ВС преследует следующие основные цели: повышение производительности системы за счет ускорения процессов обработки данных, повышение надежности и достоверности вычислений, предоставление пользователям дополнительных сервисных услуг и т.д.

Отличительной особенностью ВС по отношению к ЭВМ является наличие в них нескольких вычислителей, реализующих параллельную обработку.

***Основные принципы построения, закладываемые при создании ВС:***

- возможность работы в разных режимах;
- модульность структуры технических и программных средств, что позволяет совершенствовать и модернизировать вычислительные системы без коренных их переделок;
- унификация и стандартизация технических и программных решений;
- иерархия в организации управления процессами;
- способность систем к адаптации, самонастройке и самоорганизации;
- обеспечение необходимым сервисом пользователей при выполнении вычислений.



## **Классификация вычислительных систем**

### ***По назначению ВС делят на:***

- универсальные,
- проблемно-ориентированные
- специализированные.

*Универсальные* предназначаются для решения широкого класса задач. *Проблемно-ориентированные* используются для решения определенного круга задач в сравнительно узкой сфере. *Специализированные* ориентированы на решение узкого класса задач.

### ***По типу ВС делят на:***

- многомашинные;
- многопроцессорные.

Вычислительная система может строиться на базе либо целых компьютеров (многомашинная ВС), либо на базе отдельных процессоров (многопроцессорная ВС).

### ***По типу ЭВМ или процессоров различают:***

- однородные ВС – строятся на базе однотипных компьютеров или процессоров.

- неоднородные ВС – включает в свой состав различные типы компьютеров или процессоров.

### ***Территориально ВС делятся на:***

- сосредоточенные (все компоненты располагаются в непосредственной близости друг от друга);
- распределенные (компоненты могут располагаться на значительном расстоянии, например, вычислительные сети);

### ***По методам управления элементами ВС различают:***

- централизованные;
- децентрализованные;
- со смешанным управлением.

В первом случае управление выполняет выделенный компьютер или процессор, во втором – эти компоненты равноправны и могут брать управления на себя.

В системах со смешанным управлением совмещаются процедуры централизованного и децентрализованного управления. Перераспределение функций осуществляется в ходе вычислительного процесса исходя из сложившейся ситуации.

***По принципу закрепления вычислительных функций*** за отдельными ЭВМ (процессорами) различают системы с жестким и плавающим закреплением функций.

### ***По режиму работы ВС различают системы, работающие в***

- оперативном;
- неоперативном временных режимах.

*Оперативные* системы работают в реальном масштабе времени, в них реализуется оперативный режим обмена информацией – ответы на запросы поступают незамедлительно. В *неоперативных* ВС допускается режим “задержанного ответа”, когда результаты выполнения запроса можно получить с некоторой задержкой (иногда даже в следующем сеансе работы системы).

***Кроме этого, ВС могут быть структурно***

- одноуровневыми (имеется лишь один общий уровень обработки данных);

- многоуровневыми (иерархическими) структурами. В иерархических ВС машины или процессоры распределены по разным уровням обработки информации, некоторые машины (процессоры) могут специализироваться на выполнении определенных функций.

***Структура вычислительной системы***

*Структура ВС* - это совокупность комплексированных элементов и их связей. В качестве элементов ВС выступают отдельные ЭВМ и процессоры.

***Классы архитектур вычислительных систем***

***Архитектура системы*** – совокупность свойств системы, существенных для пользования.

***Архитектурой компьютера*** называется его описание на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т.д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного ЗУ, внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Наиболее распространены следующие архитектурные решения.

***Классическая архитектура*** (архитектура фон Неймана) — одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд — программа. Это *однопроцессорный компьютер*. К этому типу архитектуры относится и архитектура персонального компьютера с *общей шиной*. Все функциональные блоки здесь связаны между собой общей шиной, называемой также *системной магистралью*.

***Многопроцессорная архитектура***. Наличие в компьютере нескольких процессоров означает, что *параллельно может быть организовано много потоков данных и много потоков команд*. Таким образом, параллельно могут выполняться несколько фрагментов одной задачи. Многопроцессорная архитектура показана на рисунке 9.

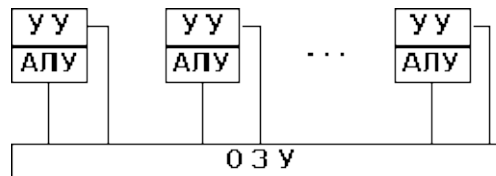


Рисунок 9 – Многопроцессорная архитектура

**Многомашинная вычислительная система.** Здесь несколько процессоров, входящих в вычислительную систему, не имеют общей оперативной памяти, а имеют каждый свою (локальную). Каждый компьютер в многомашинной системе имеет классическую архитектуру, и такая система применяется достаточно широко. Однако эффект от применения такой вычислительной системы может быть получен только при решении задач, имеющих очень специальную структуру: она должна разбиваться на столько слабо связанных подзадач, сколько компьютеров в системе.

Преимущество в быстродействии многопроцессорных и многомашинных вычислительных систем перед однопроцессорными очевидно.

**Архитектура с параллельными процессорами.** Здесь несколько АЛУ работают под управлением одного УУ. Это означает, что множество данных может обрабатываться по одной программе — то есть по одному потоку команд. Высокое быстродействие такой архитектуры можно получить только на задачах, в которых одинаковые вычислительные операции выполняются одновременно на различных однотипных наборах данных. Архитектура с параллельными процессорами показана на рисунке 10.

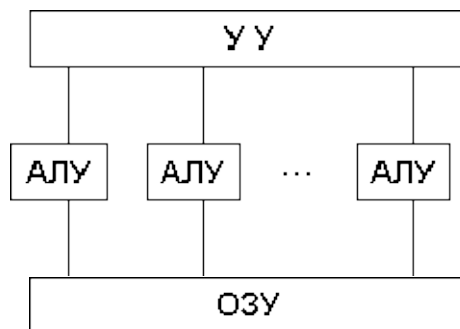


Рисунок 10 – Архитектура с параллельными процессорами

Самой ранней и наиболее известной является классификация архитектур вычислительных систем, предложенная в 1966 году Майклом Флинном. Классификация базируется на понятии потока, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором. На основе числа потоков команд и потоков данных Флинн выделяет четыре класса архитектур:

- **ОКОД** — Вычислительная система с одиночным потоком команд и одиночным потоком данных (**SISD**, Single Instruction stream over a Single

Data stream). К этому классу относятся, прежде всего, классические последовательные машины, или иначе, машины фон-неймановского типа, например, PDP-11 или VAX 11/780. В таких машинах есть только один поток команд, все команды обрабатываются последовательно друг за другом и каждая команда инициирует одну операцию с одним потоком данных. Не имеет значения тот факт, что для увеличения скорости обработки команд и скорости выполнения арифметических операций может применяться конвейерная обработка - как машина CDC 6600 со скалярными функциональными устройствами, так и CDC 7600 с конвейерными попадают в этот класс.

- **ОКМД** — Вычислительная система с одиночным потоком команд и множественным потоком данных (**SIMD**, Single Instruction, Multiple Data). В архитектурах подобного рода сохраняется один поток команд, включающий, в отличие от предыдущего класса, векторные команды. Это позволяет выполнять одну арифметическую операцию сразу над многими данными - элементами вектора. Способ выполнения векторных операций не оговаривается, поэтому обработка элементов вектора может производиться либо процессорной матрицей, как в ILLIAC IV, либо с помощью конвейера, как, например, в машине CRAY-1.

- **МКОД** — Вычислительная система со множественным потоком команд и одиночным потоком данных (**MISD**, Multiple Instruction Single Data). Определение подразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных. Однако ни Флинн, ни другие специалисты в области архитектуры компьютеров до сих пор не смогли представить убедительный пример реально существующей вычислительной системы, построенной на данном принципе. Ряд исследователей относят конвейерные машины к данному классу, однако это не нашло окончательного признания в научном сообществе. Будем считать, что пока данный класс пуст.

- **МКМД** — Вычислительная система со множественным потоком команд и множественным потоком данных (**MIMD**, Multiple Instruction Multiple Data). Этот класс предполагает, что в вычислительной системе есть несколько устройств обработки команд, объединенных в единый комплекс и работающих каждое со своим потоком команд и данных.

## 4.2 Примеры некоторых архитектур вычислительных систем

Рассмотрим далее примеры конкретных архитектур, а именно:

- симметричную многопроцессорную;
- асимметричную многопроцессорную (мультипроцессорную);
- массивно-параллельную;

- гибридную с неоднородным доступом к памяти;
- параллельную с векторными процессорами;
- кластерную.

*Симметричная мультипроцессорная обработка Symmetric Multiprocessing (SMP)*. SMP — архитектура суперкомпьютера, в которой группа процессоров работает с общей оперативной памятью (рис. 11).



Рисунок 11 – Симметричная мультипроцессорная архитектура

Память является способом передачи сообщений между процессорами, при этом все вычислительные устройства при обращении к ней имеют равные права и одну и ту же адресацию для всех ячеек памяти.

Работой управляет единственная копия операционной системы. Для ускорения обработки каждый процессор может также иметь собственную кэш-память. Задания между процессами распределяются непосредственно при выполнении прикладного процесса. Нагрузка между процессорами динамически выравнивается, а обмен данными между ними происходит с большой скоростью. *Достоинство* этого подхода состоит в том, что каждый процессор видит всю решаемую задачу в целом. Но поскольку для взаимодействия используется лишь одна шина, то возникают повышенные требования к ее пропускной способности. Соединение посредством шины применяется при небольшом (4—8) числе процессоров.

В подобных системах возникает проблема организации *когерентности многоуровневой иерархической памяти*.

Когерентность кэшей означает, что все процессоры получают одинаковые значения одних и тех же переменных в любой момент времени. Поскольку кэш-память принадлежит отдельному компьютеру, а не всей многопроцессорной системе в целом, данные, попадающие в кэш одного компьютера, могут быть недоступны другому. Чтобы избежать этого, следует провести синхронизацию информации, хранящейся в кэш-памяти процессоров.

*Асимметричная мультипроцессорная обработка ASymmetric Multiprocessing (ASMP)* — архитектура суперкомпьютера, в которой каждый процессор имеет свою оперативную память.

В ASMP используются три схемы (рис. 12). В любом случае процессоры взаимодействуют между собой, передавая друг другу сообщения, т. е. как бы образуя скоростную локальную сеть. Передача сообщений может осуществляться через общую шину (рис. 12, а, см. также *MPP-архитектуру*) либо благодаря межпроцессорным связям. В последнем случае процессоры связаны либо непосредственно (рис. 12, б), либо через друга друга (рис. 12, в). Непосредственные связи используются при небольшом числе процессоров.

Каждый процессор имеет свою, расположенную рядом оперативную память. Благодаря этому, если это необходимо, процессоры могут располагаться в различных, но рядом установленных корпусах. Группа устройств в одном корпусе именуется кластером. Пользователь, обращаясь к кластеру, может работать сразу с группой процессоров. Такое объединение увеличивает скорость обработки данных и расширяет используемую оперативную память. Резко возрастает также отказоустойчивость, ибо кластеры осуществляют резервное дублирование данных. Созданная таким образом система называется кластерной. В этой системе, в соответствии с ее структурой, может функционировать несколько копий операционной системы и несколько копий прикладной программы, которые работают с одной и той же базой данных (БД), решая одновременно разные задачи.

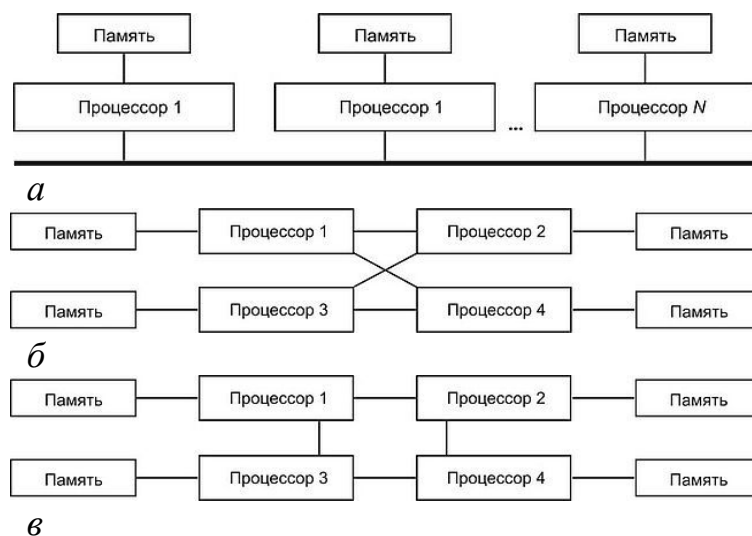


Рисунок 12 – Схемы асимметричной многопроцессорной архитектуры

*MPP-архитектура* (*massive parallel processing*) — массивно-параллельная архитектура (см. рис. 12, а). В этом случае система строится из отдельных модулей, каждый из которых содержит:

- процессор;
- локальный банк оперативной памяти (ОП);

– два коммуникационных процессора (маршрутизатора, рутера — router): один — для передачи команд, другой — для передачи данных (или сетевой адаптер);

– жесткие диски и/или другие устройства ввода-вывода.

По своей сути, такие модули представляют собой полнофункциональные компьютеры. Доступ к банку ОП из данного модуля имеют только процессоры из этого же модуля. Модули соединяются специальными коммуникационными каналами. Пользователь может определить логический номер процессора, к которому он подключен, и организовать обмен сообщениями с другими процессорами.

*Гибридная архитектура (NUMA). Организация когерентности многоуровневой иерархической памяти.* Главная особенность гибридной архитектуры NUMA (*nonuniform memory access*) — неоднородный доступ к памяти.

Гибридная архитектура воплощает в себе удобства систем с общей памятью и относительную дешевизну систем с отдельной памятью. Суть этой архитектуры — в методе организации памяти, а именно: память является физически распределенной по различным частям системы, но логически разделяемой, так что пользователь видит единое адресное пространство. Система состоит из однородных базовых модулей (плат), состоящих из небольшого числа процессоров, и блока памяти. Модули объединены с помощью высокоскоростного коммутатора. Поддерживается единое адресное пространство, аппаратно поддерживается доступ к удаленной памяти, т. е. к памяти других модулей. При этом доступ к локальной памяти осуществляется в несколько раз быстрее, чем к удаленной. По существу архитектура NUMA является MPP (массивно-параллельной) архитектурой, где в качестве отдельных вычислительных элементов берутся SMP-узлы.

Пример структурной схемы компьютера с гибридной сетью (рис. 13): четыре процессора связываются между собой с помощью кроссбара в рамках одного SMP-узла. Узлы связаны сетью типа «бабочка» (Butterfly).

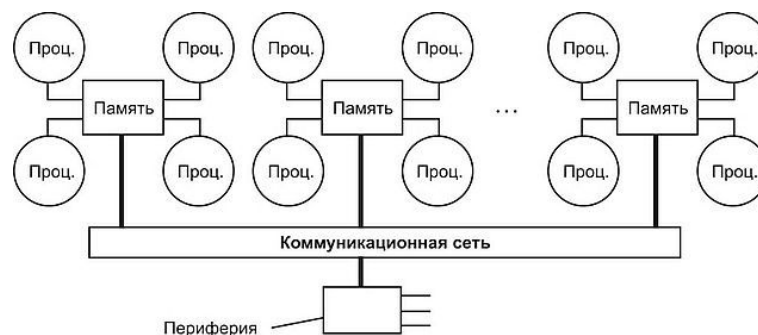


Рисунок 13 – Гибридная архитектура

*PVP-архитектура.* PVP (Parallel Vector Process) — параллельная архитектура с векторными процессорами. Основным признаком PVP-систем является наличие *векторно-конвейерных процессоров*, в которых предусмотрены команды однотипной обработки векторов независимых данных, эффективно выполняющиеся на конвейерных функциональных устройствах. Как правило, несколько таких процессоров (1 — 16) работают одновременно с общей памятью (аналогично SMP) в рамках многопроцессорных конфигураций. Несколько таких узлов могут быть объединены с помощью коммутатора (аналогично MPP). Поскольку передача данных в векторном формате осуществляется намного быстрее, чем в скалярном, то проблема взаимодействия между потоками данных при распараллеливании становится несущественной. И то, что плохо распараллеливается на скалярных машинах, хорошо распараллеливается на векторных. Таким образом, системы PVP-архитектуры могут являться машинами общего назначения. Однако, поскольку векторные процессоры весьма дороги, эти машины не являются общедоступными.

*Кластерная архитектура.* Кластер представляет собой два или более компьютеров (часто называемых узлами), которые объединяются с помощью сетевых технологий на базе шинной архитектуры или коммутатора и предоставляются пользователю в качестве единого информационно-вычислительного ресурса. В качестве узлов кластера могут выступать серверы, рабочие станции или обычные персональные компьютеры. Преимущество кластеризации для повышения работоспособности становится очевидным в случае сбоя какого-либо узла; при этом другой узел кластера может взять на себя нагрузку неисправного узла, и пользователи не заметят прерывания в доступе. Возможности масштабируемости кластеров позволяют многократно увеличивать производительность приложений для большего числа пользователей технологий на базе шинной архитектуры или коммутатора. Такие суперкомпьютерные системы являются самыми дешевыми, поскольку собираются на базе стандартных комплектующих элементов: процессоров, коммутаторов, дисководов и внешних устройств.

### **4.3 Обобщенные представления об архитектуре вычислительных машин и систем**

Рассматривая архитектуру ЭВМ, вычислительных систем, суперкомпьютеров и информационно-вычислительных сетей с общих позиций и абстрагируясь от деталей, можно воспользоваться схемой, показанной на рисунке 14.





Рисунок 14 – Абстрактное представление об архитектурах

Здесь используется принципы классификации Скилликорна, построенной на следующих элементах-объектах:

- *процессор команд* (IP — Instruction Processor) — функциональное устройство, работающее как интерпретатор команд; в системе, вообще говоря, может отсутствовать;

- *процессор данных* (DP — Data Processor) — функциональное устройство, работающее как преобразователь данных, в соответствии с арифметическими операциями;

- *иерархия памяти* (IM — Instruction Memory, DM — Data Memory) — запоминающее устройство, в котором хранятся данные и команды, пересылаемые между процессорами;

- *переключатель* — абстрактное устройство, обеспечивающее связь между процессорами и памятью.

Таким образом, имеются:

- *процессоры и блоки памяти* — информационно-вычислительная среда;

- *средства коммутации и коммуникации* — коммуникационно-коммутационная среда.

Все эти компоненты присутствуют как в ЭВМ, так и в вычислительных сетях и системах (суперЭВМ).

### ***Требования к архитектурным компонентам МВС (многопроцессорных вычислительных систем)***

Эти аспекты имеют более широкий смысл, чем просто требования к техническим характеристикам компонент вычислительной системы: процессору, дисковым массивам, памяти, коммутаторам и т.п. аппаратным средствам. Гораздо более важное значение имеют требования, предъявля-

емые к вычислительной системе, которую собираются построить для реализации конкретных целей — решения задач определённого круга (научных, экономических, информационных систем и т.п.), модель программирования.

Разработчикам необходимо, прежде всего, проанализировать следующие связанные между собой вопросы:

- Отношение стоимость/производительность.
- Надежность и отказоустойчивость.
- Масштабируемость.
- Совместимость программного обеспечения.
- Отношение стоимость/производительность.

Добиться дополнительного повышения производительности в МВС тяжелее, чем произвести масштабирование внутри узла. Основным барьером является трудность организации эффективных межузловых связей. Коммуникации, которые происходят между узлами, должны быть устойчивы к большим задержкам программно поддерживаемой когерентности. *(Вспомним, когерентность означает, что все процессоры получают одинаковые значения одних и тех же переменных в любой момент времени.)* Приложения с большим количеством взаимодействующих процессов работают лучше на основе SMP-узлов, в которых коммуникационные связи более быстрые. В кластерах, как и в MPP системах, масштабирование приложений более эффективно при уменьшении объема коммуникаций между процессами, работающими в разных узлах. Это обычно достигается путем разбиения данных.

Именно такой подход используется в наиболее известном приложении на основе кластеров OPS (Oracle Parallel Server).

Появление любого нового направления в вычислительной технике определяется требованиями компьютерного рынка. Поэтому у разработчиков компьютеров нет одной единственной цели. Большая универсальная вычислительная машина (мейнфрейм) или суперкомпьютер стоят дорого. Для достижения поставленных целей при проектировании высокопроизводительных конструкций приходится игнорировать стоимостные характеристики.

Суперкомпьютеры фирмы Cray Research и высокопроизводительные мейнфреймы компании IBM относятся именно к этой категории компьютеров. Другим крайним примером может служить конструкция, где производительность принесена в жертву для достижения низкой стоимости. К этому направлению относятся персональные компьютеры различных клонов IBM PC. Между этими двумя крайними направлениями находятся конструкции, основанные на отношении стоимость/производительность, в которых разработчики находят баланс между стоимостными параметрами и

производительностью. Типичными примерами такого рода компьютеров являются миникомпьютеры и рабочие станции.

Для сравнения различных компьютеров между собой обычно используются стандартные методики измерения производительности. Эти методики позволяют разработчикам и пользователям использовать полученные в результате испытаний количественные показатели для оценки тех или иных технических решений, и, в конце концов, именно производительность и стоимость дают пользователю рациональную основу для решения вопроса, какой компьютер выбрать.

#### 4.4 Перспективные типы процессоров ЭВМ

**Ассоциативные процессоры.** Существующие в настоящее время процессоры ориентированы на адресную обработку данных. Ассоциативный способ обработки предполагает доступ к информации не по адресу, а по признаку (критерию отбора). Все преобразования производятся только над теми данными, которые этому критерию удовлетворяют. Критерием отбора может быть совпадение с любым элементом данных. Ассоциативный процессор относится к классу ОКМД (вычислительная система с односторонним потоком команд и множественным потоком данных) и включает несколько операционных устройств (АЛУ), способных одновременно по командам УУ вести обработку нескольких потоков данных. Информация из ассоциативной памяти поступает на схему сравнения и сравнивается с маской (признаком), т.е. не по адресу, а по содержанию.

**Матричные процессоры.** Это процессоры, являющиеся представителем класса ОКМД. Они имеют общее УУ, которое генерирует поток команд, и большое число процессорных элементов, работающих параллельно и обрабатывающих каждый свой поток данных. Производительность системы равна сумме производительностей всех процессорных элементов. Одним из первых матричных процессоров был SOLOMON, содержащий 1024 ПЭ, которые соединены в матрицу 32\*32. Каждый элемент матрицы включает процессор и ЗУ емкостью 16 Кбайт. Длина слова – переменная от 1 до 128 разрядов и устанавливается программно. По каналам связи от УУ передаются команды и общие константы. Дальнейшим развитием матричных систем явилась система ILIAS-4, состоящая из 256 ПЭ. Реальная производительность 200 миллионов операций в секунду. В России создана матричная система ПС-2000, состоящая из 64 ПЭ, обеспечивающих быстрое действие 200 млн. оп/сек. Структура матричной вычислительной системы "SOLOMON" показана на рисунке 15.



Рисунок 15 – Структура матричной вычислительной системы "SOLOMON"

**ДНК процессоры.** В последнее время все чаще используют биотехнологии, соединяя информационные технологии с молекулярными. Это позволяет решать сложные вычислительные задачи, пользуясь методами биологии, организуя вычисления с помощью живых тканей, клеток, вирусов и биомолекул. Наибольшее распространение получил подход, где в качестве основного элемента (процессора) используются молекулы ДНК (дезоксирибонуклеиновой кислоты). Центральное место в этом подходе занимает ДНК-процессор.

Он характеризуется структурой и набором команд. Структура процессора – это структура молекулы ДНК, а набор команд – это перечень биохимических операций над молекулами. Принцип устройства компьютерной ДНК-памяти основан на последовательном соединении 4 нуклеотидов (из которых состоит молекула ДНК). 3 нуклеотида, соединяясь в любой последовательности, образуют элементарную ячейку памяти – **кодон**. Кодоны затем формируют цепь ДНК. Основная трудность – проведение избирательных однокодонных реакций (взаимодействий) внутри цепи ДНК. Другая проблема – самосборка ДНК, приводящая к потере информации. Поэтому в клетку вводят специальные вещества, препятствующие самосборке. *Уже существует экспериментальное оборудование, позволяющее работать с таким процессором. В 2001 удалось реализовать вычислительное устройство на основе ДНК, которое может работать без вмешательства человека. Система имитирует машину, которая теоретически может решить любую задачу.* Молекулы ДНК по своей природе работают распадаясь и собираясь в соответствии с информацией, закодированной в цепочках химических соединений. Взаимодействуют молекулы под воздействием специальных ферментов, один из которых расщепляет

молекулу ДНК, а другой собирает ее. Поэтому фермент выполняет роль аппаратного, а молекулы ДНК – программного обеспечения.

**Клеточные процессоры.** Клеточные компьютеры представляют собой самоорганизующиеся колонии различных микроорганизмов, в геном которых удалось включить логическую схему, которая начинает работать в присутствии определенного вещества. Компьютер будет представлять собой стакан с бактериями. Каждая клетка представляет собой минилабораторию. Если микроорганизм запрограммирован, он производит нужные действия.

Основная проблема – организация всех клеток в единую рабочую систему. В настоящее время создана клетка, способная хранить на генетическом уровне 1 бит. Известно, что гены могут синтезировать белок под воздействием определенных раздражителей. Ученые изменили генетический код бактерий так, что их клетки могут выполнять логические операции. Например, для операции И в клетку подаются два вещества (операнды), под воздействием которых ген вырабатывает определенный белок.

**Достоинства** биокомпьютеров:

- простая технология изготовления;
- использование троичного кода, т.к. информация кодируется тройками нуклеотидов;
- высокая производительность;
- возможность хранить данные с плотностью, превышающей оптические диски;
- исключительно низкое энергопотребление.

**Недостатки:** - сложность считывания результатов, низкая точность вычислений, невозможность длительного хранения.

**Коммуникационные процессоры** – это микрочипы, представляющие нечто среднее между жесткими специализированными микросхемами и гибкими процессорами общего назначения. КП программируются, как и обычные процессоры, но построены с учетом сетевых задач. Эти процессоры имеют собственную память и оснащены высокоскоростными внешними каналами для соединения с другими процессорами. Присутствие такого процессора освобождает вычислительный процессор от операций обмена. Скоростной КП с RISC ядром позволяет управлять обменом данными по нескольким независимым каналам, поддерживает практически все распространенные протоколы обмена. КП позволяют эффективно работать в системах управления, которые являются распределенными и требуют разветвленных сетей.

**Процессоры баз данных** - это программно-управляемые комплексы, предназначенные для выполнения функций систем управления базами данных (СУБД). Если раньше базы данных хранили в основном числовую и текстовую информацию, то теперь они рассчитаны на различные виды

данных. Эти процессоры выполняют функции управления и распространения, обеспечивают дистанционный доступ к информации и тиражирование.

**Потоковые процессоры** - это процессоры, в основе которых лежит принцип обработки многих данных с помощью одной команды (МКОД). Эта технология позволяет выполнять одно и то же действие над несколькими наборами чисел одновременно, ускоряет обработку видео, речи, шифрование, обработку изображений.

Существуют однопоточковые процессоры SSP и многопоточковые MSP.

Представителем потоковых процессоров является семейство процессоров, начиная с Pentium 3, в основе которых лежит потоковая обработка по принципу «одна команда – много данных». Представителями многопоточковых процессоров являются матрицы процессоров ILLIAC-4, в которых единое управляющее устройство контролирует множество процессорных элементов.

**Нейронные процессоры** - это процессоры на основе обработки информации в искусственных нейронных сетях (НС). Нейронная сеть – это совокупность элементарных преобразователей информации, называемых нейронами, которые соединены друг с другом каналами обмена информации синаптическими связями. Основу нейровычислителя составляют простые однотипные элементы, имитирующие работу нейронов мозга. Каждый нейрон характеризуется своим текущим состоянием по аналогии с клетками мозга (торможение или возбуждение).

**Процессоры с нечеткой логикой.** Идея их работы основана на нечеткой математике, т.е. когда входные данные непрерывно меняются во времени и их невозможно задать однозначно (как в теории вероятности). Нечеткая логика не оперирует 0 и 1, а имеет дело со значениями, лежащими в некотором интервале. Задачи решаются следующим образом: численные данные переводятся в нечеткий формат, обрабатываются по определенным правилам и обратно переводятся в привычный формат и подаются на выход. Все это осуществляется без ЗУ и программного обеспечения.

#### 4.5 Коммуникационные среды

В самом общем смысле архитектуру компьютера можно определить как способ соединения компьютеров между собой, с памятью и с внешними устройствами. Реализация этого соединения может идти различными путями. Конкретная реализация такого рода соединений называется **коммуникационной средой** компьютера. Одна из самых простых реализаций – это использование *общей шины*, к которой подключаются как процессоры,

так и память. Сама шина состоит из определенного числа линий связи, необходимых для передачи адресов, данных и управляющих сигналов между процессором и памятью. Этот способ реализован в SMP-системах.

Основным недостатком таких систем, является плохая *масштабируемость*. Увеличение, даже незначительное, числа устройств на шине вызывает заметные задержки при обмене с памятью и катастрофическое падение производительности системы в целом. Необходимы другие подходы для построения коммуникационной среды, и одним из них является разделение памяти на независимые модули и обеспечение возможности доступа разных процессоров к различным модулям одновременно посредством использования различного рода коммутаторов.

При этом возможны различные конфигурации получающихся систем связи. Так, в компьютерах семейства Cray T3D/T3E все процессоры были объединены специальными *высокоскоростными каналами* в трехмерный тор, в котором каждый вычислительный узел имел непосредственные связи с шестью соседями. В компьютерах *IBM SP/2* взаимодействие процессоров происходит через иерархическую систему коммутаторов, также обеспечивающую возможность соединения каждого процессора с любым другим. Эти оригинальные уникальные решения значительно увеличивают цену компьютеров.

Гораздо более простым и дешевым оказалось использование связей на базе *семей Ethernet* – методика, разработанная компанией Xerox. Первоначально использовалась обычная 10-мегабитная сеть, затем стали применять *Fast Ethernet*, а в позже и *Gigabit Ethernet*. Но для *Fast Ethernet* характерна большая *латентность* (задержка в передаче данных), оцениваемая в 160-180 микросекунд, а *Gigabit Ethernet* отличается высокой стоимостью. Кроме того, он эффективен только при соединении точка–точка, при соединении нескольких узлов его эффективность резко падает, а при соединении более 5–6 узлов она не превосходит по производительности даже *Fast Ethernet*. Поэтому при создании многопроцессорных вычислительных систем часто предпочтение отдается технологиям *SCI*, *Myrinet* или *Raceway*.

### **Примеры построения коммуникационных сред на основе масштабируемого когерентного интерфейса SCI**

*SCI* (Scalable Coherent Interface) принят как стандарт в 1992 г. Он предназначен для достижения высоких скоростей передачи с малым *временем задержки* и при этом обеспечивает масштабируемую архитектуру, позволяющую строить системы, состоящие из множества блоков. *SCI* представляет собой комбинацию шины и локальной сети, обеспечивает реализацию когерентности кэш-памяти, размещаемой в узле *SCI*, посредством механизма распределенных директорий, который улучшает произ-

водительность, скрывая затраты на доступ к удаленным данным в модели с распределенной разделяемой памятью. Производительность передачи данных обычно находится в пределах от 200 Мбайт/с до 1000 Мбайт/с на расстояниях десятков метров с использованием электрических кабелей и километров – с использованием оптоволоконна. *SCI* уменьшает время межузловых коммуникаций по сравнению с традиционными схемами передачи данных в сетях путем устранения обращений к программным уровням – операционной системе и библиотекам времени выполнения. Коммуникации представляются как часть простой операции загрузки данных процессором. Обычно обращение к данным, физически расположенным в памяти другого вычислительного узла и не находящимся в кэше, приводит к формированию запроса к удаленному узлу для получения необходимых данных, которые в течение нескольких микросекунд доставляются в локальный кэш, и выполнение программы продолжается. Прежний подход требовал формирования пакетов на программном уровне с последующей передачей их аппаратному обеспечению. Точно так же происходил и прием, в результате чего задержки были в сотни раз больше, чем у *SCI*. Однако для совместимости *SCI* имеет возможность переносить пакеты других протоколов.

Еще одно преимущество *SCI* – использование простых протоколов типа *RISC*, которые обеспечивают большую пропускную способность. Узлы с адаптерами *SCI* могут использовать для соединения коммутаторы или же соединяться в кольцо. Обычно каждый узел оказывается включенным в два кольца (рис. 16).

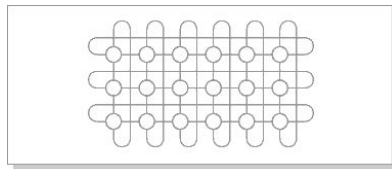


Рисунок 16 – Матрица узлов кластера на основе сети *SCI*

Данная технология оптимизирована для работы с динамическим трафиком, однако может быть менее эффективна при работе с большими блоками данных. Протокол передачи данных обеспечивает гарантированную доставку и отсутствие дедлоков (Deadlock — ситуация в многозадачной среде или СУБД, при которой несколько процессов находятся в состоянии бесконечного ожидания ресурсов, захваченных самими этими процессами.).

Протокол *SCI* достаточно сложен, он предусматривает широкие возможности управления трафиком, но использование этих возможностей предполагает наличие развитого программного обеспечения.



Традиционная область применения *SCI* – это **коммуникационные среды** многопроцессорных систем.

Модульные *SCI*-коммутаторы Dolphin позволяют потребителям строить масштабируемые кластерные решения класса предприятия на платформах *Windows NT/2000/XP*, *Linux*, *Solaris*, *VxWorks*, *LynuxWorks* и *NetWare* с использованием стандартизированного оборудования и программного обеспечения. Характеристики технологии *SCI* приведены в таблице 4.

Таблица 4 – Технология *SCI*

<b>Производители оборудования</b>	Dolphin Interconnect Solutions и др.
<b>Показатели производительности</b>	Для продуктов Dolphin: пиковая пропускная способность – 667 Мбит/с. Аппаратная латентность – 1.46 мксек
<b>Программная поддержка</b>	Драйверы для <i>Linux</i> , <i>Windows NT</i> , <i>Solaris</i> . ScaMPI – реализация MPI компании Scali Computer для систем на базе <i>SCI</i> . SISCO API – интерфейс программирования нижнего уровня
<b>Комментарии</b>	<i>SCI</i> (ANSI/IEEE 1596-1992) – стандартизированная технология. Кроме стандартной сетевой среды, <i>SCI</i> поддерживает построение систем с разделяемой памятью и с когерентностью кэшей. На коммуникационной технологии <i>SCI</i> основаны кластерные системы компании SCALI Computer, системы семейства hpcLine компании Siemens, а также cc-NUMA-сервера Data General и Sequent. Технология <i>SCI</i> использовалась для связи гиперузлов в системах HP/Convex Exemplar X-class.

### **Коммуникационная среда MYRINET**

Сетевую технологию Myrinet представляет компания Myricom, которая впервые предложила свою коммуникационную технологию в 1994 году, а в настоящее время довольно широко распространена по всему миру. Технология Myrinet основана на использовании *многопортовых коммутаторов* при ограниченных несколькими метрами длинах связей узлов с портами коммутатора. Узлы в Myrinet соединяются друг с другом через коммутатор (до 128 портов). Максимальная длина линий связи варьируется в зависимости от конкретной реализации.

Как коммутируемая сеть, аналогичная по структуре сегментам Ethernet, соединенным с помощью коммутаторов, Myrinet может одновременно передавать несколько пакетов, каждый из которых идет со скоро-

стью, близкой к 2 Гбит/с. В отличие от некоммутированных Ethernet и FDDI сетей, которые разделяют общую среду передачи, совокупная пропускная способность сети Myrinet возрастает с увеличением количества машин. На сегодня Myrinet чаще всего используют как локальную сеть (LAN) сравнительно небольшого размера, связывая вместе компьютеры внутри комнаты или здания. Из-за своей высокой скорости, малого времени задержки, прямой коммутации и умеренной стоимости Myrinet часто используется для объединения компьютеров в кластеры. Myrinet также используется как системная сеть (System Area Network, SAN), которая может объединять компьютеры в кластер внутри стойки с той же производительностью, но с более низкой стоимостью, чем Myrinet LAN. Пакеты Myrinet могут иметь любую длину. Таким образом, они могут включать в себя другие типы пакетов, в том числе IP-пакеты. Объединение вычислительных узлов с адаптерами Myrinet в сеть происходит с помощью коммутаторов, которые имеют сейчас 4, 8, 12 или 16 портов. В коммутаторах используется передача пакетов путем установления соединения на время передачи, для маршрутизации сообщений применяется алгоритм прокладки пути (wormhole, "червоточина"). Коммутаторы, как и сетевые адаптеры, построены на специализированных микропроцессорах LANai компании Myricom. Характеристики технологии Myrinet приведены в таблице 5.

Таблица 5 – Технология Myrinet

<b>Производители оборудования</b>	Myricom
<b>Показатели производительности</b>	Пиковая пропускная способность – 2 Гбит/с, полный дуплекс. Латентность – порядка 4 мксек.
<b>Программная поддержка</b>	Драйверы для Linux (Alpha, x86, PowerPC, UltraSPARC), Windows NT (x86), Solaris (x86, UltraSPARC) и Tru64 UNIX. GM – интерфейс программирования на нижнем уровне. Пакеты HPVM (включает MPI-FM, реализацию MPI для Myrinet), VIP-MPI и др.
<b>Комментарии</b>	Myrinet является <i>открытым стандартом</i> . Myricom предлагает широкий выбор сетевого оборудования по сравнительно невысоким ценам. На физическом уровне поддерживаются сетевые среды SAN (System Area Network), LAN (CL-2) и оптоволокно. Технология Myrinet предоставляет широкие возможности масштабирования сети и в настоящее время очень часто используется при построении высокопроизводительных кластеров

На физическом уровне линки (линии связи) Myrinet состоят из 9 проводников: 8 битов предназначены для передачи информации, интерпретируемой в зависимости от состояния девятого бита как байт данных или управляющий символ; при этом на каждом линке обеспечивается управление потоком и контроль ошибок. Среда Myrinet выгодно отличается от многих других сред передачи, в частности, SCI, простотой концепции и аппаратной реализации протоколов. Она содержит ограниченный набор средств управления трафиком, использующих приливно-отливный буфер, управляющие символы и таймерные интервалы. Myrinet является открытым стандартом, компания Myricom предлагает широкий выбор сетевого оборудования по сравнительно невысоким ценам. Технология Myrinet предоставляет широкие возможности масштабирования сети и часто используется при построении высокопроизводительных вычислительных кластеров.

#### **4.6 Кластерные и массивно-параллельные системы**

##### **Параллельные архитектуры**

Анализ сложности задач, возникающих при проектировании инженерных сооружений, автомобилей, судов и летальных аппаратов, военной техники, моделировании атмосферы и мирового океана, предсказании погоды и глобальных изменений в атмосфере, разработке новых лекарств, при проведении космических исследований, при создании систем визуализации, а также широкого класса других задач показал, что для их решения требуются компьютеры с производительностью  $10^{10} - 10^{16}$  оп/сек и более. Для повышения производительности компьютеров может увеличиваться быстродействие последовательных архитектур путем применения более современной технологии.

Однако последовательная архитектура имеет предел, определяемый скоростью распространения электрического сигнала по физическим линиям связи между компонентами. Альтернативой последовательным являются архитектуры, основанные на параллелизме, когда данные и алгоритмы распределяются между процессорами. Параллельные архитектуры являются наиболее перспективным и динамичным направлением увеличения скорости решения прикладных задач. К настоящему времени спроектированы и опробованы сотни различных компьютеров, использующих в своей архитектуре тот или иной вид параллельной обработки данных. Однако проверку временем выдержали только несколько из огромного множества параллельных архитектур. Вследствие отсутствия единой классификации параллельных архитектур, возможны разночтения при выделении основных классов и их количества. Приняв в качестве параметра классификации

наличие общей (разделяемой) или распределенной памяти можно выделить следующие классы параллельных компьютеров:

- Симметричные мультипроцессорные системы (SMP) - системы, в которых несколько процессоров работает над общей памятью; вся система функционирует под управление одной операционной системы (ОС).

- Массивно-параллельные системы (MPP) - системы с распределенной памятью, состоящие из вычислительных узлов, связанных через коммуникационную среду; на каждом узле работает полноценный или урезанный вариант ОС.

- Системы с неоднородным доступом к памяти (NUMA-системы) - системы, в которых память физически распределена, но логически общедоступна, то есть представляющие нечто среднее между SMP и MPP-системами.

- Параллельно-векторные системы (PVP) - системы со специальными векторно-конвейерными процессорами. В узле несколько таких процессоров, как правило, работают над общей памятью (как в SMP-системе). Несколько узлов могут быть объединены с помощью коммутаторов (как в MPP-системе).

### **Массивно-параллельные системы (MPP)**

Основные причины появления массивно-параллельных компьютеров - это, во-первых, необходимость построения компьютеров с гигантской производительностью, и, во-вторых, необходимость производства компьютеров в большом диапазоне как производительности, так и стоимости. Для массивно-параллельного компьютера, в котором число процессоров может сильно меняться, всегда можно подобрать конфигурацию с заранее заданной производительностью и/или стоимостью.

С некоторой степенью условности, MPP-компьютеры можно характеризовать следующими параметрами: используемые микропроцессоры, коммуникационная сеть, организация памяти и наличие или отсутствие host-компьютера.

MPP система состоит из однородных вычислительных узлов, включающих:

- один или несколько центральных процессоров (обычно RISC);
- локальную память;
- коммуникационный процессор или сетевой адаптер;
- устройства ввода/вывода.

В каждом узле работает своя копия ОС, а узлы объединяются коммуникационной сетью (сетью межпроцессорного взаимодействия). Поскольку как уже отмечалось, MPP системы относятся к системам с распределенной памятью, то число процессоров может быть очень большим.

Современное направление развития MPP-систем заключается в увеличении мощности вычислительного узла путем подсоединения дополнительных процессоров и по существу превращения его в SMP-узел, а так же путем увеличения вычислительной мощности применяемых процессоров.

### **Кластерные системы**

Вариантом реализации массивно-параллельных систем являются кластеры, представляющие связанный набор полноценных компьютеров, каждый из которых может обладать архитектурой SMP, NUMA.

Развитие сетевых технологий привело к появлению недорогих, но эффективных коммуникационных решений. Это и предопределило появление кластерных вычислительных систем, фактически являющихся одним из направлений развития компьютеров с массовым параллелизмом. Классические суперкомпьютеры, использующие специализированные процессоры таких фирм как, например, Cray, NEC (векторно-параллельные или массивно-параллельные), обычно недешевы, поэтому и стоимость подобных систем не сравнима со стоимостью систем, находящихся в массовом производстве.

Фактически кластеры представляют собой набор из нескольких ЭВМ, соединенных через некоторую коммуникационную структуру. В качестве такой структуры может выступать обычная компьютерная сеть (на основе сетевой технологии Ethernet, Token Ring), однако из соображений производительности желательно иметь высокоскоростные каналы на основе специализированных технологий (Memory Channel (свыше 100 Мб/с), Myrinet (140 Мб/с), SCI). Кластеры могут использоваться для построения высоконадежных систем с уровнем готовности 99,999% или для проведения суперкомпьютерных вычислений. В последнем случае на первый план выходит не высокая доступность системы, а быстродействие и способность эффективно распараллелить решаемую задачу. Высокая производительность, которая теоретически достижима при выполнении вычислений на множестве машин, может быть существенно снижена из-за задержек и низкого быстродействия межсоединений узлов кластера. Решение на базе высокоскоростного коммуникационного канала стоит недешево, однако по сравнению с традиционными коммерческими суперкомпьютерами, кластер на базе стандартных процессоров гораздо более экономичен.

### **Контрольные вопросы к Разделу 4**

1. Каковы основные принципы построения, закладываемые при создании ВС?
2. По каким критериям можно классифицировать ВС?

3. На какие группы по назначению делятся ВС?
4. Что такое архитектура вычислительной системы?
5. Перечислите наиболее распространенные архитектурные решения.
6. Классификация архитектур вычислительных систем Майкла Флинна.
7. Что такое SMP?
8. Что такое ASMP?
9. Что такое MPP?
10. Что такое NUMA?
11. Что такое PVP?
12. Ассоциативные процессоры и их принцип работы.
13. Матричные процессоры и их принцип работы.
14. Принцип работы ДНК процессоров.
15. Достоинства и недостатки клеточных процессоров?
16. Понятие и назначение коммуникационной среды.
17. Перечислите используемые параллельные архитектуры.
18. Особенности массивно-параллельных систем.
19. Назначение и особенности кластерных вычислительных систем.

## **Раздел 5. Персональные компьютеры**

### **5.1 Устройство ПК на процессорах Intel, AMD**

Лет десять назад выбор между AMD или Intel был почти бессмысленным, т.к. они были по своим параметрам очень похожими.

Но потом AMD, потеряв шансы, едва не выбыла из гонки за звание производителя лучших процессоров для настольных ПК.

С тех пор процессоры линейки Core i не знали конкуренции. Год за годом на рынке появлялись слегка модифицированные новые ЦП, хотя количество ядер процессоров от Intel не менялось: i3 — двухядерный, i5 — четырехядерный, i7 — четырехядерный с поддержкой Hyper Threading (Hyper Threading (гиперпоточность) — технология, разработанная компанией Intel для процессоров на микроархитектуре NetBurst.).

Разогнать за счет изменения множителя можно только модели с суффиксом «К» в названии — остальные чипы защищены от повышения тактовой частоты. А суффикс этот обойдется владельцу ПК в дополнительные 3000–5000 рублей.

Но в начале 2017 года ситуация коренным образом изменилась. AMD представила миру конкурентоспособные ЦП Ryzen серии 3/5/7, изменив технические правила игры. Процессоры Ryzen 7 работают на восьми ядрах и поддерживают гиперпоточность, Ryzen 5 обладают шестью ядрами, а

Ryzen 3 — четырьмя. На всех решениях от Ryzen снята защита от повышения тактовой частоты, причем на цене разблокировка множителя не отражается.

В ответ на такой ход Intel должна была что-то предпринять. И предприняла, выпустив высокопроизводительные процессоры Core новой серии X.

Перед теми, кто планирует собрать новый ПК, стоит непростой выбор: процессор от AMD или Intel. Именно от этого нужно отталкиваться при покупке остальных компонентов — материнской платы, памяти, охлаждения. Корпус, блок питания и жесткий диск можно приобрести любые, на ваш вкус.

### Бюджетная сборка или доступный офисный помощник

Требования к ПК для офисной работы, веб-серфинга и просмотра видео в сравнении с производительностью современных процессоров скромны. В худшем случае открытие очень больших документов будет требовать больше времени. Кроме того, при планировании следует учитывать ряд моментов: ПК должен занимать как можно меньше места и стоять на столе.

Материнская плата рекомендуется форм-фактора mATX. Бюджет для офисного ПК не должен быть слишком большим.

Раньше за AMD было преимущество, заключающееся в доступных чипах, однако с появлением на рынке новых центральных процессоров Pentium, выпущенных на микроархитектуре Kabu Lake, ситуация изменилась. Они работают, как обычно, на двух ядрах, но благодаря Hyper Threading могут обрабатывать сразу четыре потока, так что по производительности не уступают более дорогим Core i3.

На рисунке 17 показана разница в загрузке ядер процессора с использованием и без использования технологии Hyper Threading.

Рисунок

### Разница в загрузке ядер процессора



анием и без ис-

По встро-  
тактовая частот-  
довать мощные  
ном уровне со  
В таблиц-  
G4620 и AMD

Core i3, только  
но порекомен-  
ходится на од-  
з Intel Pentium

Таблица 6 – Сравнение характеристик процессоров Intel Pentium G4620 и AMD A10-6700

Сокет	Intel Socket 1151	AMD Socket FM2
Техпроцесс	14 nm	32 nm
Кол-во транзисторов	unknown	1178 million
Частота процессора	3700 MHz	3700 MHz
TDP	51 W	65 W
Дата выхода	Jan 2017	Jun 2013
Поддержка памяти	DDR4	DDR3
Integrated graphics	Intel HD 630	Radeon HD 8670D

Intel Pentium на одно поколение взрослее AMD A10-9700 по цене соотносится с G4620. Как и процессоры Ryzen, он подходит для платформы AM4, но относится к предыдущему поколению. A10 не дотягивает до уровня Intel Pentium, и с энергоэффективностью у него похуже из-за старой 28-нанометровой литографии — современные ЦП выпускаются на техпроцессе 14 нм.

Таким образом, до недавнего времени в категории офисных решений платформа Intel представляла собой лучший вариант. Тем не менее, это не абсолютный вывод, поскольку AMD выпустила APU Ryzen с интегрированной графикой, а это означает, что у Pentium G4620 появился очень серьезный конкурент.

## 5.2 Операционные системы ПК (DOS, Windows)

**Операционная система** - это совокупность программных средств, осуществляющих управление ресурсами ЭВМ, запуск прикладных программ и обеспечивающая их взаимодействие с внешними устройствами и другими программами, а также обеспечивающая диалог пользователя с ПК.

ОС хранится во внешней памяти ЭВМ на диске. При включении компьютера она из дискретной памяти переносится в оперативную - это процесс загрузки ОС.

ОС для ПК классифицируется:

1. по количеству одновременных работ пользователей,
2. по количеству одновременно выполняемых программ: однозадачные, многозадачные,
3. по разрядности кода ОС: 8, 16, 32, 64, 128 разрядные,
4. по типу интерфейса пользователя: командные и графические,
5. по типу использования ресурсов: локальные и сетевые.



ОС для ПК ориентированного на профессиональное применение должно содержать следующие основные программные компоненты:

1. Управление вводом и выводом,
2. Управление файловой системой,
3. Планирование процессов,
4. Анализ и выполнение команд, адресных ОС.

Каждая ОС имеет свой командный язык, который позволяет те или иные действия.

## **Семейство DOS**

С 1981 года и до недавнего времени на большинстве компьютеров была установлена операционная система MS-DOS (MS DiskOperating System – дисковая операционная система фирмы MS). 16-разрядная однозадачная ОС DOS обладала интерфейсом командной строки, т.е. все команды пользователю приходилось набирать на клавиатуре вручную, в командной строке DOS. Никакой графики и сервиса предусмотрено не было. Главным уязвимым местом и недостатком DOS оставалась работа с оперативной памятью. В начале 80-х гг. (в момент создания DOS) оперативная память большинства компьютеров не превышала 256 Кбайт, DOS могла работать с 640 Кбайтами оперативной памяти. Но время шло, и ОЗУ на компьютерах уже превышало 1Мбайт, появились программы, которым для работы требовался весь объем оперативной памяти. Стандартный сервис DOS этой возможности не предоставлял, система не могла размещать загружаемые при включении компьютера программы вне области 640 Кбайт. Возникал парадокс: какой бы объем оперативной памяти не имел бы компьютер, программа не могла на нем запускаться, если не было достаточно свободного пространства (640 Кбайт). Второй недостаток DOS – невозможность работы в полноценном графическом режиме, DOS не позволяла работать с загружаемыми драйверами для различных видеокарт. Третий недостаток – однозадачность. Она не позволяла запускать на компьютере сразу несколько программ с возможностью переключения между ними. В результате, с появлением ОС MS Windows, DOS практически сошла со сцены.

Операционные системы DOS, несмотря на свою простоту и экономичность, морально устарели, и на смену им пришли операционные системы нового поколения.

## **Семейство MS Windows**

### **Windows 3.x**

В 1985 году была выпущена первая версия графической оболочки Windows 1.0, представлявшая собой дополнение к MS-DOS, но популярность она завоевала только в 1990 году, когда вышла версия **Windows 3.0** и выпущенные затем **Windows 3.1** и **Windows for Workgroups 3.11**.

Эти версии Windows не были полноценными операционными системами, а являлись надстройками к операционной системе MS-DOS и были по сути многофункциональным расширением, добавляя поддержку новых режимов работы процессора, поддержку многозадачности, обеспечивая стандартизацию интерфейсов аппаратного обеспечения и единообразие для пользовательских интерфейсов программ. Предоставляли встроенные средства (GDI) для создания графического интерфейса пользователя.

Версия Windows 3.11 for Workgroups, она содержала полную поддержку мультимедиа и работу в локальной сети. Однако работа Windows отличалась неустойчивостью и частыми зависаниями с большим количеством ошибок.

### **Windows 95/95 OSR2.**

Появилась в 1995 г. и впервые Windows превратилась из графической настройки для DOS в полноценную ОС. Однако в качестве основы в Windows 95 по-прежнему использовалась DOS. Новая оболочка стала 32-разрядной, однако в целях совместимости со старыми программами ее ядро содержало старые 16-разрядные модули, а, следовательно, нестабильность работы системы осталась. Большое преимущество у новой ОС – абсолютно новый графический интерфейс, более удобный и красивый по сравнению с предыдущими версиями. Впервые был интегрирован программно-драйверный комплекс, предоставляющий приложениям Windows95 прямой доступ к аппаратным устройствам ПК – звуковой карте, видеоплате и т.д. Новая система обладала более совершенным механизмом настройки и конфигурации, а самое главное – Windows95 внесла серьезные изменения в способ расположения файлов на жестком диске (файловую систему).

Существовала в двух изданиях: Windows95 и Windows95OSR2 (добавлена поддержка устройств USB 1.1 и файловой системы FAT32; прежняя версия ОС понимала только FAT16).

### **Windows 98/98 SE**

Появилась в 1998 г., особых отличий Windows 98 от ее предшественниц не много. Основные изменения коснулись интерфейса – «Рабочий стол» стал красивее и полностью интегрирован со средой Internet, в нее включен был браузер Internet Explorer. Добавлены многие библиотеки и базы драйверов.

### **Windows ME (Millennium Edition — редакция тысячелетия)**

Появилась в 2000г. Операционная система Windows ME стала наследницей Windows 98, но приобрела многие новые возможности. Прежде всего, это улучшенная работа со средствами мультимедиа, возможность записывать не только аудио, но и видеоинформацию, мощные средства восстановления информации после сбоев и многое другое. Основным нововведением является новая версия пакета Internet Explorer 5.5, кроме того, в систему была введена поддержка цифровых устройств; система отказалась от поддержки режима MS-DOS и ряда программ для него. Повысилась стабильность работы.

Век Windows ME был недолог, и в 2001 году появилась новая операционная система Windows XP.

### **Windows NT**

Рассмотренные ранее ОС принадлежат к классу так называемых «домашних» ОС, однако фирма Microsoft разрабатывала и другие операционные системы, рассчитанные на корпоративных пользователей, на работу в локальных компьютерных сетях.

Первая версия 16-разрядной Windows NT (v.3.51) появилась в 1993 г. В 1998 году выпущена Windows NT (v.4.0); для этой ОС написано 6 пакетов обновлений, и последний, шестой, преобразует ОС в 32-разрядную. Эта ОС как сверхстабильная, надежная система, рассчитанная на работу, а не на entertainment (игрушки-развлечения). Случаи ошибок, крахов и зависаний встречаются крайне редко. Это происходит потому, что Windows NT надежно разделяет работающие под ее управлением программы. В Windows NT каждой программе выделяется своя доля адресного пространства оперативной памяти и системных ресурсов.

Существуют модификации Windows NT: WorkStation ( Рабочая станция) и Server ( Сервер).

Требования к компьютеру у этой системы вдвое выше, чем у Windows 98. Большая часть достоинств NT проявляется лишь в сетевом режиме работы.

### **Windows 2000**

Система Windows 2000 появилась в 1999 г., цель ее создания – установиться не только на корпоративном рынке, но и на домашних компьютерах. Она разработана на основе Windows NT и унаследовала от нее высокую надежность и защищенность информации от постороннего вмешательства, отличные сетевые возможности и сервисы NT (v.5.0), однако стала удобнее и дружелюбнее домашнему пользователю.

В ОС включено множество дополнительных библиотек, кроме того весьма богата база драйверов подключаемых устройств, многие из которых

не нуждаются в принудительной установке драйвера. Однако высокая требовательность к ресурсам компьютера, слабая поддержка игрового режима привели к тому, что Windows 2000 начала уверенно работать в корпоративном секторе, а не в домашнем. Как и Windows NT, Windows 2000 была выпущена в нескольких вариантах – серверном (Server), для установки на управляющий, главный компьютер сети, и клиентском (Professional) – для рабочих станций.

### **Windows XP (eXPerience).**

Появилась в 2001 г., формально – Windows NT v.5.1, добавлена поддержка многих периферийных устройств ( WiFi – Wireless Fibre, Bluetooth, FireWire, автоматически определяются цифровые фотоаппараты и видеокамеры). Service Pack1 привнес также автоматическую поддержку устройств USB 2.0. Существует в модификациях Home Edition (Домашнее Издание) и Professional.

Одно из серьёзных нововведений – встроенная система распознавания голосовых команд и голосового ввода данных. А самое главное, помимо привычного 32-разрядного варианта Windows, Microsoft подготовила и 64-разрядную модификацию.

Windows XP – первая операционная система Microsoft с полностью настраиваемым интерфейсом, позволяющим пользователям коренным образом изменять внешность своей ОС.

Вторым «подарком» Microsoft домашнему пользователю стало «интеллектуальное» меню «Пуск». При щелчке по этой кнопке Windows XP предлагает вам меню лишь тех программ, которыми вы часто пользуетесь, для вызова же других программ вам придется нажать на кнопку «Другие программы».

Одним из наиболее приятных нововведений стала поддержка записи CD-R и CD-RW дисков на уровне самой ОС – теперь стало возможно работать с «болванками», как с обычными дискетами, перетаскивая на чистый диск нужные файлы непосредственно в проводнике.

### **Windows Vista (2007)**

**Windows Vista** как и Windows XP, была разработана исключительно как клиентская система. Она стала ответом Microsoft своим конкурентам. В ней было добавлено много 3D и анимации. Начиная с Windows 98, Microsoft всегда пыталась улучшить рабочий стол. В Windows Vista появились виджеты.

### **Windows 7 (2009)**

**Windows 7** — операционная система следующая за Windows Vista. Имеются версии для сервера (Windows Server 2008 R2), для интегрирован-

ных систем (Windows Embedded Standard 2011), мобильная — Windows Embedded Compact 2011 (Chelan, Windows CE 7.0). Операционная система поступила в продажу 22 октября 2009 года. В состав Windows 7 вошли как некоторые разработки, исключённые из Windows Vista, так и новшества в интерфейсе и встроенных программах. Из состава Windows 7 были исключены некоторые игры; приложения (Почта Windows, Календарь Windows и пр.); из меню «Пуск» исчезла возможность вернуться к классическому меню и автоматическая пристыковка браузера и клиента электронной почты.

### **Windows 8 (2012).**

В связи с широким развитием рынка мобильных устройств, Microsoft постаралась унифицировать интерфейс системы, которая работала как на ПК, так и на ноутбуках и планшетных компьютерах. В связи с этим был введен плиточный интерфейс, ориентированный на сенсорный экран.

## **5.3 Установка ОС Windows. Настройка ОС Windows**

Для установки операционной системы Windows 7 потребуется ноутбук или настольный компьютер, а также диск установщик системы. Также понадобится инструкция с картинками по установке. При установке нужно быть очень внимательными.

Перед установкой операционной системы Windows 7 необходимо запомнить два важных момента:

**Самое первое** — нужно будет определиться с диском, куда будет устанавливаться система. Обычно, устанавливают операционную систему на диск С. Поэтому, если у вас есть какие-то важные данные на рабочем столе или на диске С, то предварительно скопируйте их на другой диск или же поместите их на внешние носители.

Естественно, с новой установкой вся предыдущая информация с диска С будет стерта. Таким образом, это не означает, что понадобится полностью копировать все папки с диска С на другие диски. Вы только засорите реестр и программы могут работать неправильно. Проще произвести новую установку впоследствии.

**Второй момент**, который также важен — это драйвера к вашему устройству, будь то ноутбук или настольный компьютер. Обычно, к ноутбукам предлагаются уже драйвера на диске и идут они в комплекте. Если драйверов нет, то их можно легко скачать с сайта производителя. Драйвера нужно будет предварительно скачать и поместить на любой диск кроме С. В противном случае, вы можете впоследствии испытать проблемы с их установкой, так как возможно, что для интернета потребуется драйвер.

## Устанавливаем Windows 7

Итак, надо зайти в BIOS. Теперь предстоит пройти процедуру установки. Чтобы туда попасть, надо нажать клавишу, после того как экран загрузится. Для различных систем это может быть — Escape, Ctrl + A и другие.

Система сама предложит зайти в BIOS. К примеру, надо нажать Enter Setup. Также можно нажать Del. Если клавиша выделена, то это значит, что вы оказались в старой программной оболочке.

Сама BIOS может несколько различаться в зависимости от устройства производителя. Но базовые принципы работы не будут особо отличаться. В появившемся BIOS нужно будет перейти в раздел — Boot Device Priority. В этой разделе мы переопределим загрузку компьютера с дисков. Вместо жесткого диска поставим загрузку с CD или DVD диска, на котором находится установщик нашей системы. Позже нужно будет восстановить приоритет загрузки устройств. Таким образом, наш диск, а точнее DVD CD привод, мы расположим первым в приоритете.

Обычно по умолчанию установлена загрузка с жесткого диска. Мы же выбираем загрузку с DVD привода. Даны подсказки, как можно переместить устройство вниз или вверх. Итак, как только мы поставим CD привод на первое место, нужно будет нажать Enter и позже нажать F10 для сохранения и выхода.

После этого произойдет перезагрузка компьютера и появится черный экран с надписью “Press any key to boot from CD or DVD”.

После появления экрана нажимаете любую клавишу, появится надпись — Windows is loading files, то есть — Установка операционной системы начинается.

Как только извлечение файлов осуществится, появится другое окошко — Starting Windows и окошко непосредственно установки операционной системы Windows 7, показанное на рисунке 18.



ь компьютер. Теперь (клавишу ввода-вывода). Чтобы попасть в BIOS, надо нажать клавишу, после того как экран загрузится. Для различных систем это может быть — Escape, Ctrl + A и другие.

чтобы попасть в BIOS, надо нажать Enter Setup. Также можно нажать Del. Если клавиша выделена, то это значит, что вы оказались в старой программной оболочке.

Рисунок 18 – Окно установки Windows

Необходимо будет выбрать язык и продолжить установку.

Следующим шагом будет выбор версии Windows 7. Естественно выбирать нужно ту версию, которую собственно вы и приобретали и на которую у вас имеется ключ для активации. Если у вас ноутбук, то ключ вы

можете видеть  
Сам код мы буд

Скажем  
Windows 7 и 3  
разрядную сист  
оперативная па  
то выбрать нуж

Далее поя  
идем далее.

Далее буд  
Наш выбор — э

Следующ  
стема. В нашем случае — это локальный диск С.

Если объем памяти вашего жесткого диска превышает 250 Гб, то целесообразно будет разделить этот объем на два раздела. В одном разделе мы будем держать собственно систему — для этого будет достаточно 50 Гб. Оставшийся объем памяти будет под хранение личных данных.

В итоге, не должно остаться незанятых мест на диске и все диски должны быть пронумерованы, а иначе операционная система в дальнейшем не сможет с ними работать.

Итак, после создания разделов, нажимаете клавишу Применить.

Устанавливать систему нужно на первый созданный диск (рис. 19) и далее запустить процесс форматирования.

При этом  
ностью стерты  
мало времени.  
20).

Процеду  
явится окошко  
можете установить пароль, а также подсказку к нему (рис. 21).



енной бумажке.

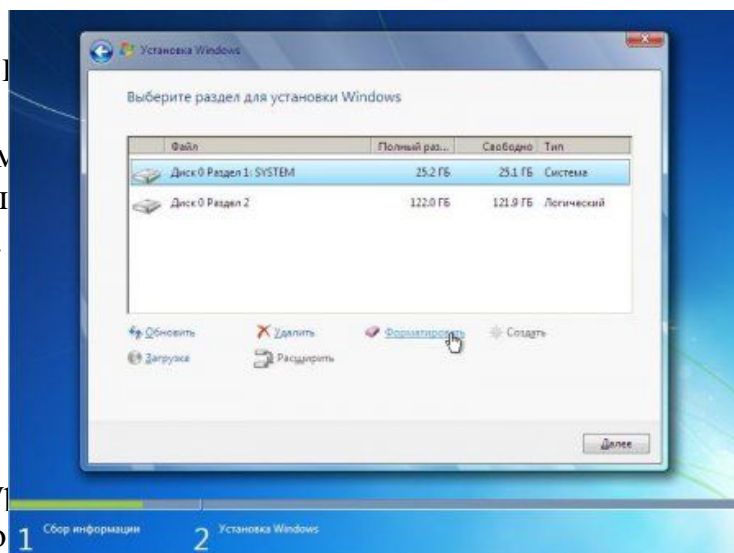
64-разрядный

но выбрать 64-  
технология и  
поддержки нет,

Омимся с ним и

ия или полная).

навливаясь си-



ые будут пол-  
с займет очень  
установки (рис.

установки по-  
а. Если хотите

Рисунок 21 – Установка пароля

Затем нужно ввести серийный код продукта. С ним вы можете ознакомиться на коробке, купленной с дистрибутивом системы. Можно пропустить этот шаг и активировать систему позже. Тогда галочку с «Автоматической активации при подключении к интернету» нужно снять.

## Настройка Windows

Далее следует выполнить некоторые настройки.

Сначала выполняется процедура выбора условий безопасности (рис. 22).

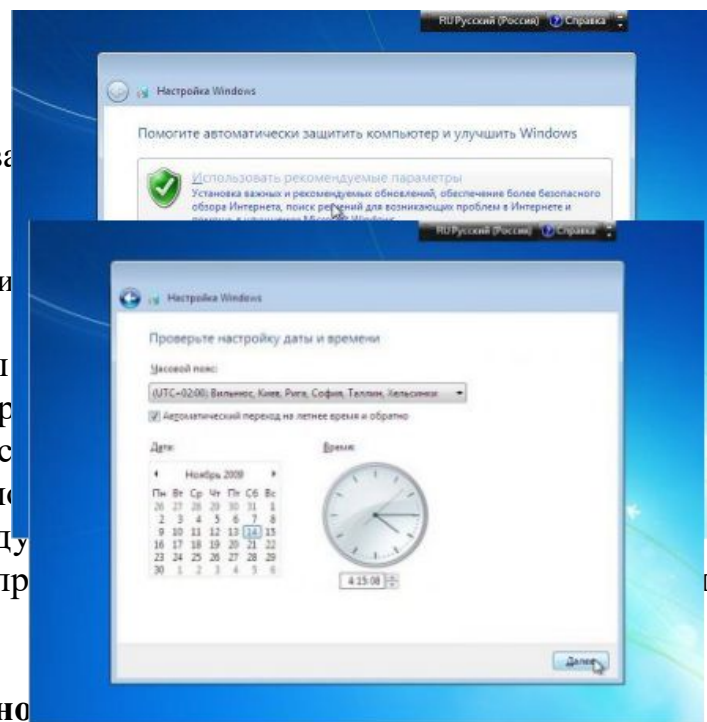
Настраива

Ри

Далее вы  
установлены др  
операционной с

Теперь мо  
вернуть предыду

На этом пр



са

в систему уже  
я Рабочий стол

BIOS, то есть

## 5.4 Устано

Хотя детали процессов установки различных программ имеют большое разнообразие, все же в них можно выделить много схожих этапов. Рассмотрим их для случая автоматизированной установки.

1. Копирование и запуск инсталлятора.

2. Ввод и проверка прав пользователя на соответствие лицензионным ограничениям. Во многих случаях копирование и распространение программного обеспечения ограничено. Чаще всего такие ограничения вызваны коммерческими интересами разработчиков программ. Этот пункт сейчас традиционно выполняется начиная с просмотра лицензионного соглашения. Пользователь должен выразить согласие с лицензией, иначе процесс установки прекратится. Если распространение программы ограничено, далее может быть запрос серийного номера программы, а может быть и пароля на установку, получаемых при покупке.



3. Запрос пользователя о будущем расположении устанавливаемой программы на дисках компьютера. Обычно здесь требуется указать каталоги (папки) в которых будут впоследствии находиться файлы программы. Часто, но не всегда такой каталог один.

4. Запросы о составе и параметрах устанавливаемого программного обеспечения. Крупные программные системы могут предусматривать различные варианты установки, отличающиеся по объему, составу устанавливаемых программ и их возможностям. Это делается с тем, чтобы система наиболее полно использовалась различными категориями пользователей с различными конфигурациями компьютеров. На этом этапе также проверяется объем свободного пространства на дисках пользователя.

5. Разархивирование и копирование файлов с дистрибутива в указанный каталог. Обычно это один из наиболее длительных этапов. В ходе этого процесса может одновременно демонстрироваться сопроводительная презентация, поясняющая возможности программного обеспечения.

6. Копирование или настройка системных библиотек, требуемых программе. Многие прикладные программы пользуются библиотеками функций и процедур, предоставляемыми разработчиками операционной системы либо третьими фирмами. В этом случае такие библиотеки также требуется расположить в заранее известных каталогах.

7. Сохранение параметров настройки прикладной программы. Эта информация сохраняется в специальных структурах данных:

- файлах конфигурации, которые часто хранятся в одном каталоге с программой;

- системных файлах; количество и структура системных файлов для сохранения такой информации различны в различных операционных системах, современные версии Windows имеют специальную системную структуру для настройки - системный реестр (registry).

8. Создание и сохранение надстроек к интерфейсу пользователя, которые далее будут облегчать запуск и применение прикладной программы. Например, к таким надстройкам относятся:

- графические ярлыки, содержащие характерные пиктограммы установленной программы;

- быстрые ссылки (shortcuts) - ссылки на исполняемые файлы программы; это позволяет пользователю не задумываться об их расположении;

- ассоциации с типами (расширениями) файлов; такая ассоциация позволяет автоматически вызывать нужную программу при обработке файла определенного типа;

- дополнительные пункты в различных меню графического интерфейса операционной системы.

9. Регистрация программного обеспечения с целью возможного последующего удаления. Данный этап позволяет впоследствии автоматически удалить все файлы программы с компьютера, если пользователь решил прекратить ее применение.

В операционных системах Microsoft Windows большинство параметров, сохраняемых при установке программ, попадают в системный реестр. Таким образом, исправность реестра очень важна для надежной работы установленных программ. Специалистам по сервису приходится иногда выполнять исправления нарушений реестра для восстановления нормальной работы программ.

## 5.5 Основы эксплуатации ЭВМ

Современные вычислительные машины представляют одно из самых значительных достижений человеческой мысли, влияние которого на развитие научно-технического прогресса трудно переоценить.

Сегодня, кроме привычных компьютеров с клавиатурами, мониторами, дисковыми, мир современной техники наполнен компьютерами-невидимками - микропроцессорами, который представляет собой компьютер в миниатюре.

Кроме обрабатывающего блока, он содержит блок управления и даже память (внутренние ячейки памяти). Это значит, что микропроцессор способен автономно выполнять все необходимые действия с информацией.

Массовое распространение микропроцессоры получили везде, где управление может быть сведено к отдаче ограниченной последовательности команд. Среди них выделяют: многопользовательские, оборудованные многими выносными терминалами и работающие в режиме разделения времени; встроенные, которые могут управлять станком, какой-либо подсистемой автомобиля, другого устройства, будучи его малой частью. Эти встроенные устройства (их называют контроллерами) выполняются в виде небольших плат.

Таким образом, созданные на основе микропроцессора вычислительные машины (микро-ЭВМ) незаменимы в современной технике.

Применение микропроцессоров даже лет 30 назад было около 2000 различных сфер: это управление производством (16%), научные исследования, транспорт и связь (17%), информационно-вычислительная техника (12%), военная техника (9%), бытовая техника (3%), обучение (2%), авиация и космос (15%), коммунальное и городское хозяйство, банковский учёт, метрология, медицина (4%) и другие области.

В настоящее время области применения ЭВМ можно разделить на группы.

*Научно-технические применения.* Для них характерно требование высокого быстродействия. Это те области науки и техники, где крайне необходимо применение вычислительных машин: ядерная физика, метеорология, ракетная техника, медицина.

*Обработка данных.* Здесь выдвигается требование большого объема запоминающих устройств. В этой группе решаются задачи из областей статистики, материально-технического снабжения, бухгалтерского учета, планирования, резервирования билетов, разработки сетевых графиков и т.д.

*Моделирование.* Вычислительные машины используются для моделирования различных сложных явлений в экономике, автоматике, биологии, военном деле и т.д.

*Управление производственными процессами.* В этом случае машина работает в так называемом реальном масштабе времени, когда арифметические и логические операции выполняются во время протекания самих производственных процессов. Роль управляющей машины может сводиться к выполнению следующих функций:

- полному информированию оператора о ходе процесса;
- сигнализации, когда существенные для протекания процесса параметры выходят за допустимые пределы;
- автономному (без участия человека) управлению протеканием процесса.

Микропроцессоры получили массовое распространение в производстве, где управление может быть сведено к отдаче ограниченной последовательности команд. Например, развиваются следующие направления автоматизации с применением микропроцессорной техники систем управления:

- станки с ЧПУ плюс робот;
- станки с ЧПУ плюс робот плюс устройство активного контроля размеров;
- станки с ЧПУ плюс робот плюс система автоматической диагностики с самовозвратом.

Сегодня вся современная техника, как бытового, так и промышленного применения, представляет собой сложные технические системы, реализованные на базе микроэлектроники и средств вычислительной техники.

Вычислительные средства являются важнейшей составной частью различных устройств техники: радиоэлектронной аппаратуры, стиральных машин, холодильников, машин химической чистки одежды и прочих технических устройств разнообразного назначения, в том числе и военного. Так, немыслимо без использования микропроцессоров управление современным двигателем - обеспечение экономии расхода топлива, ограничение максимальной скорости движения, контроль исправности и т.д.

Наибольший эффект применения микропроцессоров достигается при встраиваемом варианте его использования, когда они встраиваются внутрь приборов, устройств или машин. В настоящее время используются бытовые холодильники, стиральные машины-автоматы, печи СВЧ, телевизионные приемники, видеомагнитофоны и проигрыватели со встроенными микропроцессорами.

Таким образом, использование микропроцессоров в оборудовании позволяет повысить производительность тяжелого ручного труда, повысить качество товаров и услуг. Встраивание микропроцессоры в станки, оборудование и приборы поможет решить сложные проблемы программного регулирования технологическими процессами.

ЭВМ находят применение при выполнении широкого круга производственных задач. Так, например, диспетчер на крупном заводе имеет в своём распоряжении автоматизированную систему контроля, обеспечивающую бесперебойную работу различных агрегатов.

Компьютеры используются также для контроля за температурой и давлением при осуществлении различных производственных процессов.

Также управляются компьютером работы на заводах, например на линиях сборки автомобилей, включающие многократно повторяющиеся операции, например затягивание болтов или окраску деталей кузова.

Рассматривая использование ЭВМ в технологическом управлении, можно выделить целую группу применений, связанных с измерениями и отображениями измеренного состояния. ЭВМ оказались информационным ядром принципиально новых средств производства; гибких производственных систем (ГПС) и измерительных комплексов.

Создание на основе ЭВМ контрольно-измерительной аппаратуры, с помощью которой можно проверять изделия прямо на производственной линии, является одной из новых областей применения ЭВМ на предприятиях. Использование ЭВМ в качестве контрольно-измерительных приборов экономически более эффективно, чем выпуск в ограниченных количествах специализированных сложных приборов с вычислительными блоками. Большой эффект в машиностроении дают ГПС, состоящие из станков с числовым программным управлением, автоматизированных складских и транспортных систем, управляемых при помощи ЭВМ.

В системах управления сложными технологическими процессами за работой технологического комплекса следят многочисленные датчики-приборы, измеряющие параметры технологического процесса (например, температуру и толщину прокатываемого металлического листа), контролирующие состояние оборудования (например, температуру подшипников турбины) или определяющие состав исходных материалов и готового продукта. Таких приборов в одной системе может быть от нескольких десятков до нескольких тысяч.

Одним из важнейших свойств системы управления сложными технологическими процессами является обеспечение безаварийной работы сложного технологического комплекса. Для этого предусматривается возможность диагностирования технологического оборудования. На основе показаний датчиков система определяет текущее состояние агрегатов и тенденции к аварийным ситуациям и может дать команду на ведение облегченного режима работы или остановку вообще. При этом оператору представляют данные о характере и местоположении аварийных участков.

Таким образом, применение ЭВМ обеспечивает лучшее использование ресурсов производства, повышение производительности труда, экономию сырья, материалов и энергоресурсов, исключение тяжелых аварийных ситуаций, увеличение межремонтных периодов работы оборудования.

ЭВМ используется в техническом оснащении магазинов самообслуживания: покупки пропускают через оптическое сканирующее устройство, которое считывает универсальный код, нанесённый на покупку, по которому компьютер определяет, цену этого изделия, хранящуюся в памяти компьютера, и высвечивает ее на маленьком экране, чтобы покупатель мог видеть стоимость своей покупки. Как только все отобранные товары прошли через оптическое сканирующее устройство, компьютер немедленно выдаёт общую стоимость купленных товаров.

Мощные вычислительные системы применяются в банковских операциях, что позволяет выполнять большое количество операций, включая обработку чеков, регистрацию изменения каждого вклада, приём и выдачу вкладов, оформление ссуды и перевод вкладов с одного счёта на другой или из банка в банк.

Кроме того, крупнейшие банки имеют автоматические устройства, расположенные за пределами банка.

Банковские автоматы позволяют клиентам не выстаивать длинных очередей в банке, взять деньги со счёта, когда банк закрыт.

Чрезвычайно возрос уровень применения ЭВМ в медицине, которая становится все более и более автоматизированной. Сложные современные исследования в медицине не мыслимы без применения вычислительной техники.

К таким исследованиям можно отнести компьютерную томографию, томографию с использованием явления ядерно-магнитного резонанса, ультрасонографию, исследования с применением изотопов.

В медицине широко применяются и экспертные системы, основное назначение которых - медицинская диагностика. Диагностические системы используются для установления связи между нарушениями деятельности организма и их возможными причинами.

Кроме того, ЭВМ применяется для формирования различного рода двигательных навыков в составе тренажеров при обучении различным профессиям: летчиков, машинистов, водителей и других.

Итак, развитие вычислительной техники и сферы ее использования - процессы взаимосвязанные и взаимообусловленные.

С одной стороны, потребности народного хозяйства стимулируют поиски учеными новых путей построения ЭВМ, а с другой стороны, появление ЭВМ с большими функциональными возможностями, с существенно улучшенными показателями по производительности, надежности и т.п., создает предпосылки для непрерывного расширения областей и развития форм применения ЭВМ.

### **Контрольные вопросы к Разделу 5**

1. Охарактеризуйте различия между процессорами Intel и AMD.
2. Что такое операционная система?
3. По каким параметрам классифицируются ОС?
4. Дайте характеристику ОС семейства MS-DOS.
5. Перечислите ОС, входящие в семейство MS Windows.
6. Перечислите этапы процесса установки ОС Windows.
7. Как выполняется настройка Windows?
8. Перечислите этапы процесса установки прикладных программ.
9. Назовите основные области применения ЭВМ и микропроцессоров.

## **ЗАКЛЮЧЕНИЕ**

Данное учебное пособие представляет систематизированный учебный материал по техническому обеспечению компьютерных систем.

Пособие может активно использоваться в учебном процессе при реализации программы подготовки по специальности 09.02.03 Программирование в компьютерных системах обучающимися очной формы обучения.

Пособие может быть полезно при подготовке к практическим работам, при выполнении самостоятельной работы обучающимися. Учебное пособие представлено в электронном виде, что позволяет обучающимся в любое время использовать информационный ресурс для организации своей учебной деятельности.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Федотова, Е. Л. Информационные технологии в профессиональной деятельности: учеб. пособие для сред. проф. образования. - М.: ФОРУМ-ИНФРА-М, 2015. (основная литература)

2. Степина В.В. Архитектура ЭВМ и вычислительные системы : учебник / В.В. Степина. — М.: КУРС: ИНФРА-М, 2017. — 384 с. — (Среднее профессиональное образование). URL: <http://znanium.com/bookread2.php?book=661253> (основная литература)

3. Максимов Н.В. Архитектура ЭВМ и вычислительных систем : учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. — 5-е изд., перераб. и доп. — М. : ФОРУМ : ИНФРА-М, 2017. — 511 с. — (Среднее профессиональное образование). URL: <http://znanium.com/bookread2.php?book=814513> (основная литература)

4. Партыка Т. Л. Периферийные устройства вычислительной техники: Учебное пособие / Т.Л. Партыка, И.И. Попов. - 3-е изд., испр. и доп. - М.: Форум: НИЦ ИНФРА-М, 2014. - 432 с.: ил.; 60x90 1/16. - (Профессиональное образование)/ URL: <http://znanium.com/bookread2.php?book=424031> (дополнительная литература)