Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Амурский государственный университет»

«Кадры для регионов»

ФГБОУ ВПО «Амурский государственный университет»

Учебное пособие подготовлено в рамках реализации проекта по подготовке высококвалифицированных кадров для предприятий и организаций регионов («Кадры для регионов»)

**Д.А. Теличенко**

# КОМПЬЮТЕРНЫЕ, СЕТЕВЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Учебное пособие

Благовещенск
Издательство АмГУ
2013

ББК 32.973
Т 31

*Печатается по решению редакционно-издательского совета Амурского государственного университета*

*Работа выполнена в рамках гранта «Подготовка высококвалифицированных кадров в сфере электроэнергетики и горно-металлургической отрасли для предприятий Амурской области» по заказу предприятия-партнера ОАО «Дальневосточная распределительная сетевая компания»*

Теличенко Д.А.

**«Компьютерные, сетевые и информационные технологии».** Учебное пособие. – Благовещенск: Амурский гос. ун-т, 2013.

Настоящие методические рекомендации предназначены для лиц, обучающихся по направлению 140400.68 «Электроэнергетика и электротехника», по магистерской программе «Электроэнергетические системы и сети», а так же других студентов, магистров и бакалавров, изучающих дисциплины связанные с компьютерными, сетевыми и информационными технологиями, вычислительными машинами и их системами.

***В авторской редакции.***

*Рецензенты:* доцент кафедры энергетики, канд. техн. наук, А.А. Казакул,
зам. генерального директора по техническим вопросам – главный инженер ОАО «ДРСК», А. В. Михалев

© Амурский государственный университет, 2013
© Теличенко Денис Алексеевич, 2013



## *ПРЕДИСЛОВИЕ*

Дисциплина «Компьютерные, сетевые и информационные технологии» относится к базовой части общенаучного цикла М.1.Б.3.

Дисциплина базируется на курсе «Информатика» и аналогичных, изучаемых ранее.

Знания, умения и навыки, приобретенные при изучении дисциплины «Компьютерные, сетевые и информационные технологии» могут быть использованы в таких предметах как «Математическое моделирование и теория искусственного интеллекта», «Современные проблемы электроэнергетики», «Современные электроэнергетические системы», «Новые информационные технологии в диспетчерском управлении - научный семинар», «Информатизация образования» и других, где требуется знание компьютерных, сетевых и информационных технологий, при подготовке к государственной аттестации, а так же в последующей профессиональной деятельности.

*Цель дисциплины*: познакомить обучающихся с современными достижениями компьютерных, сетевых и информационных технологий, а так же способами их использования в профессиональной деятельности.

*Задачи дисциплины*:

- ознакомить с архитектурой, принципами построения и работы компьютерных, сетевых и информационных средств;
- привить навыки по оценке, выбору и использованию современных компьютерных, сетевых и информационных технологий для выполнения различных задач в области профессиональной деятельности.

## *ВВЕДЕНИЕ*

Настоящее учебное пособие является неотъемлемой частью изучения курса «Компьютерные, сетевые и информационные технологии».

Все темы, рассматриваемые в дисциплине можно представить в виде таблицы.

| № п/п | Раздел дисциплины | Семестр | Неделя семестра | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (час). | | | | Формы текущего контроля (*в недели семестра*), промежуточной аттестации (*в семестре*) | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Лек. | Пр. | Лаб. | Сам. | Тек. | Пром. |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| *Глава 1. «Компьютерные системы и технологии»* | | | | | | | | | |
| 1 | История развития, архитектура и базовые понятия компьютерных технологий | 1 | 1 | | 2 | | 9 | | Тест, зачет |
| | | | 2 | | | 2 | | | |
| | | | 3 | | 1 | | | | |
| 2 | Особенности вычислительных машин и компьютерных систем | | 3 | | 1 | | | | |
| | | | 4 | | | 2 | | | |
| | | | 5 | | 2 | | 9 | | |
| 3 | Современные вычислительные системы и технологии | | 7 | | 2 | | | | |
| | | | 9 | | 1 | | 6 | | |
| *Глава 2. «Сетевые технологии»* | | | | | | | | | |
| 4 | Принципы построения вычислительных сетей | 1 | 9 | | 1 | | | | Тест, зачет |
| | | | 11 | | 2 | | 2 | | |
| 5 | Коммутация и маршрутизация при передаче данных по сети | | 13 | | 2 | | | | |
| | | | 15 | | 1 | | 2 | | |
| 6 | Различные сети и их использование | | 15 | | 1 | | | | |
| | | | 17 | | 2 | | 2 | | |
| *Глава 3. «Информационные технологии»* | | | | | | | | | |
| 7 | ИТ технологии в решении офисных задач | 1 | 6 | | | 2 | 1 | ЛР-2 | Тест, зачет |
| | | | 8 | | | 2 | 1 | | |
| 8 | ИТ технологии для черчения и графики | | 10 | | | 2 | 1 | ЛР-3 | |
| | | | 12 | | | 2 | 1 | | |
| 9 | ИТ технологии для автоматизации расчетов и моделирования | | 14 | | | 2 | 1 | ЛР-4 | |
| | | | 16 | | | 2 | 1 | | |
| | | | 18 | | | 2 | | | |

В настоящем пособии детально рассмотрены главы 1 и 2 (темы 1-6). Информация по главе 3 (темы 7-6) представлена в методических указаниях для самостоятельной работы.



## *УСЛОВЫЕ ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И ОПРЕДЕЛЕНИЯ*

*Алгоритмический язык* (Algorithmic language) – искусственный язык, предназначенный для выражения алгоритмов [3];

*Архитектура вычислительной машины* (Computer architecture) – концептуальная структура вычислительной машины, определяющая проведение обработки информации и включающая методы преобразования информации в данные и принципы взаимодействия технических средств и программного обеспечения [2];

*Ассемблер* (Assembler) – программа или техническое средство, выполняющие ассемблирование [3];

*БД, база данных* (database) – совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними [1];

*Б, Байт* (Byte) – единица представления данных в виде групп из 8 бит [2];

*Бит* (Bit) – двоичная единица представления данных [2];

*ВП, внешняя память*[1] (External storage) – память, данные в которой доступны центральному процессору посредством операции ввода-вывода [2];

*Ввод данных* (Data input) – операция чтения данных с носителя, запись этих данных в память данных [2];

*Верификация программы* (Program verification) – доказательство того, что поведение программы соответствует спецификации на эту программу [3];

*Векторный дисплей* (Directed beam display device) – дисплей, в котором примитивы вывода могут быть сгенерированы в любом порядке, задаваемом программой [4];

*Видовое преобразование* (Viewing transformation) – преобразование, которое привязывает границы и внутреннее содержание окна к границам и внутреннему содержимому поля вывода [4];

---

[1] *Прим.* Сюда относят системы хранения данных на дисках, таких как жесткие – HDD, гибкие, CD, DVD и т.п.. в том числе различные твердотельные накопители типа Flash, SD.

*Виртуальный* (Virtual) – определение, характеризующее процесс или устройство в системе обработки информации кажущихся реально существующими, поскольку все их функции реализуются какими-либо другими средствами [2];

*Виртуальное пространство* (Virtual space) – пространство, в котором координаты примитивов вывода не зависят от устройств [4];

*Визуализация* (Display) – визуальное представление данных [4];

*Выделение* (Highlighting) – действие, направленное на выделение изображения примитива вывода или сегмента путем модификации его визуальных атрибутов [4];

*Вывод данных* (Data output) – операция чтения данных в памяти данных и последующая их запись на носитель данных или отображение на экране [2];

*Графический терминал* (Display console) – терминал, который включает, по крайней мере, одну поверхность визуализации и может включать одно или несколько устройств ввода [4];

*Гипермедиа* (hypermedia) – представление данных в виде информационных блоков, соединенных гиперсвязями [1];

*Дамп* (Dump) – данные, представляющие собой полное или частичное содержимое оперативной памяти, выводимое на периферийное устройство [3];

*Данные* (data) – представление информации в формализованном виде, пригодном для передачи, интерпретации или обработки [1];

*Диалоговый режим* (Conversational mode) – режим взаимодействия человека с системой обработки информации, при котором человек и система обмениваются информацией в темпе, который соизмерим с темпом обработки информации человеком [2];

*Драйвер* (Driver) – программа, предназначенная для управления работой периферийных устройств [3];

*ЗУ, запоминающее устройство* (Storage unit) – устройство, реализующее функцию памяти данных [2];



*Загрузка в память* (Load) – пересылка данных между различными уровнями памяти данных с целью непосредственного их использования в операциях центрального процессора [2];

*Защита информации* (information security) – деятельность, направленная на предотвращение утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию [1];

*Зеркальное отражение* (Mirroring) – поворот примитивов вывода на 180 градусов вокруг некоторой оси в плоскости поверхности визуализации [4];

*ИТ, информационная технология* (information technology) – процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления этих процессов и методов [1];

*ИТС, информационно-телекоммуникационная сеть* (information telecommunication network) – технологическая система, предназначенная для передачи по линиям связи информации, доступ к которой осуществляется с использованием средств вычислительной техники [1];

*ИС, информационная система* (information system) – совокупность содержащейся в базах данных информации и информационных технологий, а также технических средств, обеспечивающих ее обработку [1];

*Изображение* (Display image) – совокупность примитивов вывода и (или) сегментов, которая может быть одновременно выведена на поверхность визуализации [4];

*Интерактивный режим* (Interactive mode) – режим взаимодействия системы обработки информации с человеком, выражающийся в разного рода воздействиях на этот процесс, предусмотренных механизмом управления конкретной системы и вызывающих ответную реакцию процесса [2];

*Интерпретатор* (Interpreter) – программа или техническое средство, выполняющие интерпретацию; большинство интерпретаторов осуществляют интерпретацию программы путем последовательной интерпретации ее предложений [3];

*Интерфейс* (Interface) – совокупность средств и правил, обеспечивающих взаимодействие устройств вычислительной машины или системы обработки информации и (или) программ [2];

*Информация* (information) – сведения, сообщения, данные независимо от формы их представления [1];

*Искусственный интеллект* (Artificial intelligence) – способность вычислительной машины моделировать процесс мышления за счет выполнения функций, которые обычно связывают с человеческим интеллектом [2];

*КВВ, канал ввода-вывода*[2] (Input-output channel, I/O C) – устройство, обеспечивающее пересылку данных между основной памятью ЭВМ и периферийными устройствами [2];

*КИС, корпоративная информационная система* (corporate information system) – информационная система, доступ к которой имеет ограниченный круг лиц, определенный ее владельцем или соглашением участников этой информационной системы [1];

*К, компьютер или ВМ, вычислительная машина* (computer) – техническое средство, способное выполнять множественные арифметические и логические операции на основе заданной программы и данных [1]; либо совокупность технических средств создающая возможность проведения обработки информации и получение результата в необходимой форме [2];

*Компилятор* (Compiler) – программа или техническое средство, выполняющие компиляцию [3];

*Компиляция* (Compilation) – трансляция программы с языка высокого уровня в форму, близкую к программе, на машинном языке [3];

*Компьютеризация* (Computerization) – автоматизация каких-либо процессов в любой области деятельности человека за счет применения вычислительных машин [2];

[2] *Прим*. В настоящее время более применим термин *контроллер ввода вывода*, что подчеркивает реализацию данного элемента в виде самостоятельного микропроцессорного устройства, который так же может быть интегрирован как в сам процессор, так и в другие элементы системы.



*Кувыркание* (Tumbling) – динамическое изображение примитивов вывода, вращающихся вокруг некоторой оси, ориентация которой непрерывно изменяется в пространстве [4];

*Курсор* (Cursor) – перемещаемая видимая отметка, используемая для указания позиции на поверхности визуализации, над которой будет осуществляться следующая операция [4];

*Линейная графика* (Line graphics) – область машинной графики, в которой изображения генерируются при помощи команд визуализации и координатных данных [4];

*Маркер* (Marker) – символ заданного вида, который используют для обозначения конкретной позиции на поверхности визуализации [4];

*Масштабирование* (Scaling) – увеличение или уменьшение всего изображения или его части; масштабирование можно проводить необязательно с одним и тем же коэффициентом [4];

*Машинная графика* (Computer graphics) – совокупность методов и приемов для преобразования при помощи ЭВМ данных в графическое представление или графического представления в данные [4];

*Машинное моделирование* (Simulation) – реализуемый на вычислительной машине метод исследования, предполагающий замену реального процесса его математической моделью [2];

*Машинный язык* (Machine language) – язык программирования, предназначенный для представления программ в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации; для выполнения программы на машинном языке не требуется применение трансляторов, компиляторов и интерпретаторов [3];

*Медиа-средство* (medium) – средство восприятия, отображения и/или хранения, передачи данных [1];

*МикроЭВМ*[3] *(Microcomputer)* – ЭВМ, относящаяся к классу вычислительных машин, центральная часть которых построена на одном или нескольких микропроцессорах и разработанных исходя из требования минимизации физического объем [2];

*Мультимедиа* (multimedia) – совместное использование нескольких медиа-средств [1];

*Накладываемое изображение* (Foreground image) – часть изображения, которая может изменяться при любых операциях [4];

*Невидимая линия* (Hidden line) – отрезок линии на проекции трехмерного объекта, отсутствующий на изображении, так как он закрыт поверхностью этого же или другого объекта [4];

*Носитель данных* (Data medium) – материальный объект, предназначенный для записи и хранения данных [2];

*ОП, оперативная память*[4] (Operating storage) – память, в которой размещаются данные, над которыми непосредственно производятся операции процессора [2];

*ОС, операционная система* (Operating system) – совокупность системных программ, предназначенная для обеспечения определенного уровня эффективности системы обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователю определенного набора услуг [2];

*Обработка информации* (Information processing) – систематическое выполнение операций над данными, представляющими предназначенную для обработки информацию [2];

*Обработка текстов* (Text processing) – использование вычислительной машины для ввода, редактирования, форматирования и печати текстов и документов [2];

[3] *Прим.* Малоиспользуемое в настоящее время понятие

[4] *Прим.* Иногда можно встретить обозначение ОЗУ – оперативное запоминающее устройство. В литературе часто встречается обозначение оперативной памяти, указывающее на технологию доступа к ней, а именно RAM – random access memory (память с произвольным доступом). С точки зрения способа хранения данных, данная память обычно является энергозависимой – т.е. для обеспечения хранения данных необходима постоянно подводить питание. Так же стоит упомянуть о другом типе запоминающих устройств ROM – read only memory или ПЗУ – постоянное запоминающее устройство, где данные хранятся энергонезависимо.



*Объектно-ориентированное программирование* (Object-oriented programming) – метод построения программ как совокупность объектов и классов объектов, которые могут вызывать друг друга для выбора и выполнения операций; объекты состоят из данных и операций над данными [3];

*Объектно-ориентированный язык* (Object-oriented language) – язык программирования, который соответствует концепциям объектно-ориентированного программирования [3];

*Окно* (Window) – заданная часть виртуального пространства [4];

*Операция записи данных* (Write operation) – операция занесения порции данных в память данных [2];

*Операция устройства вычислительной машины* (Operation) – однозначно определенное действие, выполняемое устройством вычислительной машины и составляющее выполнение команды или реакцию на определенные условия [2];

*Операция чтения данных* (Read operation) – операция получения порции данных из памяти данных [2];

*Основная память*[5] (Main storage) – Оперативная память центрального процессора или ее часть, представляющая единое пространство памяти [2];

*Отсечение* (Clipping) – удаление примитивов вывода или их частей, лежащих вне заданной области [4];

*ПАМ, память, память данных* (Storage) – функциональная часть вычислительной машины или системы обработки информации, предназначенная для приема, хранения и выдачи данных [2];

*ПО, Программное обеспечение* – совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ [3];

*ПП, программный продукт* (software product) – программное обеспечение и соответствующая документация, предназначенные для поставки пользователю [1];

[5] *Прим.* К этой памяти можно отнести регистры процессора или его КЭШ-память. В настоящее время данный термин чаще используют для общей характеристики памяти вычислительной системы, исключая адресное пространство внешних запоминающих устройств.

*ППП, пакет прикладных программ* (Application program package) – система прикладных программ, предназначенная для решения задач определенного класса [2];

*ПК, персональный компьютер, персональная ЭВМ* (Personal computer, PC) – настольная микроЭВМ, имеющая эксплуатационные характеристики бытового прибора и универсальные функциональные возможности [2];

*Панорамирование* (Panning) – постепенный перенос изображения с целью создания зрительного ощущения движения в горизонтальном направлении; панорамирование может быть ограничено полем вывода [4];

*Перенос* (Translating) – смещение примитивов вывода на один и тот же вектор [4];

*Пиксель* (Pixel) – наименьший элемент поверхности визуализации, которому может быть независимым образом заданы цвет, интенсивность и другие характеристики изображения [4];

*Плазменная панель* (Plasma panel) – часть дисплея, которая состоит из сетки электродов в плоской панели, наполненной газом [4];

*Поворот* (Rotation) – вращение примитивов вывода вокруг заданной оси [4];

*Подпрограмма* (Subprogram) – программа, являющаяся частью другой программы и удовлетворяющая требованиям языка программирования к структуре программы [3];

*Поле вывода* (Viewport) – заданная часть пространства визуализации [4];

*Пользователь системы обработки информации* (User) – юридическое или фактическое лицо, применяющее систему обработки информации [2];

*Поколение ЭВМ* (Computer-generation) – классификационная группа ЭВМ, объединяющая ЭВМ по используемой технологии реализации ее устройств, а также по уровню развития функциональных свойств и программного обеспечения и характеризующая определенный период в развитии промышленности средств вычислительной техники [2];

*Прерывание* (Interruption) – операция процессора, состоящая в регистрации предшествующего прерыванию состояния процессора и установлении нового состояния; прерывание является реакцией процессора на некоторые условия, возникшие в процессоре или вне его [2];



*Проблемно-ориентированный язык* (Problem-oriented language) – язык программирования, который соответствует понятиям определенного класса прикладных задач, обычно имеет набор специфических изобразительных средств [3];

*Программа* (Program) – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма [3];

*Программирование* (software product) – научная и практическая деятельность по созданию программ [3];

*Прокручивание* (Scrolling) – перемещение окна в вертикальном или горизонтальном направлениях таким образом, что новые данные появляются внутри поля вывода, а старые исчезают [4];

*Процессор*[6] (Processor) – функциональная часть вычислительной машины или системы обработки информации, предназначенная для интерпретации программ [2];

*Процессор ввода-вывода*[7] (Input-output processor) – процессор, выполняющий операции обмена данными между центральным процессором и периферийными устройствами [2];

*Прикладная программа* (Application program) – программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации [3];

*Примитив вывода* (Output primitive) – базовый графический элемент, который может использоваться для построения изображения; примитивами вывода могут быть, например, точка, отрезок линии, последовательность символов, отдельные графические элементы [4];

---

[6] *Прим*. В настоящее время более используется понятие центральный процессор; см. здесь же.

[7] *Прим*. Редко используемое обозначение, т.к. понятие контроллер ввода-вывода (см. примечание выше) вбирает в себя и это устройство.

*Примитив ввода* (Input primitive) – совокупность данных, полученных от устройств ввода; в качестве устройств ввода могут быть: клавиатура, устройство ввода альтернативы, устройство ввода позиции, устройство указания, устройство ввода чисел или устройство ввода массива позиций [4];

*Растровый дисплей* (Raster display device) – дисплей, в котором изображение генерируется с использованием методов растровой графики [4];

*Растровая графика* (Raster graphics) – область машинной графики, в которой изображения генерируются из массива пикселей, упорядоченных по строкам и столбцам [4];

*Режим пакетной обработки* (Batch processing) – режим выполнения совокупности задач, при котором все они выполняются системой обработки информации в основном автоматически без синхронизации с событиями вне этой СОИ, в частности, без связи с лицами, представившими задание для выполнения [2];

*Режим реального времени* (Real time processing) – режим обработки информации, при котором обеспечивается взаимодействие системы обработки информации с внешними по отношению к ней процессами в темпе, соизмеримом со скоростью протекания этих процессов [2];

*Рисование* (Inking) – создание линий путем перемещения устройства ввода позиций по поверхности визуализации, при котором оно оставляет за собой след, аналогично карандашу при вычерчивании линии на бумаге[4];

*СОИ, система обработка информации* (Information processing system) – совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации [2];

*Семейство ЭВМ* (Computer family) – группа ЭВМ, представляющих параметрический ряд, имеющих единую архитектуру и, в большинстве случаев, одинаковую конструктивно-технологическую базу и характеризующихся полной или ограниченной некоторыми условиями программной совместимостью [2];



*Системная программа* (System program) – программа, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности ее использования в процессе выполнения прикладных программ [3];

*Система программирования* (Programming system) – система, образуемая языком программирования, компиляторами или интерпретаторами программ, представленных на этом языке, соответствующей документацией, а также вспомогательными средствами для подготовки программ к форме, пригодной для выполнения [3];

*Супер-ЭВМ* (Supercomputer) – ЭВМ, относящаяся к классу вычислительных машин, имеющих самую высокую производительность, которая может быть достигнута на данном этапе развития технологии и в основном предназначенных для решения сложных научно-технических задач [2];

*Супервизор* (Supervisor) – часть управляющей программы, координирующая распределение ресурсов системы обработки информации [3];

*ТС, технические средства системы обработка информации* (Hardware) – все оборудование, включая носители данных, предназначенное для автоматизированной обработки информации [2];

*Транслятор* (Translator) – программа или техническое средство, выполняющие трансляцию программы; на транслятор обычно возлагаются функции диагностики ошибок, формирования словарей идентификаторов, выдачи для печати текстов программ и т.д. [3];

*Трансляция программы* (Translation) – преобразование программы, представленной на одном языке программирования, в программу на другом языке и в определенном смысле равносильную первой [3];

*Трансфокация* (Zooming) – постепенное изменение масштаба изображения с целью создания зрительного ощущения движения всей визуализируемой группы или ее части к наблюдателю или от наблюдателя; коэффициенты масштабирования должны быть одинаковыми по всем направлениям [4];

*УУ, устройство управления системы обработки информации* (Control unit) – устройство, предназначенное для выполнения функции управления в системе обработки информации в целом или ее отдельными частями [2];

*Управляющая программа* (Control program) – системная программа, реализующая набор функций управления, в который включают управление ресурсами и взаимодействием с внешней средой системы обработки информации, восстановление работы системы после проявления неисправностей в технических средствах [3];

*Устройство типа «мышь»* (Mouse) – устройство ввода позиций, которое приводится в действие перемещением по поверхности [4];

*Фоновое изображение* (Background image) – часть изображения, подобная типовой форме, которая остается неизменной при определенных последовательностях операций [4];

*ЦП, центральный процессор* (Central processing unit, CPU) – процессор, выполняющий в данной вычислительной машине или системе обработки информации основные функции по обработке информации и управлению работой других частей вычислительной машины или системы [2];

*Электронный документ* (electronic document) – документ, в котором информация представлена в электронно-цифровой форме [1];

*Электронное сообщение* (electronic message) – информация, переданная или полученная пользователем информационно-телекоммуникационной сети [1];

*ЭЦП, электронная цифровая подпись* (electronic digital signature) – реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа электронной цифровой подписи и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе [1];

*ЭВМ общего назначения, большая ЭВМ* (Mainframe computer) – ЭВМ, относящаяся к классу вычислительных машин, занимающих по шкале производи-



тельности широкий диапазон и предназначенных для решения широкого класса задач с примерно одинаковой технико-экономической эффективностью, как правило устанавливаются в вычислительных центрах [2];

*Эмуляция* (Emulation) – имитация функционирования одного устройства посредством другого устройства или устройств вычислительной машины, при которой имитирующее устройство воспринимает те же данные, выполняет ту же программу и достигает того же результата, что и имитируемо [2];

*ЯП, ячейка памяти* (Storage location) – минимальная адресуемая область памяти данных [2];

*Язык ассемблера* (Assembly language) – язык программирования, который представляет собой символьную форму машинного языка с рядом возможностей, характерных для языков высокого уровня; язык ассемблера обычно включает в себя макрокоманды [3];

*Язык высокого уровня* (High-level language) – язык программирования, понятия и структура которого удобны для восприятия человеком [3];

*Язык программирования* (Programming language) – язык, предназначенный для представления программ [3].

## ТЕМА № 1. ИСТОРИЯ РАЗВИТИЯ, АРХИТЕКТУРА И БАЗОВЫЕ ПОНЯТИЯ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

*Вопросы, рассматриваемые в данной теме*:

1. Введение. Основные понятия о процессе автоматизированной обработке данных.
2. Принцип действия вычислительной техники. Логические основы, построение и работа простейших цифровых устройств.
3. Понятие архитектуры. Классификация вычислительных машин.
4. Функциональная и структурная организация вычислительных машин.
5. Аппаратные особенности вычислительных машин различных поколений.

### Теоретические сведения

В рамках данной темы рассматривается первая часть главы 1 «История развития, архитектура и базовые понятия компьютерных технологий»

### *1. Введение.*

### *Основные понятия о процессе автоматизированной обработке данных*

Слово информация (латинское *informatio*) означает разъяснение, осведомление, изложение. Сама информация может иметь как свойства материи (т.е. может быть например передана, удалена, записана) так и абстрактные свойства (т.е. например описана математическими понятиями). Однако, безусловно, обладает особыми свойствами.

Опр. *Информация* – сведения об объекте или системе, уменьшающие степень неопределенности нашего знания.

Особые свойства информации: не выполняется закон сохранения энергии, не может существовать без материального (энергетического) носителя, является свойством материи. Для нас будет важна именно информация, обрабатывающа-



яся различными системами управления, в частности ВМ. Такая информация чаще всего представляется в виде некого знакового описания.

Важнейшая характеристика информации это адекватность.

Опр. *Адекватность* – соответствие содержания информации образу исходного объекта. Бывает нескольких видов:

а) Синтаксическая т.е. соответствие формально-структурных показателей. Например: соответствие типа носителя, формы представления, скорости передачи и т.п. Информация, рассматриваемая с синтаксической точки зрения чаще всего является именно *данными*.

б) Семантическая т.е. соответствие передаваемого смыслового содержания. Имеет смысл при единстве объекта и пользователя.

в) Прагматическая т.е соответствие содержания цели. Имеет смысл при единстве информации и цели, связанная с ценностью самой информации.

*Меры информации (в соответствии с формами адекватности).*

1. *Синтаксические меры информации.*

1.1. *Объем данных* – соответствует количеству символов (разрядов) принятого алфавита ($n$) умноженный на единицу объема ($V_1$).

$$V_{сообщ} = V_1 \cdot n$$

Для двоичной системы исчисления весь алфавит составляет две возможные «буквы» – 0 или 1; для десятичной, 10: «0», «1»… «9». Вследствие различного «веса» каждого разряда для каждой системы исчисления приняты свои единицы объема: «бит», дит, и т.д.

Пример:

сообщение для дв. сис. исчисления 1011 1100 имеет объем 8бит 1байт;

для дес. сист. исчисления 481253 имеет объем 6дит.

*Примечание*: различные системы исчисления на самостоятельное изучение. Данный вопрос так же прорабатывается на практических занятиях.

Нетрудно убедиться, что одинаковое количество разрядов в разных системах исчисления способно передать различное число всевозможных отобра-

жаемых состояний (N) или просто количество информации, которое может содержаться в сообщении длинной n разрядов

$$N = mn,$$

где $m$ – основание системы счисления (разнообразие символов, применяемых в алфавите).

1.2. *Количество переданной информации* – соответствует изменению неопределенности состояния системы. Неопределенность состояния системы может быть вычислена через энтропию. Дадим ряд определений.

Опр. *Априорная информация* – имеющаяся информация до некоторого совершаемого действия или факта.

Опр. *Апостериорная информация* – имеющаяся информация после некоторого совершаемого действия или факта.

Опр. *Энтропия* – степень неопределенности или мера недостающей информации $H(a)$ для некоторой системы $a$ по формуле Шеннона для системы, имеющей $N$ возможных состояний

$$H(a) = -\sum_{i=1}^{N} P_i \log P_i ,$$

где $P_i$ – вероятность того, что система находится в $i$-м состоянии.

Получение информации связано с изменением степени неосведомленности получателя о состоянии системы. До получения информации получатель мог иметь некоторые предварительные *(априорные)* сведения о системе $a$; мера неосведомленности о системе - H(a) и является для него мерой неопределенности состояния системы. После получения некоторого сообщения $\beta$ получатель приобрел дополнительную информацию I$\beta$(a), уменьшившую его неосведомленность так, что апостериорная (после получения сообщения $\beta$) неопределенность состояния системы стала H($a/\beta$).

Тогда *количество информации* I$\beta$(a) о системе $a$, полученное в сообщении β, будет определено как

$$Ib(a) = H(a) - H(a/b).$$



Таким образом, *количество информации* измеряется изменением (уменьшением) неопределенности состояния системы. Если конечная неопределенность H($a/\beta$) обратится в нуль, то первоначальное неполное знание заменится полным знанием и количество информации станет равно

$$Ib(a) = H(a).$$

Для случая, когда все состояния системы равновероятны, то есть $P_i = 1/N$ энтропия такова:

$$H(a) = -\sum_{i=1}^{N} \frac{1}{N} \log \frac{1}{N} = \log N .$$

При равновероятном появлении любой кодовой комбинации *количество информации* в правильном сообщении – формула-Хартли:

$$I = \log N = n \log m .$$

Если в качестве основания логарифма принять $m$, то $I=n$. В данном случае количество информации (при условии полного априорного незнания получателем содержания сообщения) будет равно объему данных $I=V_д$ (при не равновероятных состояниях всегда

$$I<V_д \quad V_д=n.$$

Наиболее часто используются логарифмы с основанием два (для двоичной системы) и десять (для десятичной системы).

1.3. *Степень информативности сообщения* (лаконичность) $Y$ определяется отношением количества информации к объему данных, то есть

$$Y=I/V_д, \ 0<Y<1.$$

С уменьшением $Y$ уменьшается работа по преобразованию информации, т.е. декодированию.

2. *Семантические меры информации.*

Для измерения смыслового содержания информации, то есть ее количества на семантическом уровне, наибольшее признание получила *тезаурусная мера* информации, предложенная Ю. И. Шнейдером, которая связывает семантические свойства информации со способностью пользователя воспринимать поступившее сообщение. Используется понятие *тезаурус пользователя.*

*Тезаурус* можно трактовать как совокупность сведений, которыми располагает данная система, пользователь.

В зависимости от соотношений между смысловым содержанием информации – *S** и тезаурусом пользователя – *Sn* изменяется количество семантической информации *Ic*, воспринимаемой пользователем и включаемой им в дальнейшем в свой тезаурус:

- *при* $S_n \approx 0$ пользователь не воспринимает, не понимает поступающую информацию;
- *при* $S_n \to \infty$ пользователь все знает, и поступающая информация ему не нужна. И в том, и в другом случае *Ic =0*.

Максимальное значение *Ic* приобретает при согласовании *S** с тезаурусом *Sn* когда поступающая информация понятна пользователю и несет ему ранее не известные (отсутствующие в его тезаурусе) сведения.

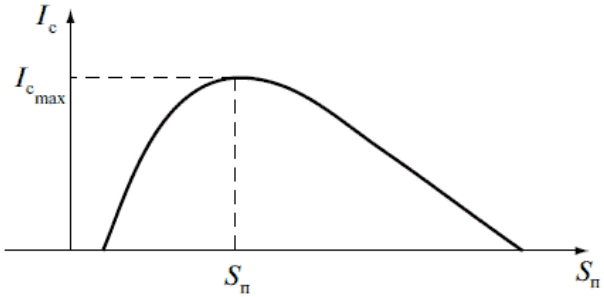



*Рис. 1.1.* Зависимость Ic=f(Sn)

Т.е. одно и то же сообщение может иметь смысловое содержание для компетентного пользователя и быть бессмысленным (семантическим шумом) для пользователя некомпетентного. Вместе с тем понятная, но известная компетентному пользователю информация представляет собой для него тоже семантический шум. При разработке информационного обеспечения систем управления следует стремиться к согласованию величин *S* Sn* так, чтобы циркулирующая в системе информация была понятна, доступна для восприятия и обладала наибольшей *содержательностью S*, то есть $S = I_c/V_д$.



3. *Прагматические меры информации.*

Это полезность информации, ее ценность для пользователя или системы управления. Эта мера также является величиной относительной, обусловленной особенностями использования информации в той или иной системе управления.

В системе управления производством, например, ценность информации определяется эффективностью осуществляемого на ее основе экономического управления, или, иначе, приростом экономического эффекта функционирования системы управления, обусловленным прагматическими свойствами информации:

$$I_{b_П}(a) = П(a/b) - П(a) = \sum_{j=1}^{Z_b} F_j R_{b_j} П_j ,$$

- $I_{b_П}(a)$ ценность информационного сообщения $\beta$ для системы управления $\alpha$;
- $П(a/b)$ ожидаемый эффект функционирования системы а при условии, что для управления будет использована информация, содержащаяся в сообщении $\beta$;
- $П(a)$ априорный ожидаемый экономический эффект функционирования, системы управления $\alpha$;
- $Z_b$ множество задач, для решения которых используется информация $\beta$;
- $F$ частоту решения каждой задачи за период времени, для которого оценивается экономический эффект;
- $R_b$ степень влияния информационного сообщения $\beta$ на точность решения задачи от 0 до 1, где $П_j$ экономический эффект от решения *j*-и задачи в системе.

<u>*Показатели качества информации*</u>

1) *Репрезентативность* – правильность, качественная адекватность отражения заданных свойств объекта. Зависит от правильности ее отбора и формирования (концепция, методика измерения, алгоритма формирования, оценки).

2) *Содержательность* информации – это ее удельная семантическая емкость, равная отношению количества семантической информации в сообщении к объему данных, его отображающих, то есть $S = Ic/V_д$. С увеличением содержатель-

ности информации растет пропускная способность информационной системы, так как для передачи одних и тех же сведений требуется преобразовывать меньший объем данных. Наряду с содержательностью можно использовать и *показатель информативности*, характеризующийся отношением количества синтаксической информации (по Шеннону) к объему данных - *Y= I/Vд*.

3) *Достаточность (полнота)* информации означает, что она содержит минимальный, но достаточный для принятия правильного решения набор показателей. Бывает и избыточная.

4) *Доступность* информации для восприятия обеспечивается выполнением соответствующих процедур ее получения и преобразования. Так, назначением вычислительной системы является увеличение ценности информации путем согласования ее с тезаурусом пользователя, то есть преобразование ее к доступной и удобной для восприятия пользователем форме.

5) *Актуальность информации* – это свойство информации сохранять свою полезность (ценность) для управления во времени. Измеряется актуальность $A(t)$

$$A(t) = Z(t) / Z(t_0),$$

где $Z(t)$ – ценность информации в определенный момент времени.

6) *Своевременность* – это свойство информации, обеспечивающее возможность ее использования в заданный момент времени.

7) *Точность* информации – это степень близости отображаемого информацией значения и истинного значения данного параметра:

- *формальная* точность, измеряемая значением единицы младшего разряда числа;
- *реальная* точность, определяемая значением единицы последнего разряда числа, верность которого гарантируется;
- *достижимая* точность – максимальная точность, которую можно получить в данных конкретных условиях функционирования системы;
- *необходимая* точность, определяемая функциональным назначением показателя и обеспечивающая правильность работы системы.

8) *Достоверность* информации – свойство информации отражать реально существующие объекты с необходимой точностью. Измеряется достоверность



информации доверительной вероятностью необходимой точности, то есть вероятностью того, что отображаемое информацией значение параметра отличается от истинного значения этого параметра в пределах необходимой точности *достоверность данных,* то есть безошибочность; измеряемая вероятностью появления ошибок в данных.

9) *Ценность* – зависит от вида информации.

## *Показатели качества информации и функционирование информационной системы*

Следует отметить, что такие показатели качества информации, как репрезентативность, содержательность, полнота, доступность, устойчивость целиком предопределяются при выборе концепции построения системы.

Показатели актуальности, своевременности, точности и достоверности обусловливаются в большей степени также на методическом уровне, однако на их величину существенно влияет и характер функционирования системы, в первую очередь, ее надежность. При этом показатели актуальности и точности жестко связаны с показателями, соответственно, своевременности и достоверности: существенное нарушение первых, приводящее к снижению эффективности функционирования системы, неизбежно вызывает нарушение вторых.

Исследования показывают, что для целевой функции оптимизации функционирования методически правильно спроектированной информационной системы в качестве ограничений, обусловливаемых показателями качества информации, достаточно использовать ограничения только *по достаточности, своевременности* и *достоверности.*

## *Выводы*

1. Процесс *обработки* (получение, преобразование, передача) предполагает представление информации в различном виде. Для нас актуальным является работа именно с «данными» – как формой представления информации.

2. Все вычислительные машины и компьютеры по своему принципу построения основываются на элементах функционирующих в ключевом режиме «вкл» «выкл», т.е. исходно описываются в двоичной системе исчисления.

3. Основной задачей обработки данных является уменьшение энтропии т.е. «незнания».

4. Основной характеристикой информации является ее адекватность, из понятия, которого вытекают все характеристики информации, и показатели качества информации перечисленные выше.

5. В дальнейшем будем считать, что вся информация, обрабатываемая вычислительной системой, имеет существенное значение, т.е. имеет прагматическую меру, а показатели качества, закладываемые на уровне проектирования задаются как для типичной информационной системы.

*Заключительные замечания*

1. Процесс автоматизированной обработки данных предполагает присутствие человека в режиме работы системы, в отличие от автоматической обработки (там человек присутствует только на этапе проектирования).

2. К основным тенденциям развития современных информационных технологий относят не только все большее освобождение человека от выполнения рутинных операций, но и вводом ИТ (основной элемент которых является ВС) в элементы принятия решений и развитие на этой базе так называемой системы искусственного интеллекта.

3. Объем, скорость, достоверность обрабатываемых данных компьютерных и информационных систем увеличивается в геометрический прогрессии.

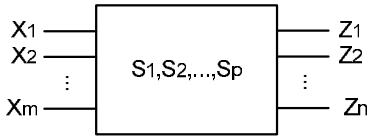
4. На настоящий момент недоступным для ВС областью деятельности человека является элемент неопределенности связанный с так называемым творчеством.



## *2. Принцип действия вычислительных машин: логические основы, построение и работа простейших цифровых устройств*

*Цифровые (дискретные) устройства* являются составной частью всех ЭВМ и ПК, систем автоматического и автоматизированного управления и предназначены для обработки, хранения и передачи дискретной (цифровой) информации.

*Модель* дискретного устройства, отражающая только его свойства по переработке сигналов, называется дискретным (цифровым) автоматом.




$X = (X_1, X_2, ..., X_m)$ – *входные сигналы*;
$Z = (Z_1, Z_2, ..., Z_n)$ – *выходные сигналы*;
$S = (S_1, S_2, ..., S_p)$ – *состояние внутренних элементов памяти*;

*Рис. 1.2.* Схема представления цифрового автомата в виде многополюсника

В общем случае, модель представляет собой многополюсный черный ящик с *m* входами и *n* выходами (рис.1.2). Состояние автомата определяется состояниями сигналов на его входах и выходах. Совокупность входных и выходных переменных *X* и *Z* образуют входное и выходное слово автомата, соответственно.

Различные значения входных переменных образуют алфавит (т.к. алфавит входных и выходных переменных един, в дальнейшем будет рассматриваться только один алфавит). В цифровой технике алфавит входного (выходного) слова содержит два значения (две буквы) "1" и "0". Каждое слово – набор переменных на входе или на выходе автомата, отличается от другого слова хотя бы одной буквой. Каждая буква слова поставлена в соответствие с номером входа (выхода) автомата.

При выполнении различных операций в современных цифровых устройствах и системах данные обычно представляются в *двоичной системе счисле-*

*ния*. Это связано с тем, что простейшее цифровое устройство – это устройство с двумя электрическими состояниями (ключ – включено, выключено). Комбинации таких простейших цифровых устройств как мы увидим ниже, позволяют получать достаточно сложные устройства.

Принято, что символ “1” (логическая единица) представляется некоторым стандартным уровнем напряжения или тока, а “0” (логический ноль) - нулевым или близким к нулю уровнем напряжения или тока.

### *Арифметические основы работы цифровых устройств*

Арифметические операции над двоичными числами могут производиться по тем же правилам, что и над десятичными, однако, с целью упрощения цифровых систем для выполнения арифметических операций применяют алгоритмы, отличные от алгоритмов действий десятичной арифметики.

В двоичной системе счисления для представления знака числа используется дополнительный знаковый разряд (один или несколько разрядов), который располагается перед старшим числовым разрядом. Для положительных чисел значение знакового разряда Зн.р.=0, для отрицательного числа Зн.р.=1.

Операция вычитания в цифровых системах реализуется с помощью операции сложения. Вычитаемое при этом представляется в дополнительном коде (если расчет не требует высокой точности - в обратном коде).

Двоичный код со знаком называют также *прямым кодом*. *Обратный код* получается путем замены всех “0” на “1” и всех “1” на “0” прямого кода (двоичного числа со знаком). Причем, знаковый разряд при этом остается неизменным. Обратный код, дополненный единицей в младшем разряде, называется *дополнительным кодом*.

*Сложение и вычитание двоичных чисел*. Правила сложения двух двоичных чисел можно показать на следующем примере:



$$0+0=0$$
$$0+1=1$$
$$1+0=1$$
$$1+1=0 \quad \textit{и } 1 \textit{ переносится в старший разряд}$$

Пример сложения многоразрядных чисел: сложить $18_{10}$ и $23_{10}$

$$\begin{array}{rcccccccc} & & & 1 & & 1 & 1 & & \\ & 18 & = & 0 & 1 & 0 & 0 & 1 & 0 \\ + & 23 & = & 0 & 1 & 0 & 1 & 1 & 1 \\ \hline & 41 & = & 1 & 0 & 1 & 0 & 0 & 1 \end{array}$$

*Вычитание* в цифровых устройствах может произвестись также как и сложение, только вычитающее представляется в дополнительном коде (знак у него не изменяется т.е. первая 1).

Операция вычитания с использованием только обратного кода (без дополнительных операций по переводу его в дополнительный код) приводит к ошибке, определяемой единицей в младшем разряде.

### *Основы алгебры логики*

Алгебра логики (*АЛ*) является основным инструментом синтеза и анализа дискретных автоматов всех уровней. *АЛ* называют также Булевой алгеброй. *АЛ* базируется на трёх функциях, определяющих три основные логические операции.

1. *Функция отрицания (НЕ)*. *F* = *X*’ читается, как *F* есть (эквивалентна) *НЕ* X. Элемент, реализующий функцию «НЕ», называется элементом «НЕ» (инвертором). Элемент «НЕ» имеет два состояния – см. рис. 1.3.

| $X$ | $F$ |
|---|---|
| 1 | 0 |
| 0 | 1 |

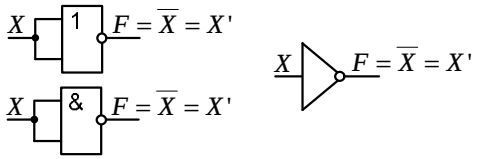



*Рис. 1.3.* Функция отрицания «НЕ»

2. *Функция логического умножения (конъюнкции).* Функция логического умножения записывается в виде $F = X_1 \cdot X_2$. Символы логического умножения &, Λ, <·>, ×. Функция конъюнкции читается так: $F$ есть (эквивалентна) $X_1$ и $X_2$, поскольку функция истинна тогда, когда истинны 1-й и 2-й аргументы (переменные). Конъюнкцию называют функцией «И», элемент, реализующий эту функцию, элементом «И». Количество переменных (аргументов), участвующих в одной конъюнкции, соответствует количеству входов элемента «И» – см. рис. 1.4.

| $X_1$ | $X_2$ | $F$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

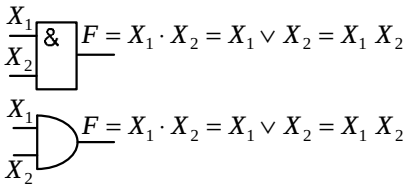



*Рис. 1.4.* Функция логического умножения «И»

3. *Логическое сложение (дизъюнкция).* Функция логического сложения записывается в виде $F = X_1 + X_2$, и читается так: $F$ есть $X_1$ или $X_2$, поскольку функция истинна, когда истинна одна или другая переменная (хотя бы одна). Поэтому функцию дизъюнкции часто называют функцией «ИЛИ». Символы логического сложения +,V.

В общем случае функция «ИЛИ» описывается как показано на рис. 1.5.

| $X_1$ | $X_2$ | $F$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

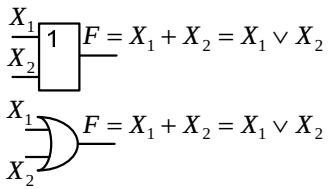



*Рис. 1.5.* «ИЛИ»

Используя операции (функции) И, ИЛИ, НЕ можно описать поведение любого комбинационного устройства, задав сколь угодно сложное булево выражение. Любое булево выражение состоит из булевых констант и переменных, связанных операциями И, ИЛИ, НЕ.



Пример булевого выражения:

$$F(X_1, X_2) = X_1 + X_2 \cdot X_1 + \overline{X_1 + \overline{X_2}}\,.$$

## *Основные законы алгебры логики*

Основные законы *АЛ* позволяют проводить эквивалентные преобразования функций, записанных с помощью операций И, ИЛИ, НЕ, приводить их к удобному для дальнейшего использования виду и упрощать запись.

Булевой алгебре свойственен принцип двойственности, что наглядно иллюстрирован в табл. 1.1. Как следует из табл. 1.1, только закон двойного отрицания не подчиняется этому принципу.

*Таблица 1.1.* Законы алгебры логики

| № | а | б | Примечание |
|---|---|---|---|
| 1 | $\overline{0} = 1$ | $\overline{1} = 0$ | Аксиомы (тождества) |
| 2 | $X + 0 = X$ | $X \cdot 1 = X$ | |
| 3 | $X + 1 = 1$ | $X \cdot 0 = 0$ | |
| 4 | $X + X = X$ | $X \cdot X = X$ | |
| 5 | $X + \overline{X} = 1$ | $X \cdot \overline{X} = 0$ | |
| 6 | $\overline{\overline{X}} = X$ | | Закон двойного отрицания |
| 7 | $X + Y = Y + X$ | $X \cdot Y = Y \cdot X$ | Закон коммутативности |
| 8 | $X + X \cdot Y = X$ | $X \cdot (X + Y) = X$ | Закон поглощения |
| 9 | $\overline{X + Y} = \overline{X} \cdot \overline{Y}$ | $\overline{X \cdot Y} = \overline{X} + \overline{Y}$ | Правило де-Моргана (дуальность) |
| 10 | $(X + Y) + Z = X + (Y + Z)$ | $X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$ | Закон ассоциативности |
| 11 | $X + Y \cdot Z = (X + Y) \cdot (X + Z)$ | $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$ | Закон дистрибутивности |

Используя законы алгебры логики, можно упростить булевы выражения, например $X_1 \cdot X_2 + \overline{X_1} \cdot X_2$.

Действительно, используя законы 2, 5 и 11 можно записать исходное выражение в виде $X_2 \cdot (X_1 + \overline{X_1}) = X_2$. Так как логическая операция $(X_1 + \overline{X_1}) = 1$ (см. з-н 5), а $X_2 \cdot 1 = X_2$ (см. з-н 2б).

## Способы задания цифровых устройств

Работу любого цифрового устройства (дискретного автомата) можно описать (задать) тремя достаточными способами: схемой, логической функцией, таблицей истинности (см. выше).

*Схемный способ (СХ).* Здесь составляется схема устройства, при этом используется как схемное обозначение трех базовых элементов (И, ИЛИ, НЕ), так и их соединений (с помощью линий связи, объединяющих входы и выходы) или более сложных устройств (исключающие или, регистры, триггеры и т.д.).

*Аналитический способ или с помощью логический функции (ЛФ).* Здесь составляется аналитическое выражение – логическая функция с использованием базовых логических операций (И, ИЛИ, НЕ). При сопоставлении ЛФ с дискретными автоматами аргументы функций, сопоставляются с входами, а сами функции с выходами дискретного автомата. Поскольку дискретный автомат имеет конечное число входов, то мы будем иметь дело с функцией конечного числа аргументов. Если автомат имеет $m$ входов, то количество входных переменных тоже $m$ и число возможных комбинаций наборов значений этих входных аргументов (переменных) $K=2^m$.

Например: $F(X) = X_1 \cdot \overline{X_2} \cdot \overline{X_3} + (X_1 + \overline{X_2})(X_2 + X_3)$.

*Табличный способ (ТИ).* При этом способе работа цифрового устройства описывается в виде таблицы истинности, представляющей собой совокупность всех наборов переменных (входных сигналов) и соответствующих им значений функции (выходов).

Таблица истинности содержит $2^m$ строк, $m$ столбцов (по количеству входов) и один столбец для записи значения функции.

Например: пусть требуется задать работу цифрового устройства с тремя входами и одним выходом.

По сути имеем булеву функцию вида $F_1(X_1,X_2,X_3)$. Здесь $m$ = 3, тогда строк в таблице (всех возможных комбинаций входов) восемь.



## Переходы от одного способа задания к другому

*Совершенная нормальная дизъюнктивная форма* (СНДФ). По таблице истинности можно составить логическое выражение, содержащее наборы переменных, в которые входят все переменные, с отрицанием или без. Одна из его форм называется СНДФ или СДНФ.

В качестве примера получения СНДФ рассмотрим случай задания логической функции в виде таблицы истинности (см. табл. 1.2) от трех переменных.

*Таблица 1.2.* Таблица истинности для функции

| № | $X_3$ | $X_2$ | $X_1$ | $F_1(X_1, X_2, X_3,)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 0 |

Из таблицы истинности видно, что функция принимает значение логической единицы только на трех наборах переменных, т.е. на 2, 4 и 5-м наборах. Для второй строки (второго набора переменных) можно записать: $X_1 = 0, \quad X_2 = 1, \quad X_3 = 0$, следовательно, функция $F(0,1,0) = 1$. Принято (по умолчанию) считать, что если переменная в "нормальном" состоянии имеет значение логической единицы, а в инверсном - логического нуля. Тогда функцию для второй строки можно представить в виде $\overline{X_1} \cdot X_2 \cdot \overline{X_3} = 1$. Для четвертой строки - $\overline{X_1} \cdot \overline{X_2} \cdot X_3 = 1$ и для пятой строки - $X_1 \cdot \overline{X_2} \cdot X_3 = 1$.

Аналитическое выражение функции выглядит как

$$F(X) = \overline{X_1} \cdot X_2 \cdot \overline{X_3} + \overline{X_1} \cdot \overline{X_2} \cdot X_3 + X_1 \cdot \overline{X_2} \cdot X_3.$$

Каждое произведение содержит все три переменные с отрицанием или без отрицания и соответствует только одной строке набора переменных, на котором функция принимает значение логической единицы. Произведения, в которых содержатся все переменные с отрицанием или без, называются *конституентами* единицы или *минтермами*. Функция будет представлять логическую сумму всех произведений, равных логической единице. В нашем примере вся

сумма (дизъюнкция) соответствует совокупности произведений переменных для трех строк.

СНДФ любой функции записывается по таблице истинности согласно следующему правилу.

Для каждого набора переменных, на которых функция принимает значение логической «1», записываются конституенты, и все эти конституенты объединяются дизъюнктивно. Переменные каждой строки, имеющие значение логического «0», в конституенты входят с отрицанием (записываются в произведение в инвертированном виде), а переменные, имеющие значения логической «1» – без отрицания.

Любую логическую (булеву) функцию можно представить дизъюнкцией конституент. Если одно из произведений не содержит хотя бы одной переменной, то такая форма называется нормальной дизъюнктивной формой.

Например:

$$F(X) = \overline{X_1} \cdot X_2 \cdot \overline{X_3} + \overline{X_1} \cdot \overline{X_2} \cdot X_3 + X_1 \cdot X_3.$$

*Совершенная нормальная конъюнктивная форма* (СНКФ). СНКФ или по другому СКНФ можно построить по таблице истинности также как СНДФ. Только для этого анализируются значения, где функция равна «0»; для каждого из таких значений составляется сумма от всех переменных (инверсии расставляются для переменных равных единицы); полученные суммы в дальнейшем умножаются.

## *Элементарные функции алгебры-логики*

Среди всех функций алгебры логики особое место занимают функции одной и двух переменных, называемые элементарными – см. табл. 1.3.

В качестве логических операций над переменными, эти функции позволяют реализовать различные функции от любого числа переменных.



*Таблица 1.3.* Функции алгебры логики

| *Переменные и функции* | | | | | *Обозначение функции* | *Назначение Функции* |
|---|---|---|---|---|---|---|
| $X_1$<br>$X_2$ | 0<br>0 | 0<br>1 | 1<br>0 | 1<br>1 | | |
| $F_0$ | 0 | 0 | 0 | 0 | $F_0 = 0$ | Генератор нуля |
| $F_1$ | 0 | 0 | 0 | 1 | $F_1 = X_1 \cdot X_2$ | «И» |
| $F_2$ | 0 | 0 | 1 | 0 | $F_2 = X_1 \cdot \overline{X_2}$ | |
| $F_3$ | 0 | 0 | 1 | 1 | $F_3 = X_1$ | |
| $F_4$ | 0 | 1 | 0 | 0 | $F_4 = \overline{X_1} \cdot X_2$ | |
| $F_5$ | 0 | 1 | 0 | 1 | $F_5 = X_2$ | |
| $F_6$ | 0 | 1 | 1 | 0 | $F_6 = X_1 \oplus X_2$ | Исключающее ИЛИ (не равнозначность) |
| $F_7$ | 0 | 1 | 1 | 1 | $F_7 = X_1 + X_2$ | «ИЛИ» |
| $F_8$ | 1 | 0 | 0 | 0 | $F_8 = \overline{X_1 + X_2}$ | «ИЛИ-НЕ» |
| $F_9$ | 1 | 0 | 0 | 1 | $F_9 = \overline{X_1 \oplus X_2} =$<br>$= X_1 \sim X_2$ | Не исключающее ИЛИ (равнозначность) |
| $F_{10}$ | 1 | 0 | 1 | 0 | $F_{10} = \overline{X_2}$ | «НЕ» $X_2$ |
| $F_{11}$ | 1 | 0 | 1 | 1 | $F_{11} = X_1 + \overline{X_2}$ | |
| $F_{12}$ | 1 | 1 | 0 | 0 | $F_{12} = \overline{X_1}$ | «НЕ» $X_1$ |
| $F_{13}$ | 1 | 1 | 0 | 1 | $F_{13} = \overline{X_1} + X_2$ | |
| $F_{14}$ | 1 | 1 | 1 | 0 | $F_{14} = \overline{X_1 \cdot X_2}$ | «И-НЕ» |
| $F_{15}$ | 1 | 1 | 1 | 1 | $F_{15} = 1$ | Генератор единицы |

## *Полная система логических функций, понятие о базисе*

Функционально полная система логических функций представляет собой набор логических функций, с помощью которых можно записать любую, сколь угодно сложную функцию. В этом случае говорят, что этот набор образует базис. Функционально полными являются 3 базиса:

1) "И-ИЛИ-НЕ" (базис конъюнкции, дизъюнкции, инверсии)

2) "И-НЕ" (базис Шеффера)

3) "ИЛИ-НЕ" (базис Пирса или функция Вебба).

Элементы, реализующие операцию «И-НЕ», «ИЛИ-НЕ» и «Исключающее ИЛИ» на принципиальных и структурных схемах изображаются так показано на рис. 1.6

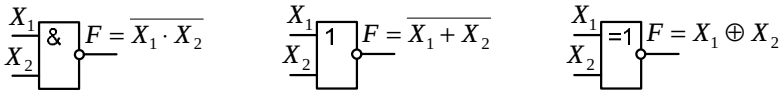



*Рис. 1.6.* Обозначение элементов

Обратим внимание на то, что любую функцию можно реализовать в базисе, если все операции (конъюнкция, дизъюнкция, инверсия) реализуются в этом базисе.

*Реализация операции «НЕ»* – показана на рис. 1.7.

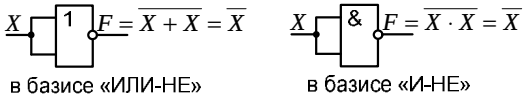



*Рис. 1.7.* Операция «НЕ»

*Реализация операции «И»* – показана на рис. 1.8.

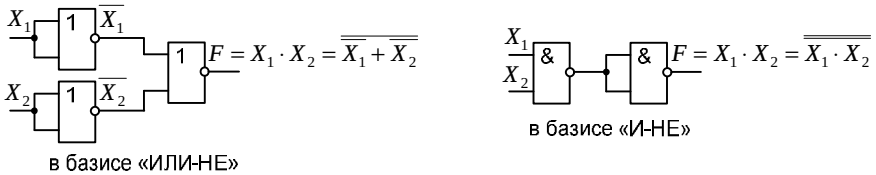



*Рис. 1.8.* Операция «И».

Реализация операции «ИЛИ» – показана на рис. 1.9.

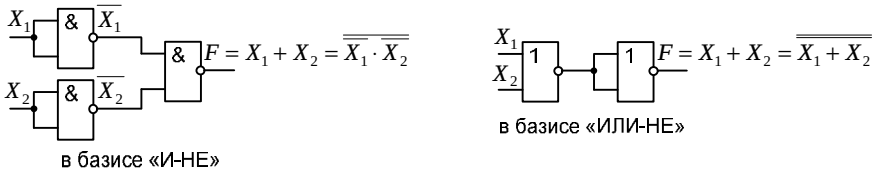



*Рис. 1.9.* Операция «ИЛИ»



*Пример* реализации комбинационного устройства в базисе «И-НЕ».

Пусть задана функция, реализуемая комбинационным устройством, в аналитической форме

$$F = X_1 \cdot \overline{X_2} + X_3 \cdot X_4 + \overline{X_1} \cdot X_4 .$$

Используя закон де Моргана и с учетом закона двойного инвертирования, запишем эту функцию в виде

$$F = \overline{\overline{F}} = \overline{\overline{X_1 \cdot \overline{X_2} + X_3 \cdot X_4 + \overline{X_1} \cdot X_4}} = \overline{\overline{X_1 \cdot \overline{X_2}} \cdot \overline{X_3 \cdot X_4} \cdot \overline{\overline{X_1} \cdot X_4}}$$

Как следует из полученного аналитического выражения, логическое устройство должно содержать три двухвходовых и один трехвходовый элемент И-НЕ. Функциональная схема комбинационного устройства, построенная в базисе И-НЕ, показана на рис. 1.10.

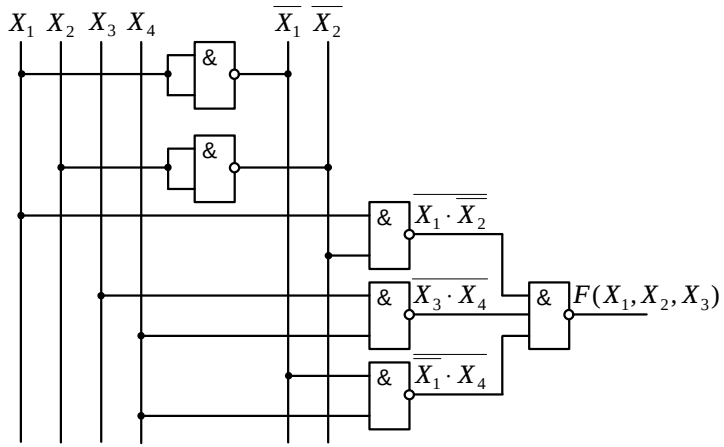



*Рис. 1.10.* Реализация функции в базисе «И-НЕ».

## <u>*Комбинационные устройства*</u>

*Комбинационной схемой* называется логическая схема, реализующая однозначное соответствие между значениями входных и выходных сигналов.

Для реализации комбинационных схем используются логические элементы, выпускаемые в виде интегральных схем. Основу комбинационных схем составляют завершенные логические схемы «И», «ИЛИ», «НЕ» и их комбинации.

В класс комбинационных устройств так же входят интегральные схемы средней (и большей) степени интеграции например схемы дешифраторов, шифраторов, мультиплексоров, демультиплексоров, сумматоров и т.п.

## Основные этапы синтеза комбинационных устройств

В задачу синтеза комбинационных устройств входит построение схемы устройства по заданным условиям его работы и в заданном базисе элементов. Задание комбинационного устройства сводится к заданию тех функций, которые оно должно реализовать.

Процесс синтеза комбинационных устройств состоит из 2-х этапов.

1. *Абстрактный синтез.* Абстрактный синтез включает:

- формирование задачи, словесное описание функций устройства, определение типа устройства;
- описание устройства на формализованных языках: таблица истинности, карта Карно, аналитическое выражение и т.д.;
- минимизация булевых функций;
- построение логической схемы устройства.

2. *Схемный синтез*. Схемный синтез состоит из следующих задач

- переход в требуемый базис;
- построение принципиальной схемы;
- разработка монтажной схемы;
- изготовление устройства и его испытания.

В результате испытаний осуществляются корректировка схемы и подготовка технической документации.

Приступим к изучению основных типов комбинационных устройств, простейшим из которых является дешифратора.



## Дешифраторы

*Дешифратор* – логическая комбинационная схема, которая имеет $n$ информационных входов и $2^n$ выходов; каждой комбинации логических уровней на входах (некое двоичное число) будет соответствовать активный уровень на одном из $2^n$ выходов (номер его соответствует поданному двоичному числу). Обычно $n$ равно 2, 3 или 4.

*Примечание:* активный уровень – это значение (0 или 1) которое является значащим для схемы или элемента (имеет место, как на входе, так и на выходе схемы или элемента).

Таким образом, при подаче на вход параллельного двоичного кода на выходе дешифратора появится сигнал на том выходе, номер которого соответствует десятичному эквиваленту двоичного кода входа.

В любой момент времени активный сигнал будет иметь место только на одном выходе дешифратора. В зависимости от типа дешифратора, этот сигнал может иметь как уровень логической единицы (при этом на всех остальных выходах уровень логического 0), так и уровень логического 0 (при этом на всех остальных выходах уровень логической 1

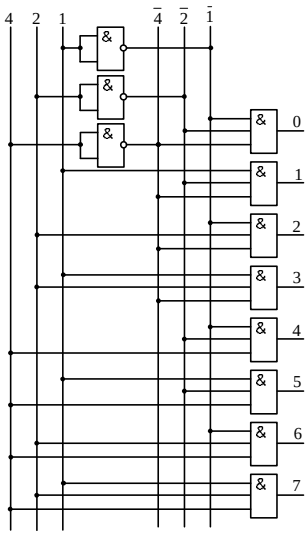



*Рис. 1.11.* Дешифратор.

Работа дешифратора может быть проиллюстрирована в соответствии с таблицей истинности 1.4.

В отечественной литературе входы дешифратора принято обозначать (1, 2, 4, 8, …), в англоязычной (*A*, *B*, *C*, …).

На рис. 1.11 представлен дешифратор, имеющий таблицу истинности 1.4.

Условное обозначения базовых элементов в табл. 1.4. соответствует обозначению, принятому в отечественной литературе.

*Таблица 1.4.* Таблица истинности дешифратора

| *Входы* | | | *Выходы* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Промышленностью стран СНГ, в том числе и России, выпускаются различные модификации дешифраторов в интегральном исполнении.

Обозначение дешифраторов на принципиальных схемах показано на рис. 1.12 – принятое в нашей стране; а на рис. 1.13 – принятое за рубежом.

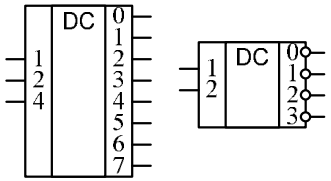



*Рис. 1.12* Дешифратор 3x8 с прямыми выходами и 2x4 с инверсными.

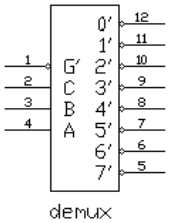



*Рис. 1.13* Дешифратор 3x8 с инверсными выходами.

Принцип формирования выходного сигнала (для схемы рис. 1.13) можно описать следующим образом:

$$Y_i = \begin{cases} 0, \text{ если } i = k, \\ 1, \text{ если } i \neq k, \\ k = 2^2 \cdot C + 2^1 \cdot B + 2^0 \cdot A. \end{cases}$$

Помимо информационных входов *A,B,C* дешифраторы обычно имеют дополнительные входы управления. Сигналы на этих входах, например, разрешают функционирование дешифратора или переводят его в пассивное состояние, при котором, независимо от сигналов на информационных входах, на всех выходах установится уровень логической единицы.



Принцип формирования выходного сигнала в таком дешифраторе с учетом сигнала управления описывается следующим образом:

$$Y_i = \begin{cases} \overline{1 \cdot \overline{G}}, \text{ если } i = k, \\ 1, \text{ если } i \neq k, \\ k = 2^2 \cdot C + 2^1 \cdot B + 2^0 \cdot A. \end{cases}$$

Так же существуют дешифраторы с двумя и более сигналами разрешения.

## Шифраторы

*Шифратор* – логическая комбинационная схема, которая имеет $2^n$ входов (где $n$ – число информационных выходов; часто $n=3$, тогда $2^n=8$). При подаче на один из входов активного сигнала, из его выходов может быть сформировано двоичное число, которое будет эквивалентное номеру входа, на котором появился активный уровень.

*Примечание*: формирование двоичного числа на выходе шифратора означает следующее: каждый из выходов шифратора считается определенным разрядом искомого двоичного числа. Принцип работы шифратора противоположен принципу работы дешифратора.

Простейшим случаем применения шифратора может служить, например, схема отслеживание нажатия одной кнопки.

## Мультиплексоры

*Мультиплексор* – комбинационная логическая схема, представляющая собой управляемый переключатель, который подключает к выходу один из входов данных. Номер подключаемого входа равен числу (адресу), определяемому комбинацией логических уровней на входах управления.

*Демультиплексорами* – называются устройства, которые позволяют подключать один вход к нескольким выходам.

Функционально мультиплексор состоит из $m$ элементов конъюнкции, выходы которых объединены дизъюнктивно с помощью элемента ИЛИ. На одни

входы всех элементов конъюнкции подаются информационные сигналы, а другие входы этих элементов соединены с соответствующими выходами дешифратора с n входами.

Рассмотрим пример простейшей схемы мультиплексора. Для этого воспользуемся базовым логическим элементом «И» с таблицей истинности 1.5.

*Таблица 1.5.*

| *Входы* | | *Выход* |
|---|---|---|
| *X* | *Y* | *Out* |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Теперь один из входов элемента будем рассматривать как информационный вход электронного ключа, а другой вход – как управляющий. По таблице истинности отчетливо видно, что пока на управляющий вход *Y* подан логический уровень «0» сигнал с входа *X* на выход *Out* не проходит. При подаче на управляющий вход *Y* логической «1», сигнал, поступающий на вход *X*, поступает на выход *Out*. То есть логический элемент «И» можно использовать в качестве электронного ключа. При этом не важно, какой из входов элемента будет использоваться в качестве управляющего входа, а какой – в качестве информационного. Остаётся только объединить выходы элементов «И» на один выход. Это делается при помощи элемента «ИЛИ» (рис. 1.14).

Чаше всего для управления требуется много входов, поэтому в схему мультиплексора включают дешифратор. Это позволяет управлять переключением входов микросхемы на выход при помощи двоичных кодов. Пример такой схемы приведен на рис. 1.15.

Кроме информационных и управляющих входов, схемы мультиплексоров содержат вход разрешения, при подаче на который активного уровня мультиплексор переходит в рабочее состояние. При подаче на вход разрешения пассивного уровня мультиплексор перейдет в нерабочее состояние, в котором сигнал на выходе сохраняет постоянное значение независимо от значений информационных и управляющих сигналов. Число информационных входов у мультиплексоров обычно равно 2, 4, 8 или 16.



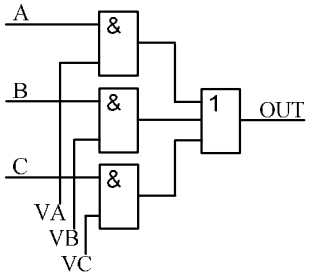



*Рис. 1.14.* Принципиальная схема мультиплексора

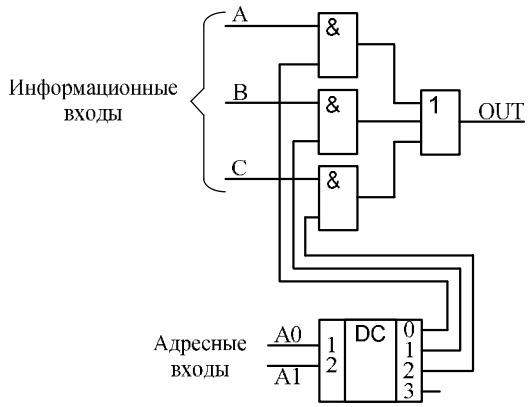



*Рис. 1.15.* Мультиплексор, управляемый двоичным кодом

*Реализация логических функций на мультиплексоре*

Функционирование мультиплексора, представленного на рис.1.16 (обозначение принятое за рубежом), описывается характеристическим уравнением, связывающим сигнал на выходе (*Y*) с разрешающим (*G*), входными информационными (*DO...D7*) и управляющими (*A, B, C*) сигналами:

$$Y = \left( \begin{array}{l} \overline{C} \cdot \overline{B} \cdot \overline{A} \cdot D0 + \overline{C} \cdot \overline{B} \cdot A \cdot D1 + \overline{C} \cdot B \cdot \overline{A} \cdot D2 + \overline{C} \cdot B \cdot A \cdot D3 + \\ + C \cdot \overline{B} \cdot \overline{A} \cdot D4 + C \cdot \overline{B} \cdot A \cdot D5 + C \cdot B \cdot \overline{A} \cdot D6 + C \cdot B \cdot A \cdot D7 \end{array} \right) \cdot \overline{G}$$

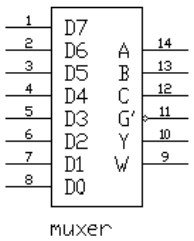



*Рис. 1.16.* Мультиплексор

Как видно из уравнения, на мультиплексоре можно реализовать логические функции, для чего нужно определить, какие сигналы и логические константы следует подавать на входы мультиплексора.

Логическая функция *n* переменных определена для $2^n$ комбинаций значений переменных. Это позволяет реализовать функцию *n* переменных на мультиплексоре, имеющем *n* управляющих и $2^n$ информационных входов.

В этом случае каждой комбинации значений аргументов соответствует единственный информационный вход мультиплексора, на который подается значение функции.

## *Сумматоры и полусумматоры*

Широкое применение в цифровой технике находят элементы, выполняющие различные арифметические действия. Операция суммирования – базовая арифметическая операция в двоичной алгебре.

1. *Сумматоры по модулю два.* Построение двоичных сумматоров обычно начинается с сумматора по модулю два. Ниже представлена таблица истинности этого сумматора. *Таблица 1.6.*

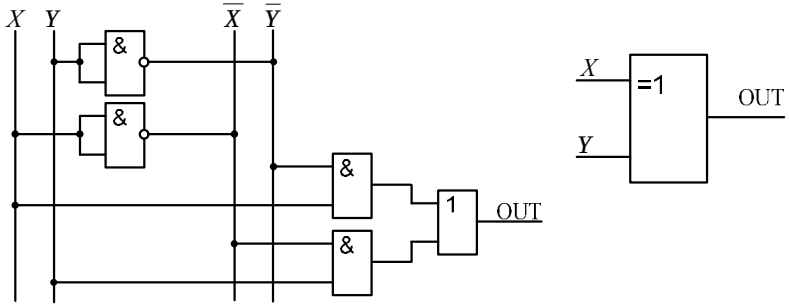
| *Входы* | | *Выход* |
|---|---|---|
| *X* | *Y* | *Out* |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Из таблицы 1.6 видно, что логическая функция, выражающая принцип работы сумматора по модулю два, имеет вид:

$$F = X \oplus Y = X\overline{Y} \vee \overline{X}Y$$

и представляет собой «исключающее ИЛИ».

На рис. 1.17 (а) представлена схемная реализация сумматора по модулю два, составленная по таблице истинности 1.6; а на рис. 1.17 (б) приведено условное обозначение этой же схемы в виде одного элемента – «Исключающего ИЛИ».




а) Расширенная схема б) «Исключающее ИЛИ»

*Рис. 1.17.* Сумматор по модулю два



Однако представленная схема имеет недостаток, который можно увидеть и в таблице 1.6 – не формируется перенос при сложении *X*=1 и *Y*=1.

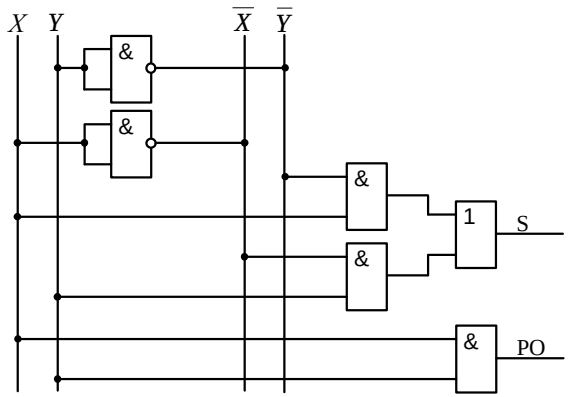
Сумматор по модулю два выполняет суммирование без учета переноса. В обычном двоичном сумматоре требуется учитывать перенос, поэтому требуются схемы, позволяющие формировать перенос в следующий разряд.

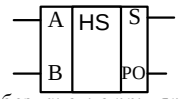
2. *Полусумматоры.* Рассмотрим более сложное устройство – полуссуматор, таблица истинности которого представлена ниже. *Таблица 1.7.*

| *Входы* | | *Выходы* | |
|---|---|---|---|
| *A* | *B* | *S* | *PO* |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

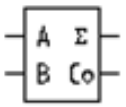
На рис. 1.18 представлена соответствующая схемная реализация полусумматора на базе логических элементов (а) и в виде одного устройства (б).

Схема полусумматора формирует перенос в следующий разряд (*PO*), но не может учитывать перенос из предыдущего разряда, поэтому она и называется полусумматором. Для реализации же полного суммирования (пусть пока и одноразрядного) необходимо помимо формирования переноса в следующий разряд учитывать еще и перенос из предыдущего разряда (это нужно для формирования многоразрядных сумматоров).




*Обозначение принятое в нашей стране*




*Обозначение принятое за рубежом*

а) Расширенная схема б) Условное обозначение

*Рис. 1.18.* Схема полусумматора

*3. Одноразрядные сумматоры.* Таблица истинности полного двоичного одноразрядного сумматора приведена ниже.

*Таблица 1.8.* Таблица истинности одноразрядного сумматора

| *Входы* | | | *Выходы* | |
|---|---|---|---|---|
| *PI* | *A* | *B* | *S* | *PO* |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Здесь, помимо формирования переноса в следующий разряд (*PO*) учитывается еще и перенос из предыдущего разряда (*PI*).

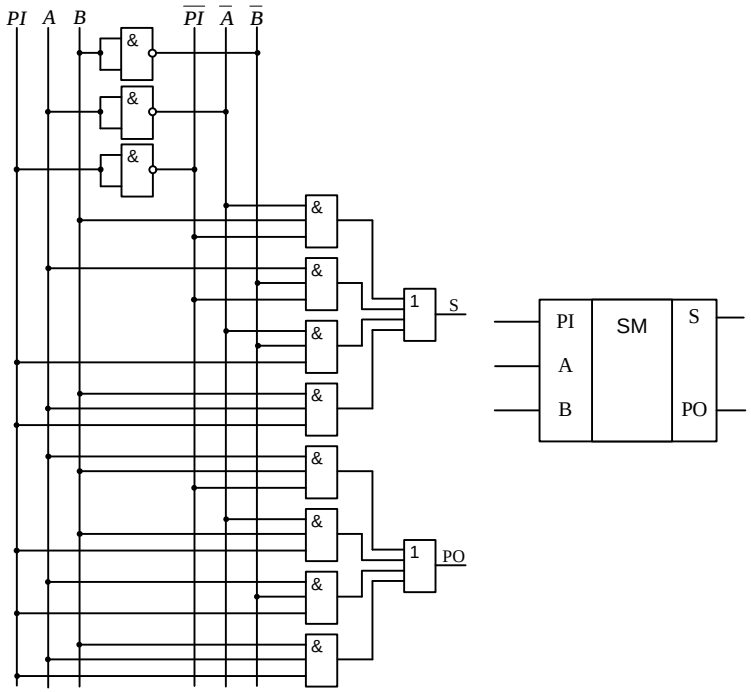
На рис. 1.19 представлена соответствующая схемная реализация сумматора на базе логических элементов (а) и в виде одного устройства (б).

## Многоразрядные сумматоры

Для того чтобы получить многоразрядный сумматор, необходимо соединить входы и выходы переносов соответствующих двоичных разрядов. Схема соединения для трехразрядного сумматора приведена на рисунке 1.20. Здесь же приведено условное обозначение данного сумматора, применяемого на схемах.

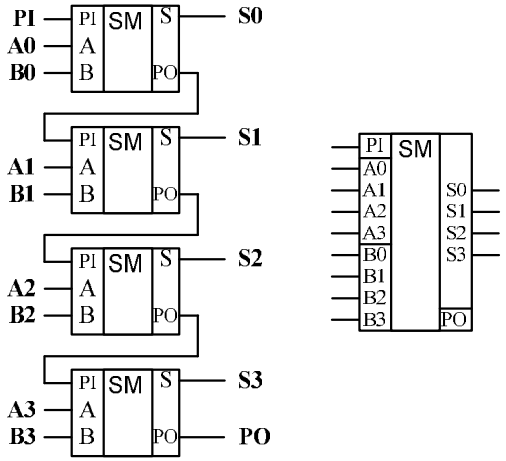
Схема на рис. 1.20 составлена согласно принципам работы. На практике, для увеличения скорости работы применяется отдельная схема формирования переносов для каждого двоичного разряда.






а) Расширенная схема б) Условное обозначение

*Рис. 1.19.* Схема двоичного одноразрядного сумматора




а) Расширенная схема б) Условное обозначение

*Рис. 1.20.* Схема полного двоичного трехразрядного сумматора

## Вычитатели

*Одноразрядный полувычитатель* – цифровое устройство, на входы которого поступают два одноразрядных числа *A* и *B*, а на выходе формируются одноразрядная разность *D* и заем *V*. Таблица истинности имеет вид 1.9.

*Таблица 1.9.* Таблица истинности полуразрядного вычитателя

| Входы | | Выходы | |
|---|---|---|---|
| $A_i$ | $B_i$ | $D_i$ | $V_i$ |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| A-B | | | |

По таблице можно составить логические функции для разницы *D*, заема *V*, и схему, показанную на рис. 1.21.

$$D_i = \overline{A_i} \cdot B_i + A_i \cdot \overline{B_i} = A_i \oplus B_i \qquad V_i = \overline{A_i} \cdot B_i$$

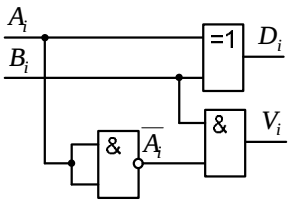



*Рис. 1.21*

Возможны различные реализации полувычитателей на основании тождественных преобразований полученных функций. Схема полувычитателя (см. рис. 1.20) отличается от схемы полусумматора только наличием инвертора по сигналу *A*.

Полный вычитатель (см. табл. 1.10 и рис. 1.21) это устройство, реализующее операцию вычитания двух одноразрядных чисел с формированием сигнала заема из старшего разряда и учета заема из предыдущего разряда.

Логическая функция для разности имеет вид

$$\begin{aligned} D_i &= \overline{A_i} \cdot \overline{B_i} \cdot V_{i-1} + \overline{A_i} \cdot B_i \cdot \overline{V}_{i-1} + A_i \cdot \overline{B_i} \cdot \overline{V}_{i-1} + A_i \cdot B_i \cdot V_{i-1} = \\ &= V_{i-1}\left(\overline{A_i}\,\overline{B_i} + A_i B_i\right) + \overline{V}_{i-1}\left(\overline{A_i} B_i + A_i \overline{B_i}\right) = \\ &= V_{i-1}\overline{\left(A_i \oplus B_i\right)} + \overline{V}_{i-1}\left(A_i \oplus B_i\right) = V_{i-1} \oplus (A_i \oplus B_i) \end{aligned}$$



*Таблица 1.10.* Таблица истинности полного вычитателя

| Входы | | | Выходы | |
|---|---|---|---|---|
| $A_i$ | $B_i$ | $V_{i-1}$ | $D_i$ | $V_i$ |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| $A_i$-$B_i$-$V_{i-1}$ | | | | |

Логическая функция для заема

$$\begin{aligned} V_i &= \overline{A_i} \cdot \overline{B_i} \cdot V_{i-1} + \overline{A_i} \cdot B_i \cdot \overline{V}_{i-1} + \overline{A_i} \cdot B_i \cdot V_{i-1} + A_i \cdot B_i \cdot V_{i-1} = \\ &= \overline{A_i} \cdot \overline{B_i} \cdot V_{i-1} + \overline{A_i} \cdot B_i \cdot (\overline{V}_{i-1} + V_{i-1}) + A_i \cdot B_i \cdot V_{i-1} = \\ &= \overline{(A_i \cdot \overline{B_i} + A_i \cdot B_i)} \cdot V_{i-1} + \overline{A_i} \cdot B_i = \\ &= \overline{(A_i \oplus B_i)} \cdot V_{i-1} + \overline{A_i} \cdot B_i \end{aligned}$$

Составим схему вычитателя (см. рис. 1.22).

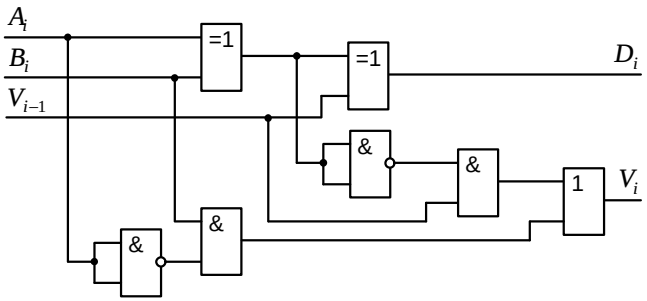



*Рис. 1.22.* Схема полного двоичного одноразрядного вычитателя

Соединяя друг с другом полувычитатели и вычитатели получают устройства, называемые параллельным вычитателями.

Для упрощения схемной реализации вычислительных устройств было бы удобно иметь одно универсальное устройство на базе сумматоров и вычитателей, реализующих операцию сложения и вычитания. Оказывается, что использование простых математических приёмов позволяет приспособить сумматоры для выполнения операции вычитания.

Один из таких приемов – *сложение в системе с обратным кодом*. В этой системе вычитаемое представляется в обратном коде (все разряды кроме знакового инвертируются). Далее между уменьшаемым и обратным кодом вычитаемого производят операцию сложения. Ошибку в единицу устраняют путем использования дополнительной микросхемы и входа переноса. Для инвертирования числа можно использовать «исключающее ИЛИ». Простейшая схемная реализация такого универсального устройства показана на рис. 1.23. Здесь если на вход *F* подается единица, то код числа «*B*3 *B*2 *B*1 *B*0» инвертируется. Если *F* = 0, то схема работает как обычный сумматор.

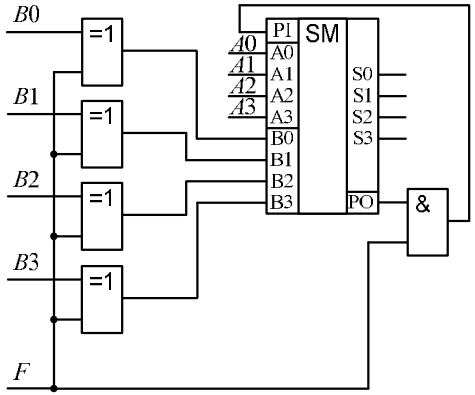



*Рис. 1.23*

Сложение/вычитание в обратном коде

Такой сумматор-вычитатель часто применяется в параллельных двоичных машинах в основном из-за простоты получения обратного кода. К недостаткам следует отнести малое быстродействие из-за наличия циклического переноса.

В вычислительных машинах наиболее часто применяют *сложение в системе с дополнительным кодом*. Здесь вычитаемое переводится в дополнительный код. Схема такого устройства идентична 1.21, но здесь нет коррекции ошибки в младшем разряде.



*Пороговая ячейка*

Составим логическую цепь трехвходовой пороговой ячейки, сигнал на выходе которой будет равен «1»; когда на ее входах присутствует не менее двух единиц.

| | $X_1$ | $X_2$ | $X_3$ | F |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 1 |
| 5 | 1 | 0 | 0 | 0 |
| 6 | 1 | 0 | 1 | 1 |
| 7 | 1 | 1 | 0 | 1 |
| 8 | 1 | 1 | 1 | 1 |

1. Составим таблицу функционирования.

2. Для составления логической функции воспользуемся СДНФ получим:

$$F = \overline{X}_1 \cdot X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + \\ + X_1 \cdot X_2 \cdot \overline{X}_3 + X_1 \cdot X_2 \cdot X_3.$$

3. Минимизируем полученное выражение. Применим для членов 3 и 4 закон склеивания

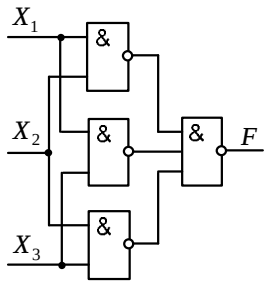
$$F = \overline{X}_1 \cdot X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + X_1 \cdot X_2 \cdot (\overline{X}_3 + X_3) = \\ = \overline{X}_1 \cdot X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + X_1 \cdot X_2.$$

Умножим последнее слагаемое на $(1 + X_3)$ т.к. это не изменит общего выражения, и повторим предыдущую процедуру с первым и последним членами

$$F = \overline{X}_1 \cdot X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + X_1 \cdot X_2 \cdot (1 + X_3) = \\ = \overline{X}_1 \cdot X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + X_1 \cdot X_2 + X_1 \cdot X_2 X_3 = \\ = X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + X_1 \cdot X_2.$$

Вновь умножим третье слагаемое на $(1 + X_3)$ и применим для слагаемых 2 и 4 закон поглощения, получим минимизированную функцию

$$F = \overline{X}_1 \cdot X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + X_1 \cdot X_2 \cdot (1 + X_3) = \\ = X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + X_1 \cdot X_2 \cdot (1 + X_3) = \\ = X_2 \cdot X_3 + X_1 \cdot \overline{X}_2 \cdot X_3 + X_1 \cdot X_2 + X_1 \cdot X_2 \cdot X_3 = \\ = X_2 \cdot X_3 + X_1 \cdot X_2 + X_1 \cdot X_3.$$




4. Приведем функцию к виду, удобному для реализации на элементах И-НЕ. Для этого возьмем двойное отрицание и применяя правило Моргана:

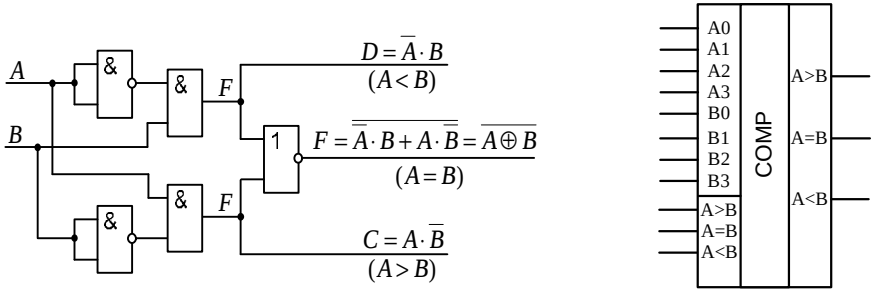
$$F = \overline{\overline{X_2 \cdot X_3 + X_1 \cdot X_2 + X_1 \cdot X_3}} = \overline{\overline{X_2 \cdot X_3} \cdot \overline{X_1 \cdot X_2} \cdot \overline{X_1 \cdot X_3}}.$$

5. Составим схему.

## *Компараторы*

Компараторы предназначены для сравнения двух чисел, заданных в двоичном коде. В зависимости от схемного исполнения компараторы могут определять равенство *A=B* или вид неравенства: *A<B* или *A>B*. Результат сравнения отображается соответствующим логическим уровнем на выходе. Цифровые компараторы, как правило, выполняют все эти операции и имеют три выхода. Используются для выявления нужного числа в потоке цифровой информации, для отметки времени в числовых приборах, для выполнения условных переходов в вычислительных устройствах и т.д.

Схема одноразрядного компаратора показана на рис. 1.24 (а)




а) б)

*Рис. 1.24.* Схема одноразрядного компаратора

Промышленность выпускает четырёхразрядные цифровые компараторы (например К561ИП2), условное графическое изображение которых показано на рисунке 1.24 (б).

*Два двоичных числа равны*, когда попарно равны между собой все разряды этих чисел. Если *A* и *B* – четырехразрядные, то признаком их равенства будет

A3=B3; A2=B2; A1=B1, A0=B0.

*Неравенство A>B* обеспечивается в четырех случаях:

1. A3>B3;
2. A3=B3, A2>B2;



3. A3=B3, A2=B2, A1>B1;
4. A3=B3, A2=B2, A1=B1, A0>B0.

*Условие B > A* выполняется просто, достаточно в указанных соотношениях B и A поменять местами.

Микросхема имеет входы каскадирования A>B, A=B и A<B, которые позволяют наращивать разрядность обоих чисел без дополнительных логических элементов. Компараторы можно соединять каскадно и пирамидально.

При каскадном соединении на входы первой микросхемы A=B подается $U^1$, а A<B, и A>B – $U^0$, выходы предыдущей микросхемы соединяются с входами последующей, сравниваемые коды подаются на входы $A_i$ и $B_i$.

На входы каскадирования при пирамидальном соединении компараторов подаются сигналы: A=B$\rightarrow U^1$, A>B и A<B$\rightarrow U^0$. Выходы микросхемы, на вход которой подаются младшие разряды сравниваемых кодов, соединяются с одноименными входами микросхемы второй ступени.

Выходы A<B и A>B остальных микросхем первой ступени соединяют с входами $A_i$ и $B_i$ микросхемы второй ступени.

## *Последовательностные схемы*

*Последовательностные цифровые устройства* часто называют последовательностными схемами, последовательностными автоматами, дискретными автоматами с памятью, многотактными автоматами.

Кроме комбинационных устройств, рассмотренных в предыдущих разделах, существует класс цифровых устройств, в которых при одинаковых воздействиях на входе, на выходе автомата могут возникать различные выходные состояния.

Состояние выхода такого устройства зависит не только от того, какие сигналы присутствуют на его входах в данный момент времени, но и от того, какие последовательности сигналов поступали на входы устройства в *предшествующие моменты времени* (внутреннее состояние схемы) Т.е. как говорят –

цифровой автомат имеет *память*. Поэтому такие устройства называют последовательностными или многотактными автоматами.

*Последовательностным цифровым устройством* называется цифровой автомат, состояние выходов которого определяется совокупностью комбинацией входных значений и его предшествующим состоянием.

Для описания последовательностного автомата с памятью, помимо состояний входов $X(t)$ и выходов $Y(t)$, необходимо также знать его внутреннее состояние $S(t)$. В общем виде, последовательностный автомат можно рассмотреть из двух частей: комбинационного устройства (КУ) и памяти, состоящей из элементов памяти (ЭП) (рис. 1.25).

*Рис. 1.25.*
Схема последовательностного автомата

В качестве элементов памяти могут быть применены как однобитовые элементы памяти (различные типы триггеров), так и многобитовые (многоразрядные) цепочки триггеров.

Функционирование (т.е. изменение состояния устройства) многотактного автомата происходит в дискретные моменты времени, ход которого обозначается натуральными числами $t$ = 1, 2, 3 и т.д. В каждый момент дискретного времени $t$ автомат находится в определенном состоянии $S(t)$, воспринимает через входы соответствующую данному моменту комбинацию входных переменных $X(t)$, выдает на выходах некоторую функцию выхода $Y(t)$, определяемую как

$$Y(t) = f\left(S(t), X(t)\right)$$

и переключается в новое состояние $S(t+1)$, которое определяется *функцией переходов* φ как

$$S(t+1) = j\left(S(t), X(t)\right)$$

Закон функционирования последовательностных автоматов может задаваться в виде уравнений, таблиц, графов и временных диаграмм.



*Под законом функционирования* понимается совокупность правил, описывающих последовательность переключения состояний автомата и последовательность выходных сигналов в зависимости от последовательности поступления входных сигналов.

ЭП является простейшим типом *триггера*, реализованного с помощью элементов базиса И-НЕ или ИЛИ-НЕ с соответствующими обратными связями и позволяет хранить один бит цифровой информации.

*Триггер* – это последовательностная схема с двумя состояниями, каждое из которых при определенных условиях поддерживается постоянным (т.е. стабильным). Каждому из этих состояний ставится в соответствие логическое значение, которое “хранит” триггер (если на выходе триггера высокий уровень напряжения - “1” и “0” - в противном случае).

Вообще говоря, у триггера должна быть по крайней мере одна выходная линия, представляющая логическое значение, соответствующее состоянию триггера. Когда на выходной линии логическая «1», говорят, что триггер *установлен*, в противном случае говорят, что триггер сброшен. Триггер имеет несколько входных линий, сигналы на которых (вместе с текущим состоянием триггера) определяют следующее состояние триггера. Тип триггера определяется алгоритмом его работы. В зависимости от алгоритма триггер может иметь установочные, информационные и управляющие входы. Установочные входы служат для перевода триггера в одно из определенных состояний, независимо от состояния других входов. Входы управления разрешают запись данных, подающихся на информационные входы. Для дальнейшего рассмотрения цифровых устройств последовательностного типа введем еще один способ описания цифрового автомата.

## Временная диаграмма

*Временная диаграмма* в общем случае представляет собой график, по оси абсцисс которой откладывается время в тактах, а по оси ординат необходимая

цифровая величина (вход или выход цифровой схемы), принимающее значения «0» или «1».

Если у цифрового устройства сигнал синхронизации отсутствует (частный случай), то за каждый такт принимают очередное изменение входных сигналов.

Необходимо отметить, что чаще всего осей ординат несколько (по количеству входных переменных), и они располагаются друг под другом. Собственно говоря, именно в этом и состоит основное преимущество применения временных диаграмм, работа цифрового устройства наглядно представляется во времени (см. рис. 1.26.а).

В случае наличия сигнала синхронизации значение входных переменных должны быть установлены между импульсами синхронизации, а выход цифрового автомата примет свое новое устойчивое состояние либо по переднему, либо по заднему фронту импульса синхронизации (рис. 1.26.б).

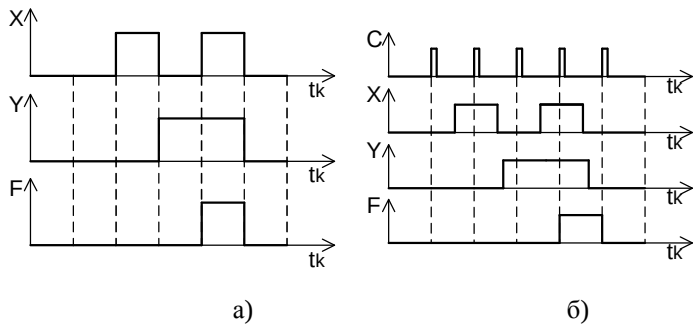



*Рис. 1.26.* Временная диаграмма

Рассмотренный выше случай работы цифрового устройства является идеальным. Т. е. в данном случае не учитывались реальные характеристики элементов, такие как сопротивление, конечное время установления сигнала, задержка прохождения импульса и т. д.

Рассмотрим кратко общую теорию синхронизации (так называемые гонки в микросхемах).



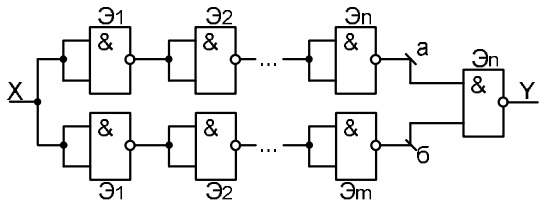
*Общая теория синхронизации.*

*Гонки в комбинационных устройствах (КУ).*

Если КУ выполнено на базе идеальных, т.е. безынерционных элементов, состояние выходов однозначно определяется состоянием входов в тот же момент времени. Однако, инерционность элементов и наличие различных факторов, приводящих к задержке распространения сигнала, приводят к задержке появления выходных сигналов КУ, т.е. сигналы на выходе КУ, соответствующие новому состоянию входных сигналов, появляются не сразу, а с некоторой задержкой. При этом в переходный период возможно появление на выходах устройства некоторых промежуточных значений сигналов, не соответствующих заданному состоянию устройства.

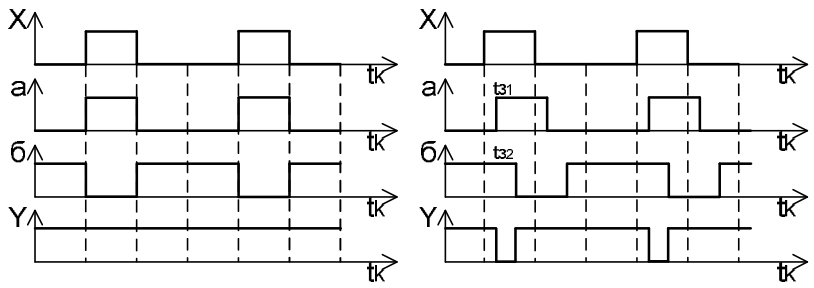
Такое явление получило название *состязаний* или *гонок*. Обычно, вырабатываемые узлами КУ промежуточные значения сигналов, представляют собой импульсы очень малой длительности, являющиеся помехой для всей цифровой системы. Они могут запускать непредусмотренное срабатывание триггеров, счетчиков и осуществлять нежелательные записи в регистры.

Рассмотрим в качестве примера фрагмент схемы комбинационного устройства (рис. 1.27), где может наблюдаться явление гонок. Для наглядности процесса формирования промежуточного значения выходного сигнала приведены временные диаграммы состояний различных цепей распространения в идеальном и реальном случаях (рис. 1.28).




*n – четное; m – нечетное;*

*Рис. 1.27.* Схема иллюстрирующая возможность появления гонок




Идеальный случай Реальный случай (m>n)

*Рис. 1.28.* Временная диаграмма, иллюстрирующая проявление гонок

Время задержки импульсов в цепях определяется средним временем задержки распространения сигнала всеми элементами этой цепи. Момент времени появления импульса помехи определяется соотношением числа инвертирующих элементов в конкурирующих цепях фрагмента схемы КУ.

Как следует из рис. 1.28, если элементы схемы идеальные, т. е. безынерционные, (что на практике достичь не удается), на выходе схемы КУ импульс помехи отсутствует. Однако в реальных схемах всегда имеет место явление гонок и требуется создать такие схемы, в которых влияние этого явления устраняется.

*Борьба с гонками*

Существует три наиболее часто встречающихся способа борьбы с гонками: тактирование; построение противогоночных схем; учет минимального времени задержки распространения сигнала.

Пример реализации тактирования может быть получен из схемы 1.27. Для этого достаточно в нее ввести дополнительный сигнал «С», который подать на третий вход последнего элемента И-НЕ. Тогда схема будет работать по факту прихода сигнала «С = 1». Время поступления выбирается в ходе проектирования.

Перейдем к простейшим типам триггерных схем.



*RS-триггер*

*RS-триггер* – простейший автомат с памятью, который может находиться в двух состояниях. Триггер имеет два установочных входа: установки *S* (*set* – установка) и сброса *R* (*reset* – сброс), на которые подаются входные сигналы от внешних источников.

При подаче на вход установки активного логического уровня триггер устанавливается в «1» ($Q=1, \overline{Q}=0$), при подаче активного уровня на вход сброса триггер устанавливается в «0» ($Q=0, \overline{Q}=1$). Если подать на оба входа установки и сброса пассивный уровень, то триггер будет сохранять предыдущее состояние выходов: $Q=1, \overline{Q}=0$ либо $Q=0, \overline{Q}=1$. Каждое состояние устойчиво и поддерживается за счет действия обратных связей.

Для триггеров этого типа является недопустимой одновременная подача активного уровня на оба входа установки и сброса, т. к. триггер по определению не может одновременно быть установлен в ноль и единицу. На практике, подача активного уровня на установочные входы приводит к тому, что это состояние не может быть сохранено и невозможно определить, в каком состоянии будет находиться триггер при последующей подаче на установочные входы сигналов пассивного уровня.

На рис. 1.29 показан *RS-триггера*, выполненных на элементах «ИЛИ-НЕ», имеющий активный уровень по входам «1»; а на рис. 1.30 показан $\overline{R}\overline{S}$-*триггер*, выполненный на базе «И-НЕ» с активным уровнем по входам «0».

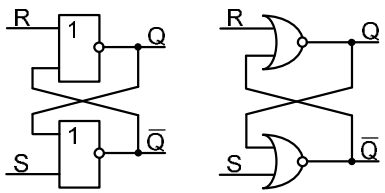



*Рис. 1.29. RS-триггер*

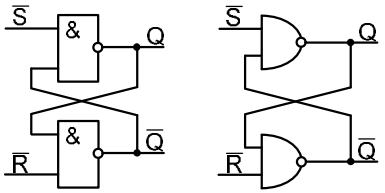



*Рис. 1.30.* $\overline{R}\overline{S}$ *-триггер*

*RS-триггер* является основным узлом для построения последовательных схем. Так, если в *RS-триггере* (рис. 1.29) вначале установить комбинацию *R=0, S=1*, а потом перейти к *R=0, S=0*, то состояние выхода будет *Q=1*. Если же вначале установить комбинацию *R=1, S=0*, а потом перейти к *R=0, S=0*, то состояние выхода будет другим – *Q=0*. Таким образом, при одном и том же входном наборе *R=0, S=0* выход триггера может находиться в разных состояниях.

Условия переходов триггеров из одного состояния в другое (алгоритм работы) можно описать табличным, аналитическим или графическим способами. Табличное описание работы *RS-триггера* (рис. 1.29) представлено в таблице 1.11 (таблица переходов и таблица функций возбуждения).

*Таблица 1.11.* Таблица переходов и функций возбуждения *RS-триггера*

| *Таблица переходов* | | | *Таблица функций возбуждений* | | | |
|---|---|---|---|---|---|---|
| *R* | *S* | $Q_{t+1}$ | $Q_t$ | $Q_{t+1}$ | *R* | *S* |
| 0 | 0 | $Q_t$ | 0 | 0 | - | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | x | 1 | 1 | 0 | - |

В таблицах использованы следующие обозначения: «$Q_t$» – предшествующее состояние выхода; «$Q_{t+1}$» – новое состояние, устанавливающееся после перехода (возможно $Q_{t+1} = Q_t$); «-» – безразличное значение сигнала: «0» или «1»; «x» – запрещенное состояние.

Аналитическое описание (характеристическое уравнение) можно получить из таблицы 1 по правилам алгебры логики.

$$Q_{t+1} = \overline{R} \cdot S + \overline{R} \cdot \overline{S} \cdot Q_t = \overline{R}(S + \overline{S} \cdot Q_t) = \overline{R}(S + Q_t)$$

Зависимость $Q_{t+1}$ от $Q_t$ характеризует свойство запоминания предшествующего состояния.



Схема триггера позволяет запоминать состояние логической схемы, но так как в начальный момент времени может возникать переходный процесс (в цифровых схемах этот процесс называется «опасные гонки»), то запоминать состояния логической схемы нужно только в определённые моменты времени, когда все переходные процессы закончены. Схемы созданные без учета сигнала синхронизации называют ***асинхронными***. Случай, когда длительности управляющих сигналов не синхронизированы (не согласованы), изображен на рис. 1.31. Здесь же показано что будет при подаче запрещенной комбинации, и чем она опасна (после подачи «пассивной комбинации»).

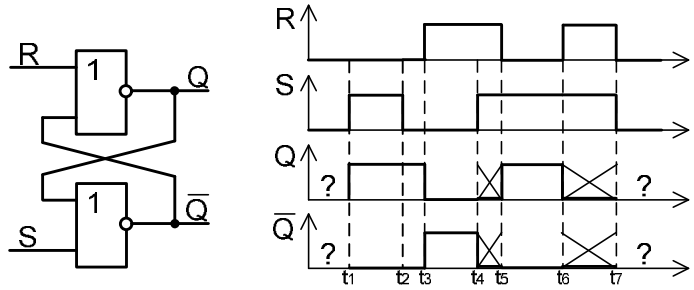
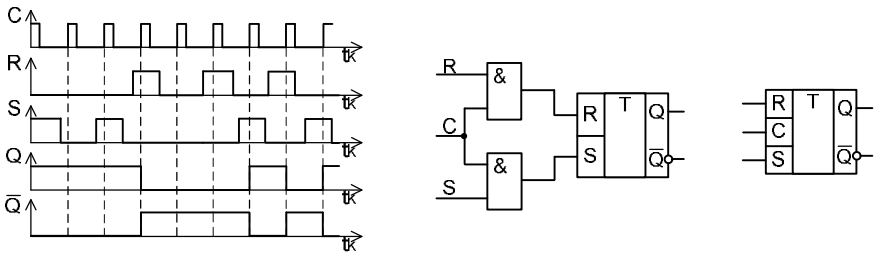



*Рис. 1.31.* Функционирование асинхронного *RS-триггера*

Триггер, построенный на базе элементов ИЛИ-НЕ, называют также дизъюнктивной бистабильной ячейкой. Бистабильные ячейки, помимо самостоятельного применения, входят в качестве составного узла в триггеры других типов.

Синхронный RS -триггер

Синхронные триггеры снабжаются дополнительным входом, по которому поступает синхронизирующий (тактирующий) сигнал. При этом изменение состояния триггера происходит (при наличии управляющего сигнала) только в те моменты времени, когда на специальный синхровход триггера поступает тактирующий импульс (рис 1.32.а). Синхронный *RS-триггер* строится в соответствии с (рис 1.32.б) а его условное изображение на принципиальных и функциональных схемах приведено на рис. 1.32в. Синхронизирующий вход обозначается С.




а) б) в)

*Рис. 1.32.* Синхронный триггер

Входные сигналы *S* и *R* являются информационными, а на входе *C* – синхронизирующими, по ним происходит переключение триггера. Следует отметить, что для надежной работы триггера необходимо, чтобы длительность переключающего сигнала (синхронизирующего сигнала) на входе *C* была не меньше времени переключения триггера. Временем переключения (срабатывания, установки) триггера называется время, которое проходит от момента изменения входных сигналов до соответствующего изменения состояния выходов и определяющееся задержками распространения сигнала логическими элементами, входящими в состав триггера.

## Двухступенчатый RS-триггер

Рассмотренные схемы *RS-триггеров* являются одноступенчатыми. Применение одноступенчатых RS-триггеров в качестве самостоятельных запоминающих элементов ограничено. Это связано с неустойчивой работой последовательностной схемы (цифрового автомата), память которой выполнена на одноступенчатых *RS-триггерах*. Сигналы переключения триггера $S(t)$, $R(t)$ формируются в цифровом автомате комбинационной схемой, в их формировании участвуют, наряду с внешними логическими сигналами, сигналы $Q(t)$ и $\overline{Q}(t)$. Для устойчивой работы триггера необходимо, чтобы сигналы $Q(t)$ и $\overline{Q}(t)$ изменялись только после прекращения действия входного сигнала $S(t)$ или $R(t)$. Это требование выполняется в двухступенчатых триггерах (MS-триггерах).



*Двухступенчатый триггер* состоит из двух секций (ступеней), соединенных каскадно, как показано на рис. 1.33.а, причем, каждая секция содержит по синхронному *RS-триггеру*. Первая секция, ведущая или М-секция (М происходит от английского MASTER) принимает информацию с входных линий *S* и *R*. Состояние выходов ведущей секции подается на вторую секцию, ведомую, или S-секцию (S происходит от английского SLAVE).

Для ведущего триггера используется обычная синхронизация, в то время как для ведомого триггера импульс синхронизации инвертируется. Изменение состояния выхода ведущего триггера будет происходить в момент появления положительного импульса синхронизации, и эти изменения будут переданы на входы ведомого триггера. Однако, никакие изменения на выходе ведомого триггера не будут происходить до тех пор, пока не появится положительный сигнал инвертированного импульса синхронизации, т.е. отрицательный (задний фронт) фронт исходного синхроимпульса. Следовательно, изменения на выходах $Q(t)$ и $\overline{Q}(t)$ не произойдет до тех пор, пока не завершится импульс синхронизации.

На рис. 1.33.б показаны временные диаграммы работы такого триггера. На функциональных схемах двухступенчатый триггер изображается в соответствии с рис. 1.33.в. Символ *TT* в поле условного обозначения означает , что триггер двухступенчатый.

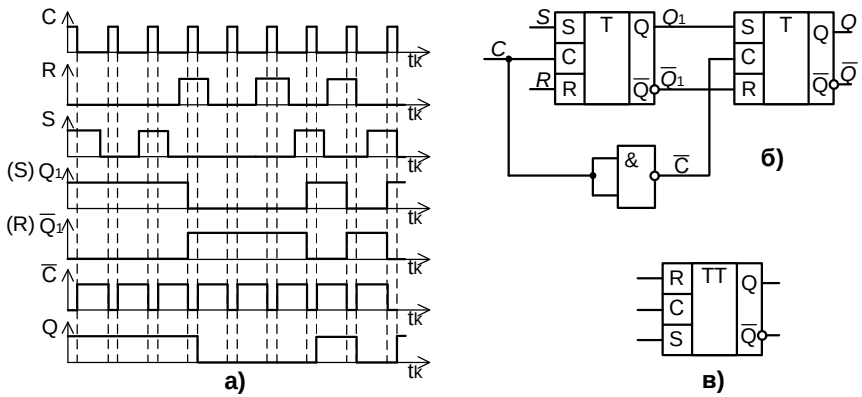



*Рис. 1.33* – Двухступенчатый *RS-триггер*

## *Триггеры с динамическим управлением*

Динамические триггеры могут опрокидываться как передним, так и задним фронтом тактирующих импульсов. Фрагменты схемного обозначения приведены на рис. 1.34. В англоязычной литературе принято в случае синхронизации задним фронтом обозначать соответствующий вход с инверсией.

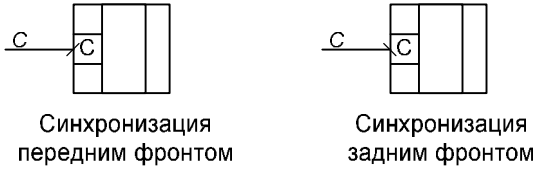



*Рис. 1.34.* Фрагменты обозначения линий синхронизации

## *JK-триггер*

Триггер JK-типа имеет более сложную по сравнению с *RS-триггером* структуру и более широкие функциональные возможности.

*JK-триггер* – цифровой автомат с памятью, который может находиться в двух устойчивых состояниях и имеет два информационных входа: установки *J* и сброса *K*, на которые подаются входные сигналы.

Помимо информационных входов *J* и *K* и прямого и инверсного выходов $Q(t)$ и $\overline{Q}(t)$, *JK-триггер* имеет вход управления *C* (этот вход также называют тактирующим или счетным), а также асинхронные установочные *R* и *S*. Обычно активными уровнями установочных сигналов являются нули, как в схеме *R'S'-триггера*. Установочные входы имеют приоритет над остальными. Активный уровень сигнала на входе *S* устанавливает триггер в состояние *Q=1*, а активный уровень сигнала на входе *R* – в состояние *Q=0*, независимо от сигналов на остальных входах.

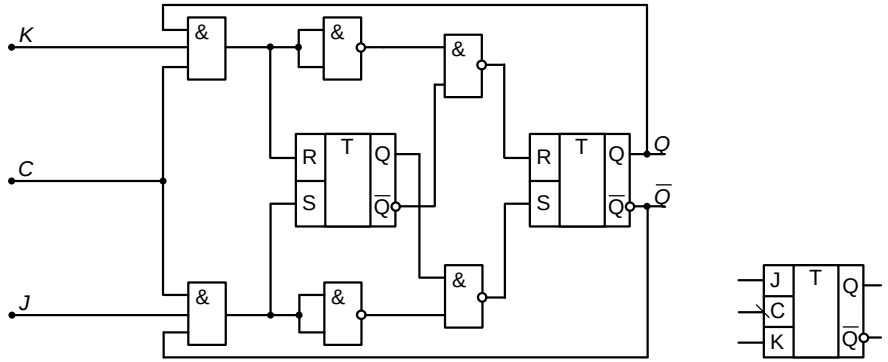
Если на входы установки одновременно подать пассивный уровень сигнала, то состояние триггера будет изменяться по фронту импульса на входе *C* в зависимости от состояния входов *J* и *K*, как показано в таблице 1.12.



*Таблица 1.12.* Таблица переходов и возбуждения *JK-триггера*

| *Таблица переходов* | | | *Таблица функций возбуждений* | | | |
|---|---|---|---|---|---|---|
| $J$ | $K$ | $Q_{t+1}$ | $Q_t$ | $Q_{t+1}$ | $K$ | $J$ |
| 0 | 0 | $Q_t$ | 0 | 0 | x | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | $\overline{Q}_t$ | 1 | 1 | 0 | x |

Один из вариантов функциональной схемы *JK-триггера* и его условное графическое обозначение приведены на рис. 1.35, временные представлены на рис. 1.36. В данном случае все изменения выхода происходят только в момент отрицательного перепада тактового сигнала.




1) Расширенная схема 2) Условное обозначение

*Рис. 1.35.* Схема *JK-триггера*

Характеристическое уравнение *JK-триггера* имеет вид

$$Q_{t+1} = \overline{K} \cdot Q_t + J \cdot \overline{Q}_t$$

Выпускаемые промышленностью ИМС *JK-триггеров* могут иметь несколько входов (до трех) *J* и такое же количество входов *K*, объединенных схемами конъюнкций. Помимо этого как отмечалось выше, схемы триггеров снабжаются асинхронными установочными входами типа *R* и *S*.

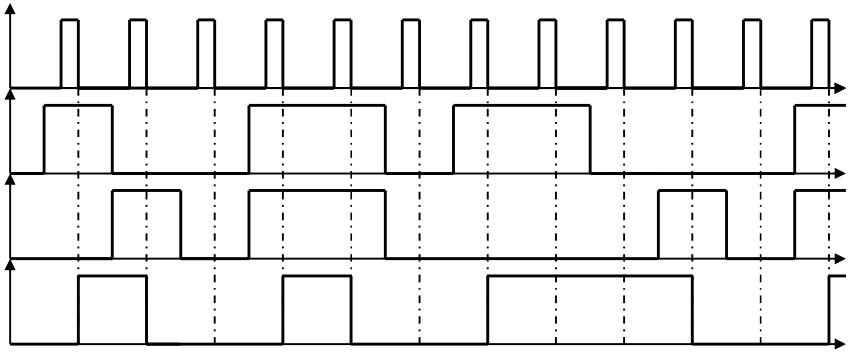


*Рис. 1.36.* Временная диаграмма синхронного *JK-триггера*

## *D-триггер*

*D-триггер* – цифровой автомат с памятью, который может находиться в двух устойчивых состояниях и имеет один информационный вход *D* (*data* – данные или *delay* – задержка).

Информация с входа *D* заносится в триггер по перепаду импульса на счетном входе *C* и сохраняется до следующего перепада на счетном входе триггера. Помимо счетного *C* и информационного *D* входов схему триггера снабжают асинхронными установочными *R* и *S* входами. Установочные входы имеют наивысший приоритет. Они устанавливают триггер независимо от сигналов на входах *C* и *D*. *D-триггер* – бывает только синхронным.

Функционирование *D-триггера* описывается таблицей 1.13 переходов и возбуждения и диаграммами рис. 1.37.

*Таблица 1.13.* Таблица переходов и возбуждения *D-триггера*

| *Таблица переходов* | | *Таблица функций возбуждений* | | |
|---|---|---|---|---|
| $D$ | $Q_{t+1}$ | $Q_t$ | $Q_{t+1}$ | $D$ |
| 0 | 0 | 0 | 0 | 0 |
| | | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| | | 1 | 1 | 1 |



Характеристическое уравнение D-триггера $Q_{t+1} = D_t$ .

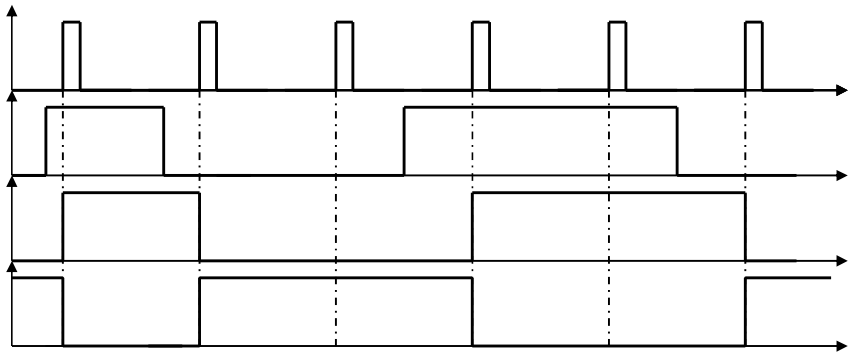
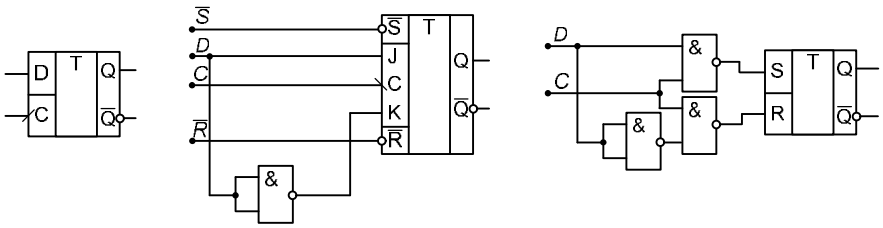


*Рис. 1.37.* Временная диаграмма *D-триггера*

Уравнение триггера показывает, что состояние триггера на *t+1* такте равно входному сигналу в момент, предшествующий тактовому перепаду сигнала *C*. Одно из условных обозначений *D-триггера* представлено на рис 1.38.а. Функциональная схема *D-триггера* может быть получена из схемы *JK-триггера* путем подключения входа *D* к входу *J* через инвертор (рис. 1.38.б) или на базе асинхронных триггеров как показано на рис. 1.38.в.




а) б) в)

*Рис. 1.38* – Схемы *D-триггера*

## Т-триггер (счетный триггер)

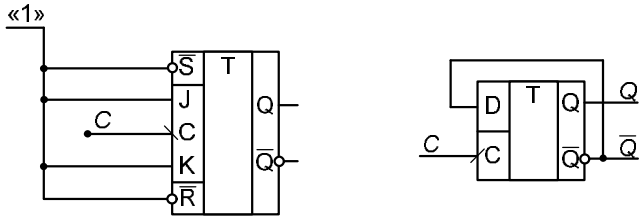
На основе *JK-триггеров* и *D-триггеров* можно построить схемы, осуществляющие так называемый счетный режим. Такие схемы называют *T-*

*триггерами* или счетными триггерами, что связано со способом их функционирования.

*Т-триггер* – цифровой автомат с памятью, который может находиться в двух устойчивых состояниях и имеет один информационный вход *Т*.

На рис. 1.39 представлены схемы организации *Т-триггера* на основе *JK-триггера* (рис. 1.39.а) и *D-триггера* (рис. 1.39.б).

Счетный режим иллюстрируется временными диаграммами рис. 1.40. В соответствии с таблицей функционирования при каждом отрицательном перепаде входного сигнала *Т* состояние триггера изменяет свое значение на противоположное.




а) На базе *JK-триггера* б) На базе *D-триггера*

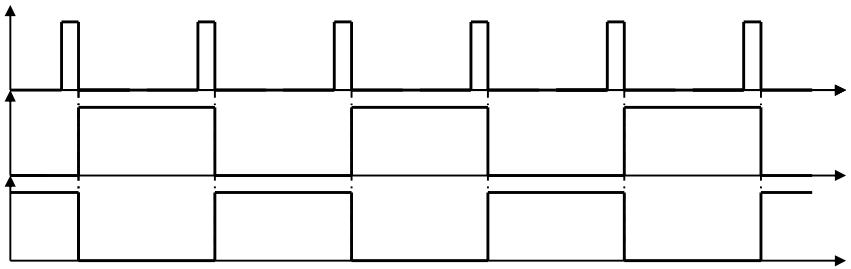
*Рис. 1.39* – Схемы *Т-триггера*

Таким образом, на каждые два входных тактовых импульса *Т-триггер* формирует один период выходного сигнала *Q*. Следовательно, триггер осуществляет деление частоты $f_T$ на его входе на два:

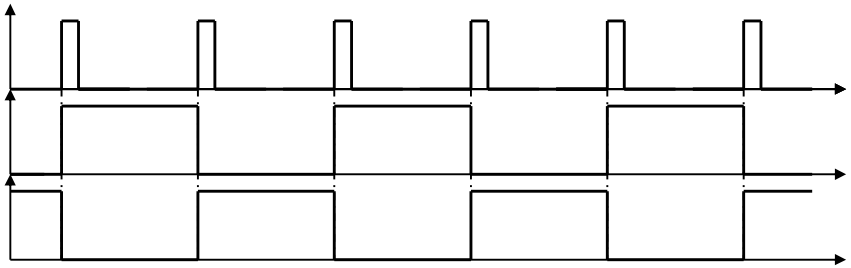
$$f_Q = \frac{f_T}{2}$$

Если один счетный триггер позволяет делить частоту на два, то для реализации делителя частоты на четыре потребуется два триггера, соединенных последовательно и т.д.





1) на базе *JK-триггера*



2) на базе *D-триггера*

*Рис. 1.40.* Временные диаграммы *Т-триггера*

## *Регистры*

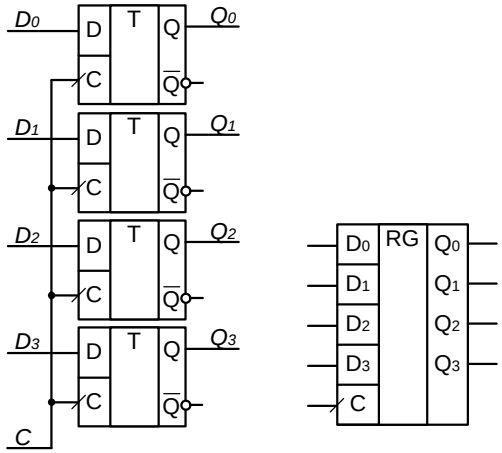
Наиболее распространенным узлом цифровой техники и устройств автоматики являются регистры. Регистры строятся на базе синхронных одно- и двухступенчатых *RS* и *D-триггеров*. Регистры могут быть реализованы также на базе *JK-триггеров*.

*Регистром* называется последовательное или параллельное соединение триггеров.

По способу приема и выдачи информации регистры делятся на следующие группы: с параллельным приемом и выдачей; с последовательным приемом и выдачей; с последовательным приемом и параллельной выдачей; с параллельным приемом и последовательной выдачей; комбинированные, с различными способами приема и выдачи и реверсивные.

*1. Параллельный регистр (регистр памяти)* служит для запоминания многоразрядного двоичного слова. При использовании для построения параллельного регистра *триггеров-защелок* регистр называется *регистр-защелка*. Количество триггеров, входящее в состав параллельного регистра определяет его разрядность. При записи информации в параллельный регистр все биты (двоичные разряды) записываются одновременно.

Схема четырёхразрядного параллельного регистра приведена на рисунке 1.41.1, а его обозначение на принципиальных схемах – на рисунке 1.41.2.




1) Расширенная схема 2) Условное обозначение

*Рис. 1.41.* Схема параллельного регистра

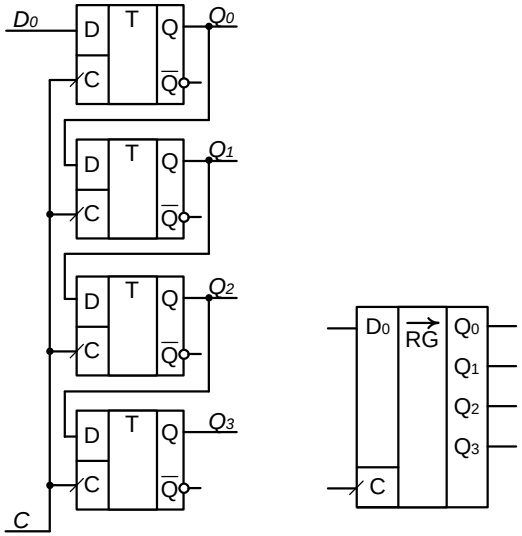
*2. Последовательный регистр* (регистр сдвига) обычно служит для преобразования последовательного кода в параллельный и наоборот. Т. е. последовательное накопление последовательной цифровой информации с последующей одновременной выдачей (преобразование последовательной цифровой информации в параллельный код) или одновременный прием (параллельный прием) информации с последующей последовательной выдачей (преобразование параллельного кода в последовательный).



Они могут быть использованы для построения умножителей и делителей чисел двоичной системы счисления, т.к. сдвиг двоичного числа влево на один разряд соответствует умножению его на два, а сдвиг вправо - делению на два.

Регистры сдвига могут служить также в качестве элементов задержки сигнала, представленного в цифровой форме. Действительно, регистры с последовательным приемом (вводом) и выводом осуществляют задержку передачи информации на m+1 тактов (m+1 - число разрядов регистра) машинного времени.

Схема регистра, осуществляющего преобразование последовательного кода в параллельный, приведена на рисунке 12.1, а его изображение на принципиальных схемах – на рисунке 12.2.

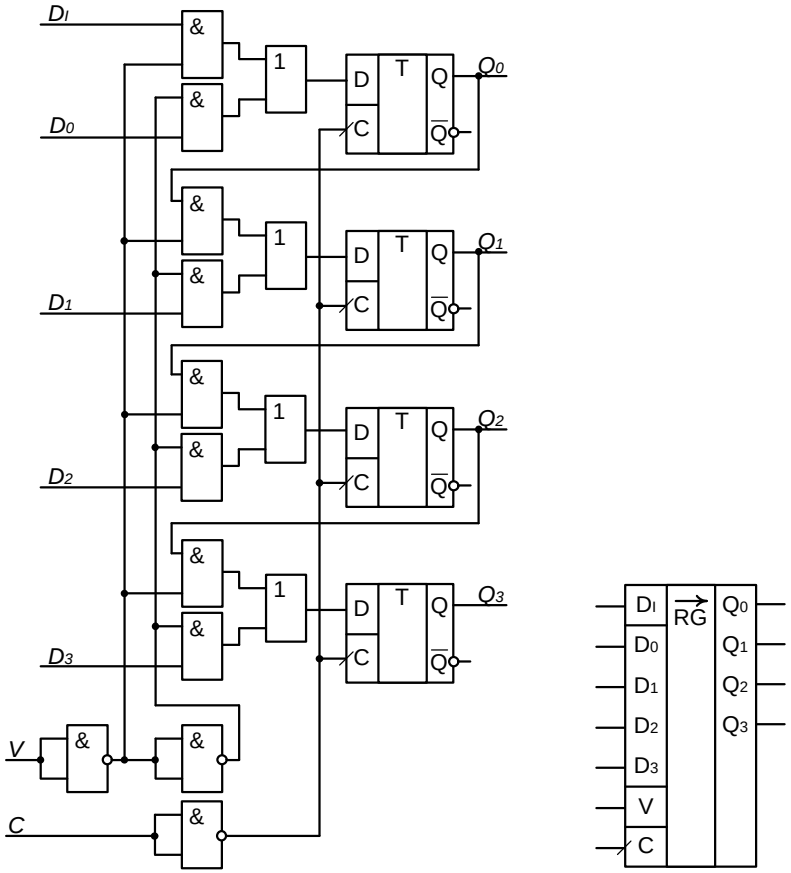



1) Расширенная схема 2) Условное обозначение

*Рис. 1.42.* Схема последовательного регистра

Регистры сдвига выполняют обычно как универсальные *последовательно-параллельные микросхемы*. Переключение регистра из параллельного режима в последовательный и наоборот осуществляется при помощи мультиплексо-

ра. Схема такого регистра приведена на рисунке 1.43.1, а его изображение на принципиальных схемах – на рисунке 1.43.2.




1) Расширенная схема 2) Условное обозначение

*Рис. 1.43* – Схема универсального регистра

## *Счетчики*

На базе счетных триггеров можно построить цифровое устройство, получившее название *электронного счетчика*.



*Счетчик* – устройство для подсчета числа входных импульсов. Число, представляющее состояние его выходов с приходом нового импульса изменяется на единицу.

Счетчики могут отличаться модулем счета и типом счетной последовательности, которая, в частности, может быть двоичной, двоично-десятичной, в коде Грея и т.п.

Цифровые последовательностные устройства, выполненные по схеме счетчика, но имеющие один счетный вход и один выход называются делителями частоты. Таким образом, любой счетчик может служить в качестве делителя частоты, если используется информация только одного из его выходов. Так как счетчики и делители имеют единую структуру, основное внимание будет уделено синтезу счетчиков.

Счетчики и делители подразделяются на асинхронные и синхронные. У синхронных счетчиков все разрядные триггеры синхронизируются параллельно одними и теми же синхроимпульсами, поступающими из источника этих импульсов. Асинхронные счетчики имеют последовательную синхронизацию, т.е. каждый последующий разрядный триггер синхронизируется выходными импульсами триггера предыдущего разряда.

*Синхронные счетчики*, в свою очередь, подразделяются на *параллельно-синхронные* и *последовательно-синхронные*. Параллельные счетчики имеют более высокую скорость счета, чем асинхронные.

Счетчики, независимо от способа синхронизации, подразделяются на счетчики *прямого счета (суммирующие)* и на счетчики *обратного счета (вычитающие)*. В интегральном исполнении выпускаются также реверсивные счетчики, в которых имеется специальный вход для переключения режима работы, т.е. направления счета. Многие типы счетчиков, выпускаемые промышленностью в интегральном исполнении, имеют дополнительные входы предустановки, позволяющие использовать эти счетчики в режиме регистра памяти.

В качестве разрядных триггеров счетчиков и делителей могут быть использованы двухступенчатые D-триггеры, T- и JK-триггеры.

Счетчики относятся к последовательностным устройствам с циклически повторяющейся последовательностью состояний. Число, соответствующее количеству импульсов (поступивших на вход счетчика), при котором счетчик “возвращается” в исходное состояние, называется *модулем* или *коэффициентом* счета.

Модуль счета, обычно, обозначают буквой *M* (или $К_{сч}$). Например, максимальный модуль счета счетчика из двух триггеров равен $M = 2^2 = 4$, трех триггеров - $M = 2^3 = 8$ и т.д. В общем случае для n - разрядного счетчика - $M = 2^n$. Модуль счета счетчика численно совпадает с модулем деления делителя частоты. Счетчик по модулю 8 позволяет реализовать (без дополнительных схемных затрат) делитель частоты на 8. Это значит, что данный делитель делит частоту входной импульсной последовательности на 8.

## Асинхронный двоичный счетчик

Асинхронный двоичный счетчик представляет собой совокупность последовательно соединенных триггеров (*D* или *JK*), каждый из которых ассоциируется с битом в двоичном представлении числа.

Счетная последовательность в двоичном суммирующем счетчике начинается с нуля и доходит до максимального числа $2^m$-1, после чего снова проходит через нуль и повторяется. В вычитающем двоичном счетчике последовательные двоичные числа перебираются в обратном порядке, и при повторении последовательности максимальное число следует за нулем.

Рассмотрим устройство двоичного суммирующего счетчика по модулю M=16, выполненного на базе JK-триггеров (рис. 1.44, а). Как видно из рисунка синхронизирующие входы всех триггеров, кроме крайнего левого, соединены с выходами предыдущих триггеров. Поэтому состояние очередного триггера меняется в ответ на изменение состояния предыдущего триггера.

Временные диаграммы, поясняющие работу асинхронного суммирующего счетчика приведены на рис. 1.44, б.



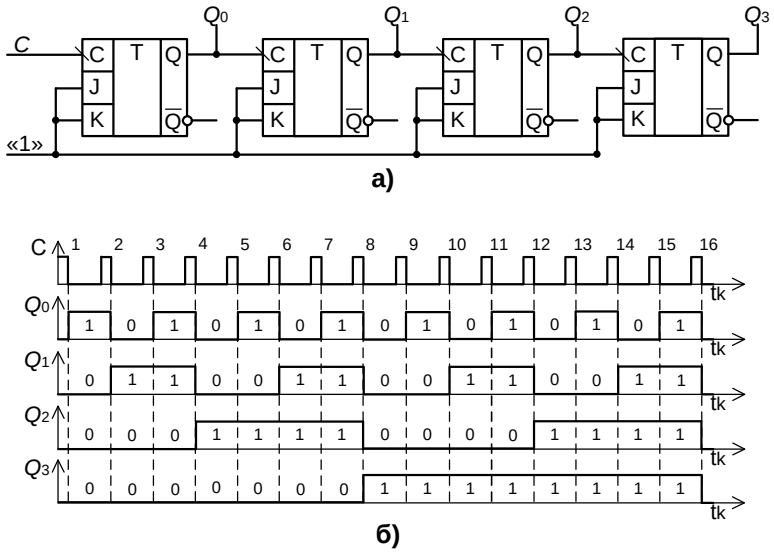



*Рис. 1.44.* Двоичный суммирующий счетчик по модулю *M*=16.

*Таблица 1.14.* Таблица счетчика рис. 1.14

| N | $Q_0$ | $Q_1$ | $Q_2$ | $Q_3$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 |
| 4 | 0 | 0 | 1 | 0 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 | 0 |
| 8 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 0 | 1 |
| 11 | 1 | 1 | 0 | 1 |
| 12 | 0 | 0 | 1 | 1 |
| 13 | 1 | 0 | 1 | 1 |
| 14 | 0 | 1 | 1 | 1 |
| 15 | 1 | 1 | 1 | 1 |

Из таблицы состояния счетчика (табл. 1.14) легко заметить, что значение разряда в выбранной позиции меняется тогда, когда в соседней справа позиции состояние переходит из “1” в “0”, управление триггерами осуществляется задним фронтом синхроимпульсов (отрицательным перепадом напряжения импульса синхронизации).

Счетчики обратного счета (вычитающие счетчики)

На рис. 1.45 приведена схема асинхронного трехразрядного двоичного вычитающего счетчика, построенного на базе *D-триггеров*. Условия для изменения состояний триггеров вычитающих счетчиков аналогичны условиям для суммирующих. Следовательно, рассмотренный выше счетчик можно превратить в вычитающий, просто переключив входы “C” триггеров с выходов $Q$ на выходы $\overline{Q}$. Или просто взяв информацию с инверсных выходов.

Работа вычитающего счетчика на *D-триггерах* наглядно иллюстрирована с помощью временных диаграмм рис. 1.45.б. Из рис. 1.45.б следует, что после нулевого состояния всех триггеров, с приходом первого синхроимпульса они устанавливаются в состояние “1”. Поступление второго синхроимпульса приводит к уменьшению этого числа на одну единицу и т.д. После поступления восьмого импульса, снова, все триггеры обнуляются и цикл счета повторяется, что соответствует модулю $M$=8.

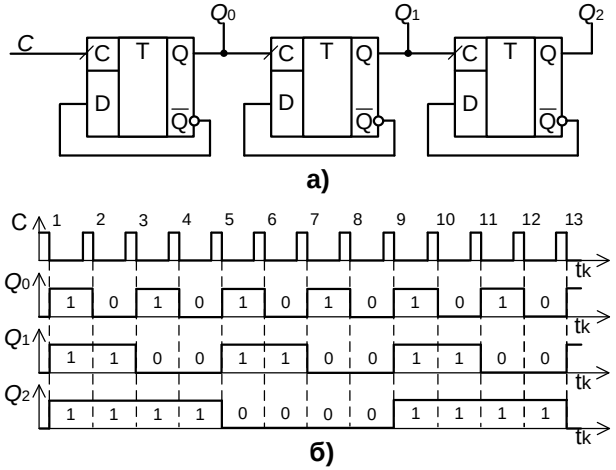



*Рис. 1.45.* Трехразрядный вычитающий счетчик

В некоторых случаях необходимо, чтобы счетчик мог работать как в прямом, так и в обратном направлении счета. Такие счетчики называются *реверсивными*.



Реверсивные счетчики могут быть как асинхронного, так и синхронного типа. Они строятся путем применения логических коммутаторов (мультиплексоров) в цепях связи между триггерами. Так, например, асинхронный реверсивный двоичный счетчик можно построить, если обеспечить подачу сигналов с прямого (при суммировании) или с инверсного (при вычитании) выхода предыдущего *JK*- или *T-триггера* на счетный вход последующего. В случае, когда реверсивный счетчик строится на базе *D-триггеров*, управляемых передним фронтом, для получения режима прямого счета следует соединить инверсный выход предыдущего со счетным входом последующего триггера.

Все рассмотренные типы счетчиков могут быть использованы в цифровых устройствах “умеренного” быстродействия, когда частота следования синхроимпульсов не превышает критического значения, при котором время задержки установки триггеров последних (старших) разрядов счетчика становится соизмеримым с длительностью периода входных тактовых импульсов. В связи с этим, асинхронные счетчики строятся на относительно небольшое количество разрядов, так как при большем количестве разрядов выходные сигналы триггеров старших разрядов появляются позднее, чем управляющие фронты синхроимпульсов (поступающих на вход первого триггера).

Синхронные счетчики

Как было уже сказано выше, параллельные счетчики бывают двух типов: *синхронные параллельные* и *синхронные последовательные*.

По способу подачи синхроимпульсов такие счетчики параллельные, т.е. синхроимпульсы поступают на все триггеры счетчика параллельно, а по способу управления (подачи управляющих импульсов) - последовательные. Схема синхронного *последовательного счетчика*, реализованного на *JK-триггерах*, приведена на рис. 1.46.

Синхронный последовательный счетчик обладает повышенным быстродействием, однако, за счет последовательного формирования управляющих уровней, на входы “J” и “K” счетных триггеров, быстродействие уменьшается.

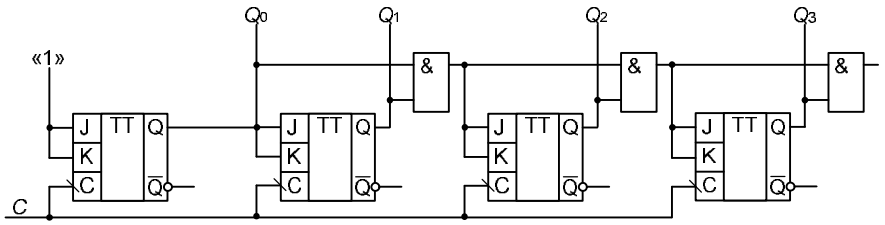



*Рис. 1.46.* Синхронный последовательный суммирующий счетчик

От этого недостатка лишены *параллельные синхронные счетчики*, в которых формирование управляющих уровней и их подача на соответствующие входы триггеров счетчика осуществляется одновременно, т.е. параллельно. Пример реализации параллельного синхронного счетчика показан на рис. 1.47.

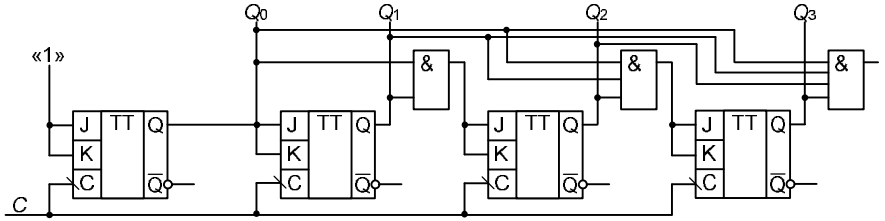



*Рис. 1.47.* Синхронный параллельный счетчик

Поскольку счетчик имеет одну общую линию синхронизации, состояние триггеров меняется синхронно, т.е. те триггеры, которые по синхроимпульсу должны изменить свое состояние, делают это одновременно, что существенно повышает быстродействие синхронных счетчиков.

## Счетчики с произвольным коэффициентом счета

Принцип построения подобного класса счетных устройств состоит в исключении нескольких состояний обычного двоичного счетчика, являющихся избыточными для счетчиков с коэффициентом пересчета, отличающимися от двоичных. При этом избыточные состояния исключаются с помощью обратных связей внутри счетчика.



Число избыточных состояний для любого счетчика определяется из следующего выражения:

$$M = 2^m - K_{сч}$$

где $M$ – число запрещенных состояний, $K_{сч}$ – требуемый коэффициент счета; $2^m$ – число устойчивых состояний двоичного счетчика.

Задача синтеза счетчика с произвольным коэффициентом счета заключается в определении необходимых обратных связей и минимизации их числа. Требуемое количество триггеров определяется из выражения

$$n = \log_2 K_{сч}$$

где округляется до ближайшего целого числа.

В каждом отдельном случае приходится применять какие-то конкретные методы получения требуемого коэффициента пересчета. Существует несколько методов получения счетчиков с заданным коэффициентом пересчета $K_{сч}$.

Один из этих методов заключается в немедленном сбросе в “0” счетчика, установившегося в комбинацию, соответствующему требуемому числу $K_{сч}$. Его называют также *методом автосброса*. Подобный прием удобно применять при использовании счетчиков в интегральном исполнении, имеющих ячейки конъюнкции (*И*) на входах установки в нуль, как это сделано в микросхеме К1533ИЕ5.

## Синтез счетчика с произвольным коэффициентом счета

Один из методов проектирования счетчиков с заданным коэффициентом счета заключается в построении таблицы переходов, в первых столбцах которых будут отражены текущие состояния триггеров счетчика, а в последующих – следующие за ними состояния. Анализ таблицы позволяет установить те переходы, которые должны быть “сделаны” триггерами, входящими в состав счетчика. Затем с помощью управляющей таблицы соответствующего триггера находятся значения логических функций на управляющих входах триггеров, позволяющие осуществить эти переходы.

Например, для счетчика на трех триггерах $K_{сч}=8$, если взять выход 1-ого и 2-го триггера, то $K_{сч}=4$. При этом $K_{сч}$ является целой степенью числа 2: *2, 4, 8, 16* и т. д.

Можно реализовать счетчик, для которого $K_{сч}$ – любое целое число. Например, для счетчика на трех триггерах можно сделать $K_{сч}$ от «2» до «7», но при этом один или два триггера могут оказаться лишними. При использовании всех трех триггеров можно получить изменение $K_{сч}=5...7$:

$$2^2 < K_{сч}, 2^3\ 2^2 < K_{сч} < 2^3$$

Счетчик с $K_{сч}=5$ должен иметь 5 состояний, которые в простейшем случае образуют последовательность: {0, 1, 2, 3, 4}. Циклическое повторение этой последовательности означает, что коэффициент деления счетчика равен «5».

Для построения суммирующего счетчика с $K_{сч}=5$ надо, чтобы после формирования последнего числа из последовательности {0, 1, 2, 3, 4} счетчик переходил не к числу «5», а к числу «0». В двоичном коде это означает, что от числа «100» нужно перейти к числу «000», а не «101». Изменение естественного порядка счета возможно при введении дополнительных связей между триггерами счетчика. Можно воспользоваться следующим способом: как только счетчик попадает в нерабочее состояние (в данном случае «101» или «110» или «111»), этот факт должен быть опознан и повлечь последующую выработку сигнала, который перевел бы счетчик в состояние «000». Рассмотрим этот способ более детально.

Факт попадания счетчика в нерабочее состояние описывается логическим уравнением:

$$F = (101) \vee (110) \vee (111) = Q_3 \cdot \overline{Q}_2 \cdot Q_1 + Q_3 \cdot Q_2 \cdot \overline{Q}_1 + Q_3 \cdot Q_2 \cdot Q_1 =$$
$$= Q_3 \cdot Q_1 + Q_3 \cdot Q_2$$

Состояния «110» и «111» также являются нерабочими и поэтому учтены при составлении уравнения. Если на выходе эквивалентной логической схемы *F=0*, значит, счетчик находится в одном из рабочих состояний: $(000) \vee (001) \vee (010) \vee (011) \vee (100)$. Как только он попадает в одно из нерабочих состояний (1.16), формируется сигнал *F=1*. Появление сигнала *F=1* должно переводить счетчик в начальное состояние «000», следовательно, этот сигнал нужно использовать для воздействия на установочные входы триггеров счетчика, которые осуществляли бы сброс счетчика в состояние $Q_1 = Q_2 = Q_3 = 0$. При реализации счетчика на триггерах с входами установки логическим нулем для сброса триггеров требуется подать на вход сброса сигнал $\overline{R} = 0$ (при этом должно быть всегда $\overline{S} = 1$) Для обнаружения факта попадания в нерабочее состояние используем схему, реализующую функцию *F* и выполненную на элементах И-НЕ. Для этого преобразуем выражение для функции:

$$F = \overline{Q_3 \cdot Q_1 + Q_3 \cdot Q_2} = \overline{Q_3 \cdot (Q_1 + Q_2)} = \overline{Q_3 \cdot \overline{\overline{(Q_1 + Q_2)}}} = \overline{Q_3 \cdot \overline{(\overline{Q}_1 \cdot \overline{Q}_2)}}$$

Соответствующая схемная реализация приведена на рис. 1.48.

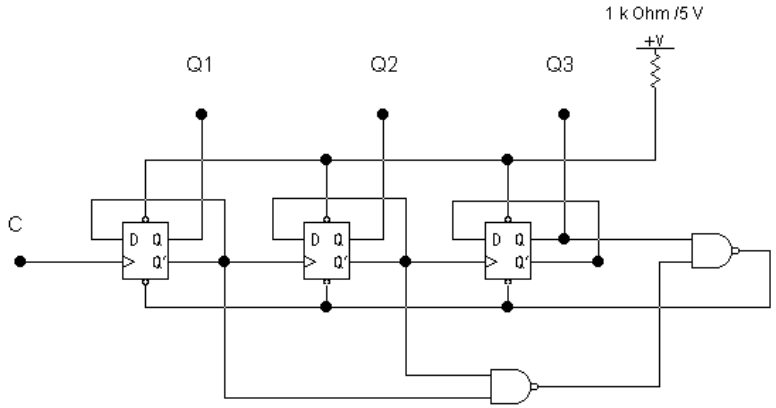



*Рис. 1.48.* Счетчик с измененным коэффициентом пересчета

Счетчик будет работать следующим образом: при счете от «0» до «4» все происходит как в обычном суммирующем счетчике с $K_{сч}=8$. Установочные сигналы равны «1» и естественному порядку счета не препятствуют. Счет происходит по положительному фронту импульса на счетном входе *C*. В тот момент, когда счетчик находится в состоянии «4» («100»), следующий тактовый импульс сначала переводит счетчик в состояние «5» («101»), что немедленно (задолго до прихода следующего тактового импульса) приводит к формированию сигнала сброса, который поступает на установочные входы сброса триггеров. В

результате счетчик сбрасывается в «0» и ждет прихода следующего тактового импульса на счетный вход. Один цикл счета закончился, счетчик готов к началу следующего цикла.

Применяя такие схемы с обратной связью для сброса счетчика, нужно иметь в виду, что операция сброса занимает конечное время, поэтому непосредственно перед сбросом счетчика в «0» на выходе первого триггера появляются кратковременные импульсы, или «иголки». Это не имеет значения при подключении счетчика напрямую к индикатору, но при использовании этого выхода счетчика в качестве источника тактовых импульсов могут возникнуть определенные проблемы.

Схема, в которой это явление устранено, использует *D-триггер*. Важным отличием является то, что схема обнаруживает не факт попадания в нерабочее состояние «101», а факт попадания в состояние «100» и в следующем такте вырабатывает сигнал сброса (триггер выполняет роль элемента задержки).

### *3. Понятие архитектуры. Классификация вычислительной техники*

Опр. *Вычислительная машина (ВМ)* – комплекс технических и программных средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

Структура ВМ определяется рядом требований, которые могут быть сформулированы с точки зрения конечных пользователей.

Опр. *Структура* – это совокупность элементов и их связей.

Различают структуры технических, программных и аппаратно-программных средств. Логически связанным с понятием структуры находится понятие *архитектура*, отличительной особенностью которого является выделение в качестве ключевого, существенного для пользователя признака, а именно организацию вычислений в ВМ.

Опр. *Архитектура* – многоуровневая иерархия аппаратно-программных средств, из которых строится ВМ.



Детализацией архитектурного и структурного построения ВМ занимаются различные категории специалистов вычислительной техники. Инженеры-схемотехники проектируют отдельные технические устройства и разрабатывают методы их сопряжения друг с другом. Системные программисты создают программы управления техническими средствами, информационного взаимодействия между уровнями, организации вычислительного процесса. Программисты-прикладники разрабатывают пакеты программ более высокого уровня, которые обеспечивают взаимодействие пользователей с ВМ и необходимый сервис при решении ими своих задач. Перечисленные специалисты рассматривают понятие архитектуры в более узком смысле.

С точки зрения обычного пользователя архитектура (точнее сказать в данном смысле это структура), чаще рассматривается с точки зрения характеристик ВМ, которые бывают следующих типов:

а) технические и эксплуатационные характеристики ВМ (быстродействие и производительность, показатели надежности, достоверности, точности, емкость оперативной и внешней памяти, габаритные размеры, стоимость технических и программных средств, особенности эксплуатации и др.);

б) характеристики и состав функциональных модулей базовой конфигурации ВМ; возможность расширения состава технических и программных средств; возможность изменения структуры;

в) состав программного обеспечения ЭВМ и сервисных услуг (операционная система или среда, пакеты прикладных программ, средства автоматизации программирования).

Введем пояснения технических и эксплуатационных характеристик.

*1. Быстродействие и производительность.*

*Быстродействие* – число команд выполняемых за одну секунду.

*Производительность* – объем работ, выполняемых в единицу времени.

Несмотря на то, что эти понятия тесно связаны между собой, имеются достаточно существенные отличия. Ярким примером отличия является выпуск

микропроцессоров серии AMD, на которых указана не реальная частота в герцах, а ее сопоставление с аналогами.

На настоящий момент не существует единого метода по их определению. Приведем некоторые способы оценки.

*а) MIPS* (Million Instructions Per Second - миллион операций в секунду). В качестве операций здесь обычно рассматриваются наиболее короткие операции типа сложения. На настоящий момент имеет ограничение по использованию, так как (набор команд современных микропроцессоров различен, MIPS существенно зависит от особенностей самой программы, в современных МПС имеются различные сопроцессоры – для чисел с плавающей точкой и без).

*б) MFPOPS* (Million Floating Point Operations Per Second - миллион операций с плавающей точкой в секунду). Для персональных ЭВМ этот показатель практически не применяется из-за особенностей решаемых задач и структурных характеристик.

в) Различные программные *тестовые наборы* (выпускаемые фирмами производителями, стандартные или универсальные, специализированные).

2. *Емкость запоминающих устройств.*

Характеризуется количеством структурных единиц информации, которые одновременно можно разместить в памяти (см. выше 1).

Простейшей единицей измерения бит, 8бит=1байт, 1Кбайт=1024байт, и так далее по «2» в степени 10. Разделяют емкость оперативной и внешней памяти.

3. *Надежность.*

Оценивается в соответствии со стандартом ISO (Международная организация стандартов) это способность выполнять требуемые функции при различных условиях. В современных ВМ непрерывно повышается, например, за счет применения СБИС (уменьшая тем самым количество межмодульных соединений), за счет удачной компоновки элементов (охлаждение, защита), за счет наличия функций самодиагностики.



4. *Точность*

Возможность различать почти равные значения (стандарт ISO-2382/2-76). Точность получения результатов обработки в основном определяется разрядностью микропроцессора (32, 64 и 128 двоичных разрядов). Во многих применениях ЭВМ не требуется большой точности, например при обработке текстов и документов, при управлении технологическими процессами. В этом случае достаточно воспользоваться 8- и 16-разрядными двоичными кодами. При выполнении же сложных математических расчетов следует использовать высокую разрядность.

5. *Достоверность*

Это свойство информации быть правильно воспринятой. Достоверность характеризуется вероятностью получения безошибочных результатов. Заданный уровень достоверности обеспечивается аппаратно-программными средствами контроля самой ВМ.

## *Классификация вычислительных машин*

Вычислительные машины могут быть классифицированы по ряду признаков, в частности: по принципу действия; по этапам создания и элементной базе; по назначению; по размеру вычислительной мощности; по функциональным возможностям (или применению – принцип Академика Глушкова); по способу организации вычислительного процесса (см. далее); и по способности к параллельному выполнению программ (см. однопроцессорные и многопроцессорные системы).

1. *Классы по принципу действия*

*Рис. 1.49.* Классы по принципу действия

а) АВМ – аналоговые, работают с информацией представленной непрерывно во времени (ток, напряжение). Недостатки – небольшая точность.

б) ЦВМ – цифровые, работают с информацией представленной дискретно во времени, т.е. информация кодируется в двоичной системе исчисления.

в) ГВМ – гибридные, работают с информацией представленной как в цифровой форме, так и в дискретной.

2. *По этапам создания и элементной базе*

1-е поколение, 50-е годы: ВМ на электронных вакуумных лампах.

2-е поколение, 60-е годы: ВМ на дискретных полупроводниковых приборах (транзисторах).

3-е поколение, 70-е годы: компьютеры на полупроводниковых интегральных схемах с малой и средней степенью интеграции (сотни-тысячи транзисторов в одном корпусе).

4-е поколение, 80-90-е годы: компьютеры на больших и сверхбольших интегральных схемах (десятки тысяч-миллионы активных элементов на одном кристалле, расстояние до 0.1 микрона)

5-е поколение, настоящее время: компьютеры со многими десятками параллельно работающих микропроцессоров, позволяющих строить эффективные системы обработки знаний; компьютеры на сверхсложных микропроцессорах с параллельно-векторной структурой, одновременно выполняющих десятки последовательных инструкций программы.

6-е и последующие поколения: оптоэлектронные компьютеры с массовым параллелизмом и нейронной структурой, с распределенной сетью большого числа (десятки тысяч) несложных микропроцессоров или нейронных системы.

3. *По назначению*

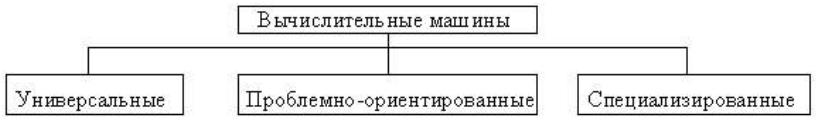



*Рис. 1.50.* Классификация по назначению



а) Универсальные – предназначены для решения самых различных задач (отличаются высокой производительностью, разнообразие форм обрабатываемых данных: двоичных и выполняемых операций, большой емкостью оперативной памяти, развитой системой ввода-вывода)

б) Проблемно-ориентированные предназначены для решения более узкого круга задач, связанных, как правило, с управлением технологическими объектами, с регистрацией, накоплением и обработкой относительно небольших объемов данных, с выполнением расчетов по относительно несложным алгоритмам; они обладают ограниченными по сравнению с универсальными компьютерами аппаратными и программными ресурсами.

в) Специализированные предназначены для решения определенного узкого круга задач или реализации строго определенной группы функций. Такая узкая ориентация компьютеров позволяет четко специализировать их структуру, существенно снизить их сложность и стоимость при сохранении высокой производительности и надежности их работы. К специализированным компьютерам можно отнести, например, программируемые микропроцессоры специального назначения, адаптеры и контроллеры, выполняющие логические функции управления отдельными несложными техническими устройствами, агрегатами и процессами, устройства согласования и сопряжения работы узлов вычислительных систем.

4. *По размерам и вычислительной мощности*

*Рис. 1.51.* Классы по принципу действия

Несмотря на тесную взаимосвязь в современных ВМ их размеров и вычислительных мощностей под мощностью здесь понимается функциональные возможности, которые определяются основными характеристиками, перечисленными выше (быстродействие, производительность, …). Исторический путь

развития ВМ, как отмечено выше, тесно связан с развитием элементной базы (см. классы ВМ, п.2 – «по этапам создания и элементной базе») и в рамках данного признака включал следующее обобщенное развитие: Большие ЭВМ, СуперЭВМ, Малые ЭВМ, микроЭВМ.

На настоящий момент данная классификация может быть раскрыта следующим образом:

а) *Большие ВМ* за рубежом часто называют *мэйнфреймами* (mainframe) - к ним относят, как правило, компьютеры, предназначенные для реализации многопользовательского режима работы с централизованной обработкой данных (количество обслуживаемых одновременно пользователей 1000 и более).

Основные направления эффективного применения мэйнфреймов – решение научно-технических задач, работа в вычислительных системах с пакетной обработкой информации, работа с большими базами данных, управление вычислительными сетями и их ресурсами. Последнее направление – использование мэйнфреймов в качестве больших серверов вычислительных сетей – часта отмечается специалистами как наиболее актуальное.

Родоначальником современных больших компьютеров, по стандартам которых в последние несколько десятилетий развивались машины этого класса в большинстве стран мира, являются машины фирмы IBM. Модели IBM 360 и IBM 370. На настоящий момент данное направление интенсивно развивается. Например, на 1999 год в моделях IBM AS/400 серий 720, 730 и 740 используется до 12 процессоров PowerPC и Pentium II (емкость оперативной памяти 16 Гбайт, а дисковой памяти - 2,1 Тбайт).

б) *СуперЭВМ* это дальнейшее развитие класса больших компьютеров. Отличаются использованием направления массового параллелизма, при котором одновременно могут работать десятки тысяч процессоров. На настоящий момент выпускаются несколькими фирмами: nCube (гиперкубические ЭВМ), Connection Machine, Mass Par, NCR/Teradata, KSR, IBM RS/6000, MPP и др.

Особенно эффективно применение суперЭВМ при решении задач проектирования, в которых натурные эксперименты оказываются дорогостоящими,



недоступными или практически неосуществимыми. В этом случае ЭВМ позволяет методами численного моделирования получить результаты вычислительных экспериментов, обеспечивая приемлемое время и точность решения, т.е. решающим условием необходимости разработки и применения подобных ЭВМ является экономический показатель, производительность, стоимость. СуперЭВМ позволяют по сравнению с другими типами машин точнее, быстрее и качественнее решать масштабные задачи, обеспечивая необходимый приоритет в разработках перспективной вычислительной техники.

*в) Малые ЭВМ*. Все модели миникомпьютеров разрабатываются на основе микропроцессорных наборов интегральных микросхем, 32, 64 и 128-разрядных микропроцессоров. Основные специфические особенности: архитектуру с большой модульностью; аппаратная реализация большинства системных функций ввода-вывода информации; лучшее, чем у мэйнфреймов, соотношение производительность-цена. Миникомпьютеры ориентированы на использование в качестве управляющих вычислительных комплексов. Традиционная для подобных комплексов широкая номенклатура периферийных устройств дополняется блоками межпроцессорной связи, благодаря чему обеспечивается реализация вычислительных систем с изменяемой структурой. Наряду с использованием миникомпьютеров для управления технологическими процессами, они успешно применяются для вычислений в многопользовательских вычислительных системах, в системах автоматизированного проектирования, в системах моделирования несложных объектов, искусственного интеллекта. Родоначальником современных миникомпьютеров можно считать компьютеры PDP-11 фирмы *DEC* (США),

г) *Микрокомпьютеры* весьма многочисленны и разнообразны.

*Универсальные* – предназначенные для решения различных задач.

*Специализированные* – для решения определенного класса задач (серверы предназначены в основном для обработки сетевых запросов; рабочие станции – предназначены для работы с определенным кругом специализированной информации, существенной для одного пользователя, например, графическая).

*Рис. 1.52.* Представители класса микрокомпьютеров

*Персональные ПК* – класс микрокомпьютеров. Черты: малая стоимость, автономность, гибкая архитектура, дружественный интерфейс пользователя.

## *Заключение*

Необходимо отметить, что любая предложенная классификация ВМ очень быстро устаревает и нуждается в корректировке. Например, понятия микроЭВМ исчезают из обихода. Однако существует целый ряд закономерностей развития вычислительной техники, которые позволяют предвидеть и предсказывать основные результаты этого поступательного движения.

*Принцип академика В.М. Глушкова.* Существуют три глобальные сферы деятельности человека, которые требуют использования качественно различных типов ВМ. Собственно говоря, это классификация ВМ с точки зрения их использования человеком.

*а) Первое направление* является традиционным – применение ЭВМ для автоматизации вычислений. Научно-техническая революция во всех областях науки и техники постоянно выдвигает новые научные, инженерные, экономические задачи, которые требуют проведения крупномасштабных вычислений (задачи проектирования новых образцов техники, моделирования сложных процессов, атомная и космическая техника и др.). Отличительной особенностью этого направления является наличие хорошей математической основы, заложенной развитием математических наук и их приложений. Первые, а затем и последующие вычислительные машины классической структуры в основном и создавались для автоматизации вычислений.



*б) Второе направление* – применение ВМ в системах управления. Оно зародилась примерно в шестидесятые годы, когда ЭВМ стали интенсивно внедряться в контуры управления автоматических и автоматизированных систем. Математическая база этой новой сферы практически отсутствовала, в течение последующих 15-20 лет она была создана. Новое применение вычислительных машин потребовало видоизменения их структуры. ЭВМ, используемые в управлении, должны были не только обеспечивать вычисления, но и автоматизировать сбор данных и распределение результатов обработки. Сопряжение с каналами связи потребовало усложнения режимов работы ЭВМ, сделало их многопрограммными и многопользовательскими. Для исключения взаимных помех между программами пользователей в структуру машин были введены средства разграничения: блоки прерываний и приоритетов, блоки защиты, средства измерения времени и т.п. Для управления разнообразной периферией стали использоваться специальные процессоры ввода-вывода данных или каналы. Именно тогда и появился дисплей как средство оперативного человеко-машинного взаимодействия пользователя с ЭВМ. Новой сфере работ в наибольшей степени отвечали мини-ЭВМ. Именно они стали использоваться для управления отраслями, предприятиями, корпорациями. Машины нового типа удовлетворяли следующим требованиям:

- были более дешевыми по сравнению с большими ЭВМ, обеспечивающими централизованную обработку данных;
- были более надежными, особенно при работе в контуре управления;
- обладали большой гибкостью и адаптируемостью настройки на конкретные условия функционирования;
-• имели архитектурную прозрачность, т.е. структура и функции ЭВМ были понятны пользователям.

Одновременно со структурными изменениями ЭВМ происходило и качественное изменение характера вычислений. Доля чисто математических расчетов постоянно сокращалась, и на сегодняшний день она составляет около 10% от всех вычислительных работ. Машины все больше стали использоваться для

новых видов обработки: текстов, графики, звука и др. Для выполнения этих работ в настоящее время применяются в основном ПЭВМ или ПК.

*в) Третье направление* связано с применением ВМ для решения задач, имеющих статистический и вероятностный подход, т.е. получение не точного результата, а чаще всего осредненного в статистическом, вероятностном смысле. Примеров подобных задач много: задачи робототехники, доказательства теорем, машинного перевода текстов с одного языка на другой, планирования с учетом неполной информации, составления прогнозов, моделирования сложных процессов и явлений и т.д. Это направление постепенно набирает силу. Во многих областях науки и техники создаются и совершенствуются базы данных и базы знаний, экспертные системы. Для технического обеспечения этого направления нужны качественно новые структуры ЭВМ с большим количеством вычислителей (ЭВМ или процессорных элементов), обеспечивающих параллелизм в вычислениях. По существу, ЭВМ уступают место сложнейшим вычислительным системам.

### *4. Функциональная и структурная организация ВМ*

Организация вычислительного процесса может рассматриваться с двух точек зрения:

1. Точка зрения пользователя.

*Учитывается* набор функций и услуг, обеспечивающих эффективное решение его задач;

*Не учитывается*: техническая реализация ВМ.

Опр. *Функциональная организация ВМ* - это абстрактная модель, описывающая функциональные возможности машины и предоставляемые ею услуги. Функциональная организация ЭВМ в значительной степени определяется предъявляемыми к ней требованиями, уровнем подготовки потенциальных пользователей, типом решаемых ими задач, потребностями в развитии компьютера (по емкости ЗУ, разрядности, составу периферийных устройств и др.).



2. Точка зрения разработчика (включая программиста).

*Учитывается* техническая реализация ВМ, диктуемая требованиями пользователя (т.е. является наиболее полной). С этой точки зрения вводится понятие

Опр. *Структурной организацией* физическая модель, устанавливающая состав, порядок и принципы взаимодействия основных функциональных частей машины.

По степени детальности различают структурные схемы, составленные на уровне: устройств, блоков, узлов, элементов.

*Элемент* это простейшее устройство ВМ, выполняющее одну операцию над входными сигналами (например, логический элемент).

*Узел* это часть, состоящая из нескольких более простых элементов и представляющая собой сборочную единицу (например, логическая схема).

*Блок* это функциональный компонент ЭВМ, состоящий из элементов и узлов и выполняющий операции над машинными словами или управляющий такими операциями (например, это: шифратор, дешифратор, мультиплексор).

*Устройство* это наиболее крупная функциональная часть ЭВМ, состоящая из элементов, узлов, блоков и выполняющая глобальные операции над кодированными данными (например, это процессор, память, контроллер).

Блоки и устройства часто изготавливаются в виде самостоятельных конструктивных модулей.

Движущим фактором в развитии структуры является рост производительности ВМ. Основным направлением развития является разделение функций системы и максимальная специализация подсистем ВМ, предназначенных для выполнения определенных задач.

Рассмотрим обобщенную структуру ВМ, в наиболее общем виде, с точки зрения деления ее на определенные подсистемы.

Обобщенная структура ЭВМ состоит из следующих составных частей: обрабатывающей подсистемы; подсистемы памяти; подсистемы ввода-вывода;

подсистемы управления и обслуживания. Данная структура и взаимодействие между ее составными частями показано на рис.

Взаимодействие данных подсистем представлено на рис. 1.53.

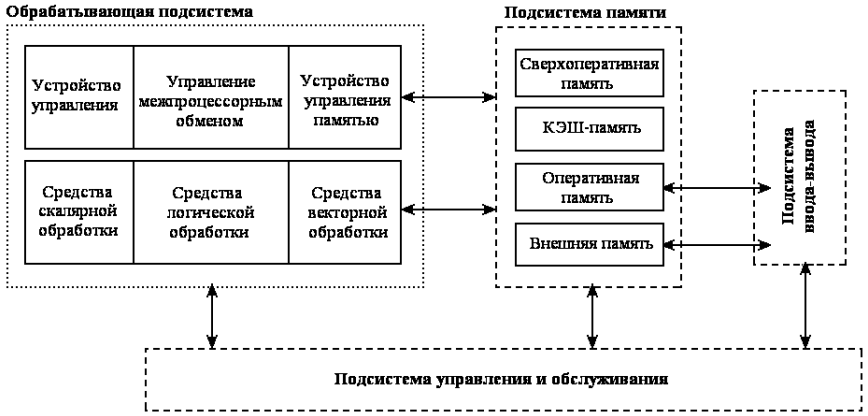



*Рис. 1.53.* Обобщенная структура вычислительной машины

Отметим, что данная структура является общим представлением, и далее в зависимости от необходимости будет детализироваться и рассматриваться с других точек зрения.

Приведем характеристики каждой из подсистем и архитектурные способы повышения производительности всей системы в целом.

Обрабатывающая подсистема. Выполняет основные операции всей системы. Выполняет задачи логической и арифметической обработки данных, содержит средства скалярной обработки команд, устройство управления памятью. В сложных системах (ВС) содержит так же средства векторной обработки, устройство управления межпроцессорным обменом. Наиболее динамично развивающаяся подсистема.

*Повышение производительности достигается* за счет применения методов конвейерной обработки наряду с совершенствованием алгоритмов диспетчеризации и исполнения команд.

Подсистема памяти. Подсистема памяти современных компьютеров имеет иерархическую структуру, состоящую из нескольких уровней:



- сверхоперативный уровень (локальная память процессора, кэш-память первого и второго уровня и др.);

- оперативный уровень (оперативная память и др.);

- внешний уровень (внешние запоминающие устройства на дисках и др.).

Каждый уровень состоит из запоминающих устройств, обладающих различным быстродействием и емкостью. Чем выше уровень, тем выше быстродействие соответствующей памяти, но меньше ее емкость.

*Эффективными методами повышения производительности* ВМ являются увеличение количества регистров общего назначения процессора, использование многоуровневой кэш-памяти, увеличение объема и пропускной способности оперативной памяти, буферизация передачи информации между разными уровнями памяти. Увеличение пропускной способности оперативной памяти будет рассмотрено ниже.

Подсистема ввода-вывода. В состав подсистемы ввода-вывода входит набор специализированных устройств, между которыми распределены функции ввода-вывода. Основная задача освободить систему от загрузки в режиме работы с внешними устройствами. Содержит обработчики команд ввода-вывода, контроллеры интерфейсов и линии передачи и приема информации.

*Основными направлениями повышения производительности* подсистем ввода-вывода являются канальная технология ввода-вывода, матричная топология коммутации периферийных устройств (ПУ), увеличение количества и пропускной способности каналов.

Заметим что весь обмен информацией происходит через оперативную память, которая является ядром взаимодействия всех других систем.

Подсистема управления и обслуживания. Осуществляет инициализацию, тестирование, отладку и контроль за эффективностью работы всей системы в целом.

*Повышение производительности* достигается за счет применения новых алгоритмов управления и способов взаимодействия с другими подсистемами.

Примечание: повышение эффективности работы ВМ достигается не только повышением эффективности работы каждой из подсистем, а их сбалансированным сочетанием. Например, быстродействие обрабатывающей подсистемы должно сочетаться с объемами и скоростью передачи информации в подсистемы памяти и пропускной способностью подсистемы ввода.

Очевидно, что структура вычислительной системы, представленная на рис. 1.54 актуальна и в настоящее время. Заметим, что ее основу составили, и до сих пор составляют принципы, предложенные много лет назад.

Принцип организации вычислительного процесса

Принцип организации вычислительного процесса базируется на концепции Дж. Фон Неймана, выдвинутой им в 1945 г. Согласно данной концепции выделяется автономно работающая ВМ, содержащая устройство управления, арифметико-логическое устройство (АЛУ), память и устройства ввода, вывода (рис. 1.54).

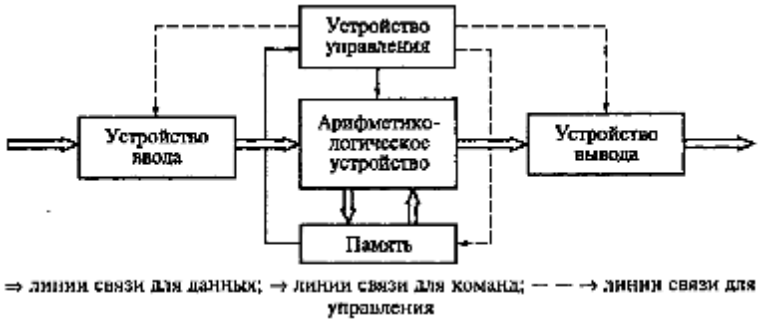



*Рис. 1.54.* Принцип организации вычислительного процесса, предложенный ВМ Дж. Фон Неймана

В связях, соединяющих устройства, выделены потоки данных, команд и управляющих сигналов. Преобразование данных осуществляется последовательно под централизованным управлением от программы, состоящей из команд. Набор (список) команд составляет машинный язык низкого уровня.



Автором концепции для организации вычислительного процесса предложены следующие принципы (ниже приводится не дословная концепция, а ее адаптация под современную терминологию и уровень научно-технического процесса).

1. Двоичное кодирование информации, разделение ее на слова фиксированной разрядности.

2. Линейно-адресная организация памяти ($N$ ячеек по $n$ разрядов). Аппаратные средства для записи, хранения и чтения слова из $n$ двоичных разрядов называют ячейкой памяти. Ячейки пронумерованы по порядку (0, 1, ..., $N$- 1). Номер ячейки это адрес. В командах программы адрес является именем (идентификатором) переменной, хранящейся в соответствующей ячейке.

3. Представление алгоритма программой, состоящей из команд. Команда является предписанием, определяющим шаг процесса выполнения программы. Она содержит код операции, адреса операндов и другие служебные коды.

4. Хранение команд и данных в одной памяти. Различие их заключается только в способе использования и интерпретации считанного из памяти слова.

5. Вычислительный процесс организуется как последовательное выполнение команд в порядке, определяемом программой.

6. Жесткость архитектуры предполагает неизменность в процессе работы ВМ, ее структуры, списка команд, методов кодирования данных.

При работе вычислительной машины наиболее интенсивное взаимодействие осуществляется между АЛУ и устройством управления. С развитием элементной базы в целях повышения производительности за счет уменьшения задержек в связях эти устройства объединили в один блок, называемый *процессором*.

Процессор считывает и выполняет команды программы, организует обращение к оперативной памяти, в нужных случаях инициирует работу устройств ввода и вывода.

Устройство ввода преобразует входные сигналы в сигналы, принятые для представления данных на шине, соединяющей устройство ввода с АЛУ.

В памяти хранятся команды и данные, которыми оперирует процессор. В нее же записываются результаты промежуточных вычислений. Результаты выполнения программы поступают в устройство вывода.

Устройство вывода преобразует выходные сигналы в форму, удобную для восприятия человеком (тексты, графические образы и др.).

Выборка команды из памяти и ее выполнение циклически повторяются. Цикл включает следующие фазы: выборку, дешифрацию и исполнение.

В фазе выборки команды содержимое ячейки памяти, адресуемое специальной схемой устройства управления (программным счетчиком), выбирается из памяти и помещается в специальный регистр процессора, называемый регистром команд. В фазе дешифрации код команды из регистра команд поступает в дешифратор команд процессора и преобразуется в последовательность управляющих сигналов, обеспечивающих настройку АЛУ для выполнения функции, определяемой командой. После этого в устройстве управления формируется адрес следующей команды. Следующей является фаза исполнения. После выполнения команды происходит возврат к первой фазе - выборке следующей команды.

Принципы Дж. фон Неймана имеют глубокую внутреннюю согласованность и в значительной мере используются в архитектуре современных компьютеров. На ранних этапах развития вычислительной техники (в первых моделях ВМ) существенным было требование максимальной простоты машинного языка и в соответствии с этим - устройств ВМ. В настоящее время это требование не является определяющим и наблюдается отход от исходных принципов. В значительной мере он связан с введением параллелизма на различных уровнях организации вычислительного процесса и расширением функций, реализуемых аппаратно. Эти изменения связаны с постоянным ростом сложности выполняемых преобразований и требований к скорости вычислений. Возможности реализации изменений обусловлены успехами микроэлектроники и совершенствованием элементной базы современных компьютеров.



*Заключение*

Таким образом под аппаратными средствами вычислительной машины будем понимать отдельные ее составные части, классификация и предназначение которых вытекает из структуры Дж. фон Неймана и в обобщенном виде изложены выше.

## *5. Аппаратные особенности вычислительных машин различных поколений*

Выше была представлена классификация ВМ по этапам создания и элементной базе – так называемые поколения ВМ. Данное деление было основано не только на временных интервалах, но и было привязано к определенным достижениям в микроэлектронике. Здесь же основное внимание уделим основным архитектурным особенностям разных поколений.

### Вычислительные машины первого и второго поколений

Схема ЭВМ, отвечающая программному принципу управления, логично вытекает из последовательного характера преобразований, выполняемых человеком по некоторому алгоритму (программе). Обобщенная структурная схема ЭВМ первых поколений представлена на рис. 5.1.

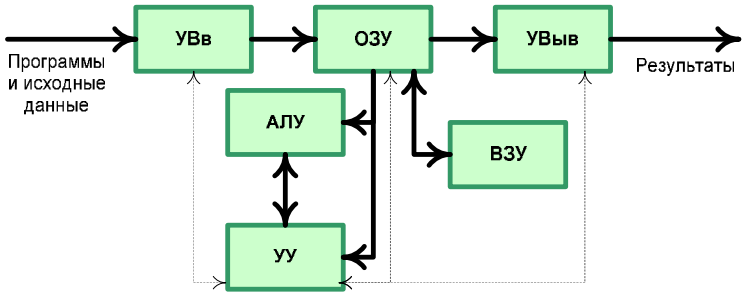



*Рис. 1.55.* Структура ВМ 1 и 2 поколений

В любой ЭВМ имеются устройства ввода информации (УВв), с помощью которых пользователи вводят в ЭВМ программы решаемых задач и данные к ним. Сначала введенная информация полностью или частично запоминается в

оперативном запоминающем устройстве (ОЗУ), а затем переносится во внешнее запоминающее устройство (ВЗУ), предназначенное для длительного хранения информации, где преобразуется в специальный программный объект – файл.

*Файл* – это имеющий имя информационный массив (программа, данные, текст и т.п.), размещаемый во внешней памяти и рассматриваемый как неделимый объект при пересылках и обработке.

При использовании файла в вычислительном процессе его содержимое переносится в ОЗУ. Затем программная информация команда за командой считывается в устройство управления.

Устройство управления (УУ) предназначается для автоматического выполнения программ путем принудительной координации работы всех остальных устройств ЭВМ. Цепи сигналов управления показаны на рис. 1.55 штриховыми линиями. Вызываемые из ОЗУ команды дешифрируются устройством управления: определяются код операции, которую необходимо выполнить следующей, и адреса операндов, принимающих участие в данной операции.

Арифметико-логическое устройство (АЛУ) выполняет арифметические и логические операции над данными. Основной частью АЛУ является операционный автомат, в состав которого входят сумматоры, счетчики, регистры, логические преобразователи и др. Оно каждый раз перестраивается на выполнение очередной операции. Результаты выполнения отдельных операций сохраняются для последующего использования на одном из регистров АЛУ или записываются в память. Отдельные признаки результатов $r$ ($r$=0, $r$<0, $r$>0 и др.) устройство управления использует для изменения порядка выполнения команд программы. Результаты, полученные после выполнения всей программы вычислений, передаются на устройства вывода (УВыв) информации. В качестве УВыв могут использоваться экран дисплея, принтер, графопостроитель и др.

Центральным элементом данной структуры является АЛУ, рассмотрим его более подробно.

Структура простейшего АЛУ

Рассмотрим функционирование АЛУ на примере операции сложения.



*Управляющие сигналы регистров* – управляют записью в них кодов данных. *Управляющий сигнал комбинационной схемы* настраивает ее на выполнение одной из операций (сложение, вычитание, логические операции, и т.п.). *Управляющий сигнал регистра признаков* (предназначен для побитного отображения свойств результата: равен нулю, положителен, отрицателен и т.п.) разрешает/запрещает изменение его состояния. Каждая операция выполняется регистровой передачей и осуществляется за один такт. Т.е. сначала записываются данные в Регистр 1, затем в Регистр 2, потом осуществляется передача данных из регистров 1, 2 в комбинационную схему (настроенную на соответствующую операцию) и реализуется операция сложения с записью результата в *Регистр результата* с модификацией *Регистра признаков*.

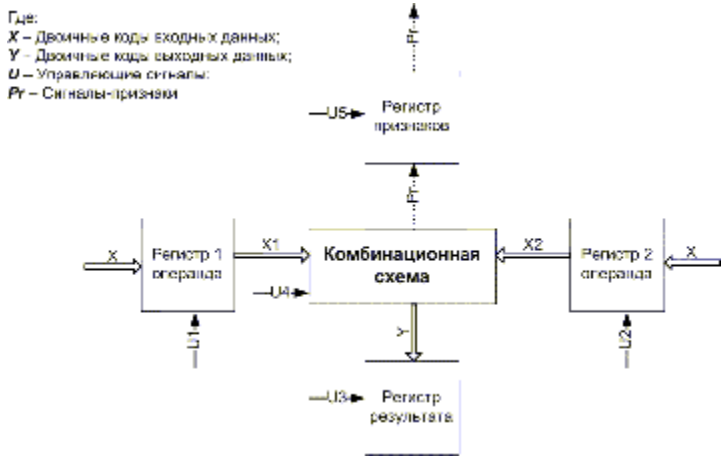



*Рис. 1.56.* Структура АЛУ

Уже в первых ЭВМ для увеличения их производительности широко применялось совмещение операций. При этом последовательные фазы выполнения отдельных команд программы (формирование адресов операндов, выборка операндов, выполнение операции, отсылка результата) выполнялись отдельными функциональными блоками. В своей работе они образовывали своеобразный *конвейер*, а их параллельная работа позволяла обрабатывать различные фазы целого блока команд. Этот принцип получил дальнейшее развитие в ЭВМ сле-

дующих поколений. Но все же первые ЭВМ имели очень сильную централизацию управления, единые стандарты форматов команд и данных, жесткое построение циклов выполнения отдельных операций, что во многом объясняется ограниченными возможностями используемой в них элементной базы. Центральное УУ обслуживало не только вычислительные операции, но и операции ввода-вывода, пересылок данных между ЗУ и др. Все это позволяло в какой-то степени упростить аппаратуру ВМ, но значительно сдерживало рост их производительности.

*ВМ третьего поколения*

В рамках развития данного поколения произошло усложнение структуры за счет разделения процессов ввода-вывода информации и процесса ее обработки – см. рис. 1.57.

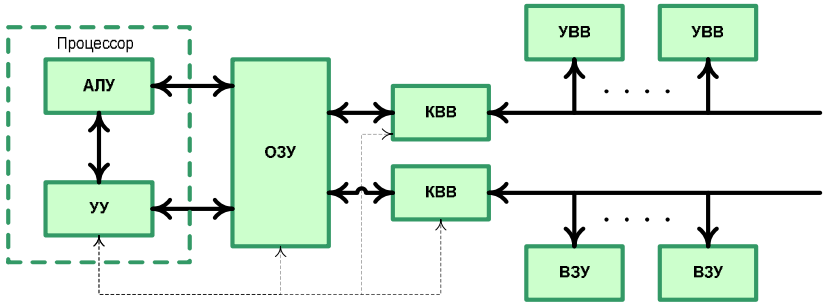



*Рис. 1.57.* Структура ВМ третьего поколения

Сильно связанные устройства АЛУ и УУ получили название *процессор*. В структуре ЭВМ появились дополнительные устройства, которые стали называться: процессоры ввода-вывода, устройства управления обмена информацией, каналы ввода-вывода (КВВ). Последние получили наибольшее распространение применительно к большим ЭВМ. Здесь наметилась тенденция к децентрализации управления и параллельной работе отдельных устройств, что позволило резко повысить быстродействие ЭВМ в целом. Среди каналов ввода-вывода выделяли мультиплексные каналы, способные обслуживать большое количество медленно работающих устройств ввода-вывода, и селекторные каналы, обслуживающие в монопольных режимах скоростные внешние запоминающие устройства (ВЗУ).



*ВМ четвертого и пятого поколения*

В первых персональных ЭВМ – так называемых ПЭВМ, относящихся к ВМ *четвертого поколения*, произошло дальнейшее изменение структуры (рис. 1.58). Они унаследовали ее от первых мини-ЭВМ.

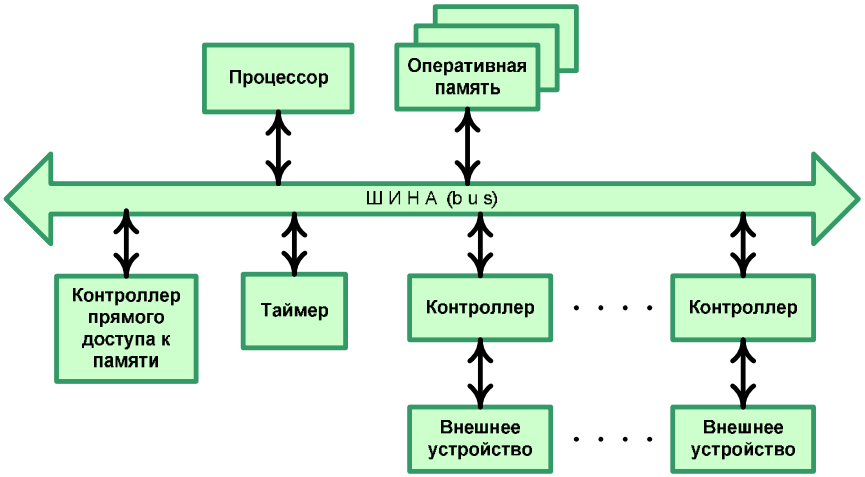



*Рис. 1.58.*Структура ВМ четвертого поколения

Соединение всех устройств в единую машину обеспечивается с помощью общей шины, представляющей собой линии передачи данных, адресов, сигналов управления и питания. Единая система аппаратурных соединений значительно упростила структуру, сделав ее еще более децентрализованной. Все передачи данных по шине осуществляются под управлением сервисных программ.

Ядро системы образуют процессор и основная память (ОП), состоящая из оперативной памяти и постоянного запоминающего устройства (ПЗУ). ПЗУ

предназначается для записи и постоянного хранения наиболее часто используемых программ управления. Подключение всех внешних устройств (ВнУ): дисплея, клавиатуры, внешних ЗУ и др., обеспечивается через соответствующие адаптеры – согласователи скоростей работы сопрягаемых устройств или контроллеры – специальные устройства управления периферийной аппаратурой.

Контроллеры в ПЭВМ играют роль каналов ввода-вывода. В качестве особых устройств следует выделить таймер – устройство измерения времени и контроллер прямого доступа к памяти (КПД) - устройство, обеспечивающее доступ к ОП, минуя процессор.

Для ВМ *пятого поколения* характерно усложнение структуры представленной на рис. 1.58 и использовании вместо одного процессора многоядерных решений, которые в одном корпусе содержат несколько независимых процессоров (см. ниже). При этом общая структура, по сути идентична представленной на рис. 1.58.

Учитывая то, что большая часть распространенных в настоящее время решений в вычислительной технике относят к пятому поколению, выделим основные принципы их построения и функционирования.

### *Основные принципы построения ВМ*

*Модульность построения* предполагает выделение в структуре ЭВМ достаточно автономных, функционально и конструктивно законченных устройств (процессор, модуль памяти, накопитель на жестком или гибком магнитном диске). Модульная конструкция ЭВМ делает ее открытой системой, способной к адаптации и совершенствованию. К ЭВМ можно подключать дополнительные устройства, улучшая ее технические и экономические показатели. Появляется возможность наращивания вычислительной мощи, улучшения структуры путем замены отдельных устройств на более совершенные, изменения и управления конфигурацией системы, приспособления ее к конкретным условиям применения в соответствии с требованиями пользователей.



Все существующие типы ЭВМ выпускаются *семействами*, в которых различают старшие и младшие модели. Всегда имеется возможность замены более слабой модели на более мощную. Это обеспечивается информационной, аппаратной и программной совместимостью. Программная совместимость в семействах устанавливается по принципу снизу вверх, т.е. программы, разработанные для ранних и младших моделей, могут обрабатываться и на старших, но не обязательно наоборот.

*Иерархическая организация*. Модульность построения предполагает жесткую *иерархическую* организацию. Устройство управления главного, или центрального, процессора определяет лишь последовательность работ подчиненных модулей и их инициализацию, после чего они продолжают работу по собственным программам управления. Результаты выполнения требуемых операций представляются ими вверх по иерархии для правильной координации всех работ.

*Децентрализация управления* является следствием иерархической организации. Подразумевает принцип параллельной работы.

В современных ЭВМ принцип децентрализации и распространен как на периферийные устройства, так и на сами ЭВМ (процессоры). Появились вычислительные системы (ВС), содержащие несколько вычислительных узлов (ЭВМ или процессоры), работающие согласованно и параллельно. Внутри самой ЭВМ произошло еще более резкое разделение функций между средствами обработки. Сначала появились отдельные специализированные процессоры, например сопроцессоры, выполняющие обработку чисел с плавающей точкой, матричные процессоры и др. Децентрализация управления и структуры ЭВМ позволила перейти к более сложным *многопрограммным* (*мультипрограммным)* режимам. При этом в ЭВМ одновременно может обрабатываться несколько программ пользователей.

*Примечание*: в однопроцессорных системах мультипрограммный режим работы реализован как постоянное переключение процессора от задачи к зада-

че, т.е. параллельная обработка организована как работа с процессами, разделенная во времени.

В настоящее время вместо однопроцессорных систем все большее применение находят многопроцессорные или многоядерные решения в которых возможна параллельная работа нескольких процессоров или ядер.

Общий принцип функционирования ВМ для 1-5 поколений

Процессор считывает и выполняет команды программы, организует обращение к памяти, и в нужных случаях инициирует работу устройств ввода-вывода.

УВвода преобразуют сигналы принятые для представления в ВМ, для последующей их передачи на линии связи ВМ.

В памяти хранится команды и данные, которыми оперирует процессор; в нее же записываются результаты промежуточных вычислений. Итоговые результаты операций поступают в УВВ.

УВывода преобразуют выходные сигналы в форму удобную для восприятия или дальнейшей передачи.

Выборка команды из памяти и ее выполнение циклически повторяются. Циклы: *выборка* (содержимое ячейки памяти, адресуемое схемой устройства управления выбирается из памяти и помещается в специальный регистр процессора – регистр команд), *дешифрирование* (код команды поступает в дешифратор команд и преобразуется в последовательность управляющих сигналов, обеспечивающих настройку АЛУ; после в УУ формируется адрес следующей команды), *исполнение* (непосредственно реализация и возвращение к выборке).

Кризис структуры фон Неймана

Практически исчерпаны структурные методы повышения производительности ЭВМ;



Плохо развиты средства обработки нечисловых данных (структуры, символы, предложения, графические образы, звук, очень большие массивы данных и др.);

Несоответствие машинных операций операторам языков высокого уровня;

Примитивная организация памяти ЭВМ;

Низкая эффективность ЭВМ при решении задач, допускающих параллельную обработку, и т.п.

Все эти недостатки аппаратуры приводят к чрезмерному усложнению комплекса программных средств, применяемого для подготовки и решения задач пользователей. Более подробно способы преодоления кризиса будут рассмотрены ниже. Здесь же отметим что центральной задачей всех производителей ВМ является увеличение производительности, с минимальными затратами. Большая часть используемых и доступных сейчас подходов практически исчерпала себя и необходимы принципиально новые решения и фундаментальный отказ от принципов работы, заложенного более 50 лет назад.

*ВМ шестого поколения*

В вычислительных машинах данного поколения, с использованием в них встроенного искусственного интеллекта, предполагается дальнейшее усложнение структуры и архитектуры. В первую очередь это касается совершенствования процессов общения пользователей с ЭВМ (использование аудио-, видеоинформации, систем мультимедиа и др.), обеспечения доступа к информационным хранилищам (базам данных и базам знаний), организации параллельных вычислений. Несомненно, что этому должны соответствовать новые параллельные структуры с новыми принципами их построения, более экономически выгодные нежели простое увеличение количества решающих элементов.

Данное поколение сейчас находится на стадии разработки.

## Зачетные вопросы, рассматриваемые в теме

1. Основные понятия о процессе автоматизированной обработки данных: определения, меры информации.

2. Основные понятия о процессе автоматизированной обработки данных: показатели качества информации, функционирование вычислительной машины.

3. Принцип действия ВМ: системы исчисления, основы алгебры логики, логические операции и логические элементы.

4. Принцип действия ВМ: разные формы представления работы цифрового устройства к другой.

5. Принцип действия ВМ: комбинационные и последовательностые элементы, шифраторы, дешифраторы, мультиплексоры, сумматоры, полусумматоры, счетчики, регистры.

6. Архитектура и классификация ВМ: определения, основные характеристики ВМ.

7. Архитектура и классификация ВМ: классификация ВМ по принципу действия, элементной базе, по назначению, по мощности; основные области применения ВМ.

8. Общие понятия о функциональной и структурной организации ВМ. Обобщенная структура ВМ, ее подсистемы.

9. Аппаратные особенности ВМ первого и второго поколений. Структура простейшего АЛУ.

10. Аппаратные особенности ВМ третьего, четвертого и пятого поколений.

11. Основные принципы построения и функционирования ВМ. Кризис структуры фон-Неймана, ВМ шестого поколения.



## Примечания

При разработке данной темы первоисточником явились изданиях [5-7], содержание которых было дополнено актуальной информацией полученной из открытых источников.

При изучении настоящей темы так же рекомендуется обратиться к соответствующим презентациям по дисциплине, где представлены в качестве сбавочных данных некоторые сведения по имеющимся в настоящее время достижениям в области компьютерных, сетевых и информационных технологий.

## ТЕМА № 2. ОСОБЕННОСТИ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И КОМПЬЮТЕРНЫХ СИСТЕМ

*Вопросы, рассматриваемые в данной теме*:

1. Общие вопросы организации процессора, памяти и обмена данными.
2. Персональные компьютеры: особенности архитектуры и применения.

### Теоретические сведения

В рамках данной темы рассматривается вторая часть главы 1 «История развития, архитектура и базовые понятия компьютерных технологий».

### *1. Общие вопросы организации процессора, памяти и обмена данными*

Процессор занимает центральное место в структуре ВМ. Он непосредственно осуществляет процесс обработки цифровой информации в ВМ и программное управление этим процессом. С его помощью происходит управление взаимодействием всех устройств, входящих в состав ВМ. Процессор считывает и выполняет команды программы, организует обращение к оперативной памяти, в нужных случаях инициирует работу периферийных устройств, воспринимает и обслуживает запросы прерываний, поступающие из устройств ВМ и извне.

Процессоры современных ВМ в большинстве случаев реализуют на одном кристалле с использованием технологии СБИС. Соответствующую интегральную схему называют *микропроцессором* (МП). Понятие МП в функциональном отношении совпадает с понятием процессор и отражает лишь особенности, связанные с использованием технологии СБИС при его реализации.

Основными достоинствами МП, как одного из наиболее массовых типов средств вычислительной техники, являются программируемость, дешевизна, малые габариты и вес, надежность, простота эксплуатации. Появление МП сыграло революционную роль в автоматике и вычислительной технике, послужив мощным ускорителем развития средств и систем управления и контроля. С созданием МП впервые появилась возможность применять вычислительные устройства в таких областях, где раньше это казалось неэффективным либо просто невозможным. Уже первые использования МП определили два основных направления их применения: использование МП в качестве центральных процессоров ВМ, реализация на базе МП встраиваемых систем управления различными объектами.



#### *Организация простейшего процессора*

В самом общем случае структурно-функциональную схему простейшего МП можно представить в виде композиции трех функциональных блоков: операционного блока (ОБ), блока управления и интерфейсного блока (рис. 2.1).

Кроме них в состав микропроцессора могут входить и некоторые другие блоки, участвующие в организации вычислительного процесса, например, блок прерывания, блок защиты памяти, блоки контроля, диагностики и др.

Здесь для примера взята простейшая схема микропроцессора с аккумуляторной архитектурой, используемая в первых версиях коммерчески успешных решений.

*Операционный блок*. Предназначен для выполнения некоторого функционально полного набора логических и арифметических операций. Как правило, в его состав входят АЛУ, буферные регистры операндов, регистр результата (аккумулятор), регистр признаков и блок регистров общего назначения (РОН). Комбинационная схема (см. ранее), являющаяся основой АЛУ, содержит двоичный сумматор и набор логических схем. В АЛУ выполняются несколько простейших арифметических (сложение, вычитание) и поразрядных логических (И, ИЛИ, НЕ и др.) операций. В многоуровневой организации вычислительного процесса указанные операции реализуются на уровне регистровых передач между источниками операндов и приемником результата. Операции по обработке данных, для которых в ОБ отсутствуют аппаратные средства, выполняют

программно с помощью процедур. Такие процедуры реализуются в виде последовательности простых операций ОБ, т.е. выполняются на более высоком уровне организации вычислительного процесса, чем уровень регистровых передач.

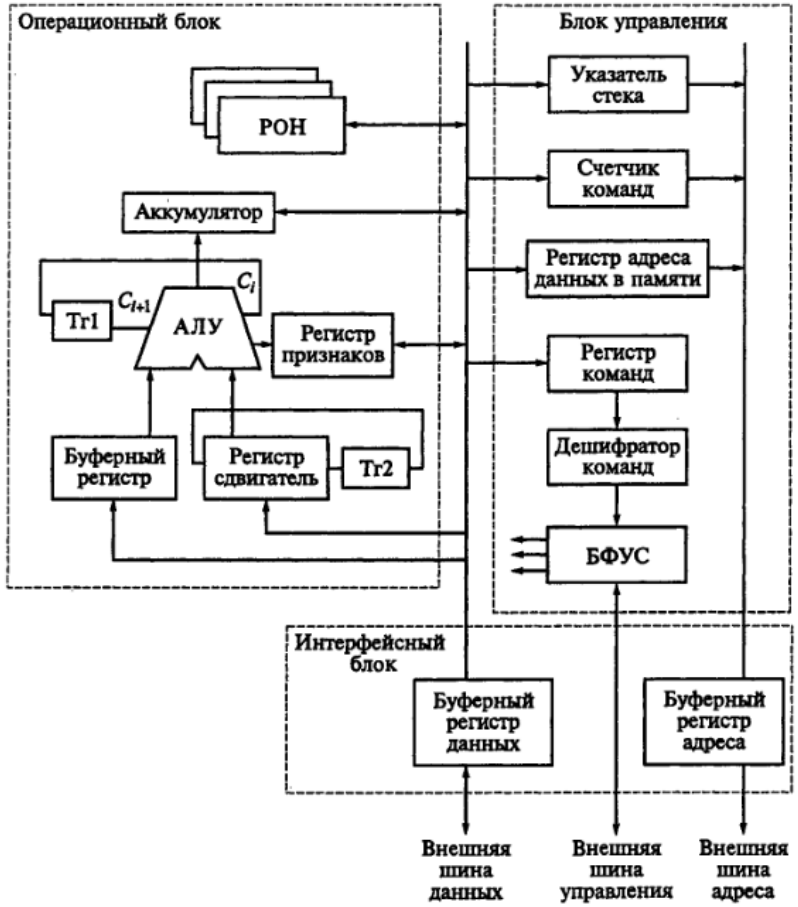



*Рис. 2.1.* Типовая структура МП:

РОН – регистры общего назначения; Trl, Tr2 – триггеры переносов соответственно арифметических и сдвиговых операций; АЛУ – арифметико-логическое устройство; $C_i$ , $C_{i+1}$ – соответственно входной и выходной переносы АЛУ; БФУС – блок формирования управляющих сигналов.



Кроме универсального АЛУ МП может содержать одно или несколько *специализированных* АЛУ. В качестве последних обычно используют блоки аппаратного умножения и деления, а также блоки для выполнения операций с плавающей точкой. Наличие специализированных АЛУ естественно увеличивает сложность СБИС МП, но за счет выполнения дополнительных операций на уровне регистровых передач производительность МП повышается. Важной составляющей ОБ современных МП является блок внутренней памяти, реализованный в виде набора программно доступных регистров, называемых регистрами общего назначения (РОН). Время обращения к РОН меньше, чем к любым другим устройствам памяти, поэтому память на РОН относится к сверхоперативной, а устройство, в виде которого она реализована, – сверхоперативным запоминающим устройством (СОЗУ). Число РОН в МП невелико (6-16), тем не менее их наличие существенно ускоряет выполнение операций. При наличии блока РОН операнды команд могут размещаться в одной из двух запоминающих сред – основной памяти или СОЗУ. Использование СОЗУ позволяет исключить значительную часть обращений МП к ОП через общую системную шину. С одной стороны, это повышает производительность за счет более быстрого обращения к СОЗУ, с другой стороны, появляется возможность параллельно с работой МП использовать системную шину для обмена информацией между другими устройствами ВМ. Используя специальные команды, пользователь может либо записывать информацию в РОН, либо считывать ее из РОН при выполнении различных арифметических и логических операций.

Число, назначение и разрядность регистров блока РОН в различных МП могут существенно отличаться. В использовании РОН имеются два крайних подхода. При первом подходе (реализуемом в МП компании Motorola при выпуске первых версий МП) почти все регистры МП выполняют абсолютно одинаковые функции, т.е. являются универсальными и взаимозаменяемыми. При втором подходе (характерном для первых версий МП компании Intel) многие регистры наряду с возможностью использования в качестве универсальных (если эта возможность вообще сохранена) в некоторых командах могут выполнять

специфические функции, закрепленные только за этими регистрами. Специализация регистров при выполнении наиболее часто используемых операций позволяет в соответствующих командах не указывать их адреса, что обеспечивает уменьшение необходимой для управления информации и более компактное кодирование команд. В частности, конкретные регистры МП Pentium используются в командах умножения и деления, управления циклами, ввода-вывода, при табличных преобразованиях, в стековых и сдвиговых операциях.

В большинстве ранних моделей МП один из общих регистров выделился в качестве главного регистра. Наделение главного регистра, называемого *аккумулятором*, или регистром результата, особыми функциями позволяло реализовать ОБ в виде одноадресного устройства. В таком ОБ один из исходных операндов арифметических и логических операций обязательно должен размещаться в аккумуляторе и в него же помещается результат. Другой операнд названных операций может находиться в памяти или РОН. Входные данные поступают в аккумулятор с внутренней шины МП, а аккумулятор, в свою очередь, может посылать данные на эту шину.

Функциональные возможности ОБ, содержащего аккумулятор (см. рис. 2.1), достаточно широки. С его помощью можно реализовывать различные операции. Рассмотрим основные микрооперации, с использованием которых реализуется выполнение команд.

Содержимое любого РОН или ячейки памяти по внутренней шине данных может быть передано через аккумулятор в буферный регистр или напрямую в регистр-сдвигатель. АЛУ обеспечивает выполнение арифметических и логических операций над содержимым регистра-сдвигателя и буферного регистра с записью результата в аккумулятор, а признаков – в регистр признаков. Из аккумулятора результат операции может быть передан в любой РОН или ячейку памяти. Операция сдвига содержимого любого РОН выполняется последовательно передачей слова из РОН в сдвигающий регистр, сдвига этого слова и последующей записи преобразованного слова в тот же регистр РОН.



Операции над словами с повышенной разрядностью реализуются путем программно последовательной обработки отдельных частей многоразрядных слов. Для обеспечения возможности обработки данных с разрядностью, превышающей разрядность АЛУ и регистров, в структуре ОБ, предусмотрены два дополнительных триггера: Trl и Тг2 (см. рис. 1). С их помощью осуществляется запоминание сигналов арифметического переноса из АЛУ и выходного бита переноса регистра сдвига. Например, с помощью 8-разрядного МП сравнительно просто осуществляется арифметическая обработка 24-разрядных слов. Для этого выполняют три цикла обработки 8-разрядных частей этих слов.

Признаки операций АЛУ (флаги), характеризующие результаты вычислений, запоминаются в одноименных флагах регистра признаков. Типичными признаками являются: *нулевой результат*, *наличие переноса*, *переполнение*, *четность*, *знак* и некоторые другие:

- *Флаг нулевого результата* ZF (Zero Flag) устанавливается в «1» при нулевом значении результата. При ненулевом результате ZF = 0.

- *Флаг переноса* CF (Carry Flag) запоминает значение переноса (заема) при сложении (вычитании) операндов. В некоторых МП флаг CF также используется для запоминания выдвигаемого бита при сдвиге операнда.

- *Флаг переполнения* OF (Overflow Flag) фиксирует переполнение разрядной сетки результата при выполнении операций со знаковыми числами. Переполнение происходит только тогда, когда коды слагаемых имеют одинаковые значения знаковых разрядов, а код суммы имеет другое значение знакового разряда. Для определения переполнения OVR используют логическую операцию XOR (*Исключающее ИЛИ*) над значениями переносов в знаковый разряд $C_{s-1}$ из знакового разряда $C_{s-1}$:

$$OVR = C_{s-1} \oplus C_s$$

- *Флаг четности или паритета* PF (Parity Flag) фиксирует наличие четного числа единичных разрядов в младшем байте результата операции. Флаг четности часто используют при контроле правильности передачи данных.

- *Флаг знака* SF (Sign Flag) дублирует значение старшего бита результата. Флаг SF при использовании дополнительного кода соответствует знаку числа.

В большинстве случаев признаки результата используются для программного управления последовательностями выполняемых команд при разветвлениях и циклах.

Среди РОН часто выделяют регистры, используемые в качестве базовых и индексных регистров. В ряде способов адресации содержимое указанных регистров участвует в формировании исполнительных адресов операндов в памяти.

*Блок управления.* В процессе выполнения программы блок управления (см. рис. 2.1) координирует работу всех блоков МП и микропроцессорной системы в целом. С помощью блока управления формируются управляющие сигналы, необходимые для организации обмена информацией с внешними устройствами, и обеспечивается выборка команд программы из памяти. В целом блок управления выполняет следующие действия:

- считывает и запоминает текущую команду;
- формирует адрес следующей команды;
- реализует выполнение по тактам алгоритма поступившей команды;
- управляет обменом информацией с внешними устройствами по системной шине.

Блок управления состоит из регистра команд (РгК), дешифратора команд (ДшК) и блока формирования управляющих сигналов (БФУС). Управляющие сигналы с выходов БФУС поступают на управляющие входы других блоков МП, настраивая их на выполнение определенных микроопераций. В состав блока управления также включают программно доступные счетчик команд PC (Program Counter) и указатель стека SP (Stack Pointer). Большинство элементов этого блока, кроме счетчика PC и указателя стека SP, являются программно недоступными. Счетчик PC (его другое название Instruction Pointer – указатель команд IP) предназначен для адресации команд программы. После выборки из



памяти очередной команды в регистре IP формируется адрес следующей по порядку команды. В командах условных и безусловных переходов, вызова подпрограмм и возврата из подпрограмм в регистр IP непосредственно загружается адрес перехода. Назначение и использование указателя стека рассматривается ниже.

Выполнение любой команды реализуется как последовательность трех фаз: выборка, декодирование и выполнение.

*Фаза выборки* обеспечивает считывание очередной команды из памяти и пересылку ее в МП. Адрес считываемой команды определяется содержимым программного счетчика PC. Любая команда, всегда содержит всю необходимую информацию о выполняемой операции и об ее операндах. Для указания этой информации команды МП имеют определенную структуру, называемую форматом или структурой команды. Форматы команд у различных типов МП в деталях отличаются. Общим является то, что структура команды состоит из двух частей: *кода операции* и *адресной части*. *Код операции* однозначно определяет тип выполняемой операции. *Адресная часть* указывает на ячейки памяти, к которым надо обратиться, выполняя команду. В ней содержится информация об адресах операндов и результата. В зависимости от типа команда может состоять из одного или нескольких байтов, при этом код операции всегда размещается в первом байте команды. Код операции текущей команды запоминается в РгК.

*В фазе декодирования* содержимое РгК с помощью ДшК преобразуется в управляющее слово. Схема синхронизации, используя это слово, вырабатывает совокупность сигналов, управляющих внутренними операциями МП и обменом информацией между МП и внешними устройствами.

После выборки и дешифрирования команды ОБ в декодированном виде получает информацию о том, какую операцию он должен выполнить, где в памяти расположены данные, куда следует направить результат операции и где расположена следующая команда. *В фазе выполнения* БФУС вырабатывает последовательности управляющих сигналов, обеспечивающих выполнение опера-

ции, заданной в команде. Адресная часть сохраняется в *Регистре адреса данных в памяти*.

При описании функций, реализуемых блоками МП, последние могут быть представлены как конечные автоматы (с памятью или без памяти) либо как группы взаимосвязанных конечных автоматов. Автоматы с памятью – синхронные. Их работа синхронизирована тактовыми импульсами. Частота этих импульсов (тактовая частота) характеризует быстродействие МП. Каждый блок за один такт может выполнить простейшую операцию. Для регистра – это запись кода или выдача хранимого кода на выходы. Для счетчика – запись кода в счетчик, прибавление или вычитание единицы, выдача хранимого кода. Для мультиплексора – передача на выход сигнала с входа, определяемого управляющим кодом. Для АЛУ – одна из простейших операций, которая выполняется на уровне регистровых передач: сложение, вычитание, сдвиг, реализация одной из поразрядных логических операций.

Элементарное действие, выполняемое в одном из узлов ОБ в течение одного такта, называется *микрооперацией*. В некоторые такты в различных узлах ОБ (регистрах, мультиплексорах, счетчиках и др.) одновременно могут выполняться несколько микроопераций. Настройка ОБ на выполнение одной из возможных микроопераций осуществляется с помощью сигналов, поступающих на его управляющие входы. Набор этих сигналов (вектор сигналов управления) вырабатывается БФУС. Совокупность одновременно выполняемых микроопераций соответствует микрокоманде. *Команда* в общем случае состоит из нескольких простейших действий и обычно выполняется за несколько тактов, т. е. требует для своего выполнения нескольких микрокоманд. Последовательность микрокоманд, обеспечивающая выполнение команды, называется *микропрограммой команды*.

Так как выполнение любой команды по тактам на уровне регистровых передач реализуется микропрограммой команды, управляющий автомат, формирующий наборы управляющих сигналов, называют микропрограммным автоматом (МПА). В соответствии с реализуемой логикой управления МПА подразделяют на МПА с «жесткой» логикой управления и МПА с «мягкой» логикой управления. Термин «микропрограммный автомат», вначале использовавшийся только применительно к управляющим автоматам с «мягкой» логикой, с начала 80-х гг. XX в. используется и для автоматов с «жесткой» логикой.



МПА с «жесткой» логикой управления представляется в виде конечного автомата (в сложных случаях используется сеть конечных автоматов). Алгоритм команды в МП с подобным МПА задан жестко соединениями схемы, поэтому список операций (система команд) МП является неизменным, и это не позволяет вносить какие-либо изменения в систему команд МП после его изготовления. Основным достоинством МПА с «жесткой» логикой» является его высокое быстродействие. Такие устройства управления используются в большинстве однокристальных МП, отличительной характеристикой которых является то, что все элементы структуры МП выполнены на одном кристалле.

При реализации МПА с «мягкой» логикой управления используется типовая структура автомата с блоком памяти микропрограмм, который, как и основная память, имеет линейно-адресную организацию. До появления технологии СБИС МПА с «мягкой» логикой управления имели преимущественное распространение. И в основном это было связано с тем, что при ограниченных возможностях технологии производства ИС и средств автоматизации проектирования МПА с «мягкой» логикой управления позволяли эффективно использовать блоки памяти с регулярной структурой и хранения микропрограмм команд проектируемой системы. Ясное представление микропрограмм команд в памяти облегчает внесение изменений в алгоритмы управления. В целом разработка УА МПА с «мягкой» логикой управления в значительной мере определяется разработкой микропрограмм команд. Следует отметить, что использование такого подхода внутри однокристально МП не является предпочтительным, так как приводит к снижению быстродействия. При необходимости изменения команд таких МП вносят изменения в схемные реализации алгоритмов команд, что при развитых средствах автоматизированного проектирования не представляет труда.

В современных МП в зависимости от системы команд и сложности алгоритмов микропрограмм команд используются оба подхода при построении МПА. На уровне регистровых передач с помощью МПА независимо от типа управления непосредственно осуществляется интерпретация машинных команд программы и их выполнение. Именно поэтому уровень регистровых передач также называют микропрограммным уровнем организации вычислительных процессов ВМ. Отметим, что выполнение программ на командном уровне является более высоким уровнем организации вычислительного процесса.

*Интерфейсный блок*. Предназначен для организации взаимодействия МП (см. рис. 2.1) с памятью и устройствами ввода-вывода, расположенными на системной шине процессора, а также для обмена данными между ОБ и внутренними устройствами МП. Непосредственное подсоединение устройств ввода-вывода к МП осуществляется с помощью специальных схем сопряжения, которые называются *интерфейсом ввода-вывода*. В общем случае интерфейсный блок процессора должен выполнять следующие функции:

- формировать выходные сигналы на шинах адреса, данных и управления в режиме вывода;
- формировать выходные сигналы адреса и управления и считывать (воспринимать) сигналы с шины данных в режиме ввода;
- синхронизировать процессы внутри процессора и на системной шине;
- реализовывать стандартный для системной шины протокол обмена.

Системная шина (СШ) объединяет сигналы шин данных, адреса и управления. Протокол обмена информацией по СШ определяет последовательности сигналов (временную диаграмму сигналов в шине), обеспечивающих правильную передачу информации между устройствами микропроцессорной системы.

Электрические спецификации сигналов, действующих на линиях СШ, определяют гарантированные уровни электрических напряжений, идентифицирующих логические состояния «0» и «1», и их нагрузочную способность. Одной из особенностей компонентов ВМ, обеспечивающей возможность их электрического сопряжения друг с другом и разнообразными периферийными



устройствами, является TTL-совместимость. Выбор такого стандарта для интерфейса определяется широким распространением разнообразных устройств на TTL-схемах. Выходные сигналы МП являются маломощными (их нагрузочная способность эквивалентна одному входу TTL-схем). Для согласования выходных сигналов МП с нагрузочными входами внешних устройств используют специальные усилители, в частности шинные формирователи.

Взаимодействие по шине (регистровая пересылка между одним из регистров ОБ процессора и ячейкой ОП либо портом периферийного устройства) осуществляется за *цикл шины*. Длительность цикла шины может изменяться в зависимости от быстродействия внешних устройств. Для согласования по быстродействию взаимодействующих при обмене по шине устройств используется принцип *квитирования*. Суть его в том, что процессор, управляющий обменом, в каждом цикле ждет уведомления (квитанции) о том, что устройство на шине выполнило операции, связанные с обменом, т.е. выставило на шину данные при вводе либо восприняло данные с шины при выводе. Для уведомления используется сигнал «Готовность» (Ready), передаваемый от внешнего устройства в процессор по одной из линий шины управления. Управление обменом осуществляет автомат, входящий в состав интерфейсного блока процессора.

## Особенности организации процессоров

Способ построения и состав внутренней памяти МПС оказывает существенное влияние на организацию самого МП. В соответствии с классификационным признаком по числу и способу использования регистров блока внутренней памяти различают следующие типы МП: *аккумуляторные*, *многоаккумуляторные* и *стековые*.

1. МП с одним регистром результата иногда называют МП *с аккумуляторной архитектурой* или просто *аккумуляторными* МП. Их отличительной характеристикой является относительная простота аппаратной реализации операционного и управляющего блоков, а также упрощенный формат команд (пример – простейшие i8080, КР580). В командах аккумуляторного МП адрес

первого операнда не указывается, а адресуется только второй операнд. Результат записывается в аккумулятор. Необходимость предварительной загрузки операнда в аккумулятор перед выполнением любой арифметической или логической операции, а также невозможность непосредственной записи результата выполнения команды в произвольную ячейку памяти или регистр в ряде случаев приводят к увеличению числа команд и времени выполнения программы. Отмеченные недостатки аккумуляторной архитектуры ограничивают возможности и области ее применения.

2. В большинстве современных МП специальные регистры результатов не используются, а их функции может выполнять любой регистр блока РОН или ячейка памяти. МП с подобной архитектурой часто называют *многоаккумуляторными*. В командах многоакумуляторного МП оба операнда задаются явно, а результат операции и чаще всего помещается на место одного из операндов.

3. *Стековые МП*. *Стек* – это память с линейно упорядоченными ячейками и специальным механизмом доступа, исключающим необходимость указания адреса при записи и чтении. В зависимости от используемого правила доступа, называемого *дисциплиной*, различают два типа организации стековой памяти: *очередь* и *стек*. Дисциплина определяется применительно к входным (записываемым) и выходным (читаемым) последовательностям слов. Очередь реализует дисциплину FIFO (First-In-First-Out – первый поступивший удаляется первым). Стек в отличие от очереди организован в соответствии с дисциплиной LIFO (Last-In-First-Out – последний поступивший извлекается первым), т.е. информация из стека выбирается в обратном по отношению к записи порядке.

*Примечание*. В МП системах стековую организацию достаточно часто используют. Рассмотрим механизм доступа к стеку более подробно.

Физически стек представляет собой набор регистров (аппаратурный стек) или ячеек оперативной памяти, снабженный указателем стека SP см. рис. 2.2.

Указатель SP, в качестве которого используют реверсивный счетчик, всегда адресует «вершину стека», под которой понимается ячейка стека, доступная для чтения.



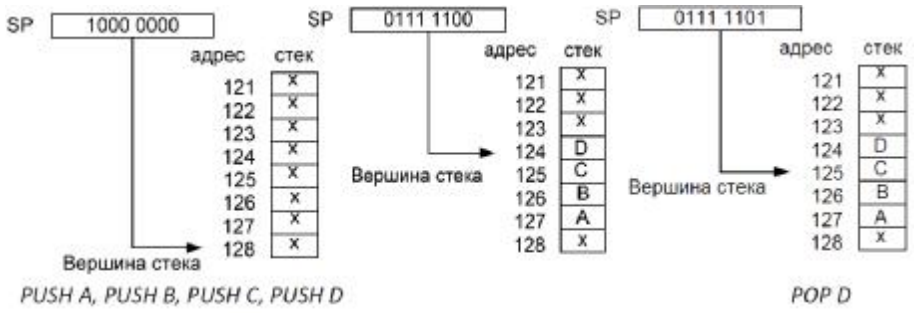



*Рис. 2.2.* Механизм доступа к стеку

По мере записи и считывания данных из стека содержимое указателя SP меняется: при записи или загрузке, например при исполнении команд *PUSH*, значение указателя SP уменьшается – стек растет в сторону младших адресов, а при чтении или выталкивании данных из стека, в частности при исполнении команд *POP*, значение указателя SP увеличивается. Указанное правило при обращении к стеку реализуется автоматически, и поэтому при операциях со стеком возможно *безадресное* задание операнда. Стековая память является безадресной.

Важнейшей характеристикой стека является его размер. МП может содержать относительно небольшой по размеру аппаратный стек (число внутренних регистров стековой памяти, как правило, не превышает 8-16) или не содержать такового совсем. Более распространена архитектура МП, использующая практически неограниченный внешний стек, моделируемый в ОП с произвольным доступом.

Стек играет важную роль в микропроцессорных системах как средство сохранения адресов возврата и состояния данных при работе с подпрограммами. Использование стека приводит к существенным упрощениям при организации вложенных подпрограмм, когда одна программа вызывает другую, которая в свою очередь может вызвать третью и т.д. В таких случаях при каждом вызове адрес возврата текущей программы и другая необходимая информация (со-

держимое РОН) загружаются в стек. При возврате информация в обратном порядке выбирается из стека. Заметим, что при организации стека, моделируемого в памяти с произвольным доступом, время обращения к элементам данных стека равно времени обращения к памяти. Однако стек наряду с отмеченными особенностями его использования эффективнее обычной памяти. Во-первых, используемые при обращении к стеку команды PUSH и POP короче стандартных команд обращения к памяти, так как в них один из операндов неявно адресуется через регистр SP, и, во-вторых, инкремент или декремент указателя SP с образованием нового адреса производится автоматически.

В современных процессорах стек применяется для разных целей: временного хранения данных, когда для них нет смысла выделять фиксированные места в памяти; организации прерываний, вызовов процедур и возвратов из них; передачи и возврата параметров при вызовах процедур. Кроме того, стековая память является важнейшей компонентой процессоров со стековой архитектурой.

*В стековом процессоре* обычно используется относительно большой аппаратный стек (регистровый файл объемом минимум 64 регистра) и дополнительный внешний стек в памяти, обращение к которому осуществляется только в случаях нехватки памяти аппаратного стека, что случается сравнительно редко. Благодаря специальному размещению операндов в стеке арифметико-логическую обработку информации в стековом процессоре можно выполнять полностью безадресными командами, содержащими только код операции. В соответствии с кодом операции такие команды извлекают из стека один или два операнда, выполняют над ним предписанную операцию и заносят результат в вершину стека, Безадресные команды на основе стековой адресации предельно сокращают формат команд, экономят память и способствуют повышению производительности ВМ. Недостатком стековых процессоров является необходимость специальной подготовки данных участвующих в вычислениях, с использованием адресных команд для загрузки стека.

Структуру стекового процессора имеет специализированный арифметический сопроцессор (блок вычислений с плавающей точкой FPU) современных



МП Pentium. Чтобы пояснить его назначение, кратко характеризуем предысторию его появления.

В связи с ограничениями на сложность схем ОБ младшие модели МП могли выполнять только операции над числами с фиксированной точкой с ограниченной длиной слова. Обработка чисел повышенной разрядности и чисел в формате с плавающей точкой в таких МП реализовывались программно. Это приводило к значительному снижению производительности. Введение стандарта IEEE 754 на численные данные и особенности их обработки усложнило форматы представления чисел. В частности, в 80-битном расширенном формате вещественных чисел (рис. 2.3) 1 бит выделяется для знака числа *(S)*, 15 бит отводятся для порядка *(E)* и 64 бит – для указания его мантиссы *(M)*.

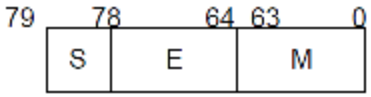



*Рис. 2.3.* Расширенный формат вещественных чисел.

Для эффективной реализации операций над числами в стандартных форматах в конце 70-х гг. XX века был предложен принцип специализации. Суть его заключается в разработке специализированных процессорных модулей (сопроцессоров) со своими системами команд, которые ориентированы на конкретные приложения. Арифметический сопроцессор с плавающей точкой является примером такого модуля. Сопроцессоры обычно работают под управлением центрального процессора совместно используют ОП. Сопроцессор не имеет своей отдельной программы и не может считывать команды из памяти, однако он может обращаться к памяти для записи и считывания данных. Часть кодов команд центрального процессора резервируется для команд сопроцессора. Сопроцессор не может выполнять команды центрального процессора, но свои команды выполняет очень быстро. В процессе работы и процессор, и сопроцессор из общего командного потока выбирают свои команды и выполняют их. Сопроцессор выполняет свои «долгие» арифметические операции параллельно с центральным процессором, при этом последний может продолжать выполнение потока собственных команд. Такое архитектурное решение фирма Intel приме-

нила на семействе программно совместных МП 8086, 80286, 80386, 80486. Обобщенно архитектуру процессов этого семейства обозначим x86. Для младших членов семейства сопроцессоры были реализованы в виде специализированных СБИС – 8087 (для 8086), 80387 (для 80386). Благодаря росту уровня интеграции микросхем в МП i486 и всех моделях Pentium арифметический сопроцессор реализован в виде устройства операций с плавающей точкой FPU как составная часть процессора, т. е. является внутренним блоком МП. Размещение устройства FPU на кристалле- процессора значительно повышает производительность. Совместно с центральным процессором сопроцессор образует мощный тандем, производительность которого в системах численной обработки в 10-50 раз выше производительности одного центрального процессора. С точки зрения программиста, система с сопроцессором выглядит как единый процессор с расширенными возможностями ОБ, большим набором команд, форматов чисел и числом регистров. Взаимодействие между центральным процессором и FPU невидимо для пользователя.

## *Организация памяти*

*Память* (ПАМ) – один из блоков ЭВМ, состоящий из запоминающих устройств (ЗУ) и предназначенный для запоминания, хранения и выдачи информации. Основными характеристиками отдельных устройств памяти являются емкость памяти, быстродействие и стоимость хранения единицы информации (бита).

*Быстродействие* (задержка) памяти определяется временем доступа и длительностью цикла памяти. *Время доступа* представляет собой промежуток времени между выдачей запроса на чтение и моментом поступления запрошенного слова из памяти. *Длительность цикла* памяти определяется минимальным временем между двумя последовательными обращениями к памяти.

Требования к увеличению емкости и быстродействия памяти, а также к снижению ее стоимости являются *противоречивыми*. Чем больше быстродействие, тем технически труднее достигается и дороже обходится увеличение емкости памяти. Стоимость памяти составляет значительную часть общей стоимости ВМ.



## Уровни памяти

1. Сверхоперативный уровень – регистры управляющих и операционных блоков процессора (регистровая), сверхоперативная память, управляющая память, буферная память (кэш-память).

*Регистровая ПАМ* – предназначена для временного хранения информации. Реализуется на основе РОН. Данная память имеет малую емкость и наибольшее быстродействие.

*Сверхоперативная ПАМ* – способ организации регистровой ПАМ в виде единого блока. Используется в некоторых ВМ (например стековых МП).

*Управляющая ПАМ* – предназначена для хранения управляющих микропрограмм процессора. Выполнена в виде постоянного ПЗУ или (ППЗУ). Применяется для хранения однажды записанных микропрограмм, управляющих программ, констант и т.п.

*Кэш-память* – буферное ЗУ, размещённое между основной (оперативной) памятью и процессором. Предназначена для кратковременного хранения и выдачи активной информации процессору, что сокращает число обращений к основной памяти, скорость работы которой меньше, чем кэш-памяти. Программно недоступна. Бывает первого уровня (интегрированная, для работы с командами) и второго уровня (промежуточная для работы с ОП).

2. Оперативной уровень. Служит для хранения активных программ и данных, то есть тех программ и данных, с которыми работает ВМ. Служит для хранения кодов программы, операндов, промежуточных и конечных результатов. Имеет сравнительно большую емкость и высокое быстродействие, однако меньшее, чем ЗУ сверхоперативного уровня. В некоторых системах оперативный уровень снабжается собственным КЭШем.

3. Внешний уровень. Используется для хранения больших массивов информации в течение продолжительного времени. Обычно не имеет непосредственной связи с процессором. Обмен информацией носит групповой характер,

что значительно сокращает время обмена. Обладает сравнительно низким быстродействием (поиск информации). В качестве носителя используются магнитные диски (гибкие и жёсткие), лазерные диски (CD, DVD) и т.п.

Замечания

Рост производительности ВМ проявляется в первую очередь в увеличении скорости работы процессора. Быстродействие ПАМ также растет, но все время отстает от быстродействия аппаратных средств процессора потому, что одновременно происходит опережающий рост ее емкости, что делает более трудным уменьшение времени цикла работы памяти. Для выравнивания их пропускных способностей и предназначена сверхоперативная буферная память небольшой емкости и повышенного быстродействия. При обращении к блоку данных, находящемуся на оперативном уровне, его копия пересылается в сверхоперативную буферную память. Последующие обращения к этому блоку данных производится к буферной памяти.

1. Сверхоперативный уровень: внутренняя память процессора.

Частично организация данного уровня рассмотрена выше, и сюда можно отнести вопросы организации процессоров при использовании внутренних регистров.

Помимо регистровой памяти к данному уровню относят кэш-память, которую и рассмотрим более подробно.

В функциональном отношении кэш-память рассматривается как буферное ЗУ, размещенное между основной (оперативной) памятью и процессором. Основное назначение кэш-памяти - кратковременное хранение и выдача активной информации процессору, что сокращает число обращений к основной памяти, скорость работы которой меньше, чем кэш-памяти.

За единицу информации при обмене между основной памятью и кэш-памятью принята строка, причём под строкой понимается набор слов, выбираемый из оперативной памяти при одном к ней обращении. Хранимая в оперативной памяти информация представляется, таким образом, совокупностью строк с последовательными адресами. В любой момент времени строки в кэш-памяти представляют собой копии строк из некоторого их набора в ОП, однако расположены они необязательно в такой же последовательности, как в ОП.



Построение кэш-памяти может осуществляться по различным принципам, которые будут рассмотрены ниже.

Типовая структура кэш-памяти представлена на рис. 2.4.

*Рис. 2.4.* Типовая структура кэш-памяти

Строки, составленные из информационных слов, и связанные с ними адресные теги (вспомогательная информация об адресе, представляющая в общем случае целое двоичное число: тип, длина и т.д.) хранятся *в накопителе*, который является основой кэш-памяти.

Адрес требуемого слова, поступающий от центрального процессора (ЦП), вводится *в блок обработки адресов*, в котором реализуются принятые в данной кэш-памяти принципы использования адресов при организации их сравнения с адресными тегами.

Само сравнение производится в *блоке сравнения адресов* (БСА), который конструктивно совмещается с накопителем, если кэш-память строится по схеме ассоциативной памяти. Назначение БСА состоит в выявлении попадания или промаха при обработке запросов от центрального процессора. Если имеет место кэш-попадание (т.е. искомое слово хранится в кэш-памяти, о чём свидетель-

ствует совпадение кодов адреса, поступающего от центрального процессора, и одного из адресов некоторого адресного тега), то соответствующая строка из кэш-памяти переписывается *в регистр строк*.

С помощью *селектора-демультиплексора* из строки данных выделяется искомое слово, которое и направляется в центральный процессор. В случае промаха с помощью *блока формирования запросов* осуществляется инициализация выборки из ОП необходимой строки. Адресация ОП при этом производится в соответствии с информацией, поступившей от центрального процессора. Выбираемая из памяти строка вместе со своим адресным тегом помещается в накопитель и регистр строк, а затем искомое слово передается в центральный процессор.

Для высвобождения места в кэш-памяти с целью записи выбираемой из ОП строки одна из строк удаляется. Определение удаляемой строки производится посредством *блока замены строк*, в котором хранится информация, необходимая для реализации принятой стратегии обновления находящихся в накопителе строк.

Способы размещения данных в кэш-памяти делятся на четыре вида: прямое распределение, полностью ассоциативное, частично ассоциативное, распределение секторов.

Рассмотрим подробно каждый способ размещения и механизмы преобразования адресов. Предположим, что кэш содержит 128 строк, размер строки 16 слов, а основная память может содержать 16384 строки. Для адресации основной памяти используется 18 бит. Из них 14 старших показывают адрес строки, а младшие 4 – адрес слова внутри этой строки. Строки Кэш-памяти указываются 7-разрядными адресами.

а) *Прямое распределение*.

При прямом распределении место хранения строк в кэш-памяти однозначно определяется по адресу строки. Способ прямого распределения реализуется довольно просто, однако из-за того, что место хранения строки в кэш-памяти однозначно определяется по адресу строки, вероятность сосредоточения



областей хранения строк в некоторой части кэш-памяти высока, то есть замены строк будут происходить довольно часто. В этой ситуации эффективность кэш-памяти заметно снижается.

б) *Полностью ассоциативное распределение*.

При таком способе размещения данных каждая строка основной памяти может быть размещена на месте любой строки кэш-памяти. При полностью ассоциативном распределении механизм преобразования адресов должен давать ответ на вопрос, существует ли копия строки с произвольным адресом в кэш-памяти, и, если существует, то по какому адресу. По этому способу при замене строк кандидатом на удаление могут быть все строки в кэш-памяти.

в) *Частично ассоциативное распределение*.

При данном способе размещения, несколько соседних строк (фиксированное число, не менее двух) из всех строк кэш-памяти образуют структуру называемую группой. При пересылке новой строки в кэш-память удаляемая из нее строка выбирается из четырех строк соответствующего набора (группы).

г) *Распределение секторов*.

По этому способу основная память разбивается на секторы, состоящие из фиксированного числа строк, кэш-память также разбивается на секторы, состоящие из такого же числа строк. При данной организации кэш-памяти, распределение секторов в кэш-памяти и основной памяти осуществлено полностью ассоциативно, то есть, каждый сектор A основной памяти может соответствовать любому сектору D в кэш-памяти. К каждой строке V, хранящейся в кэш-памяти, добавляется один бит достоверности (действительности); он показывает, совпадает или нет содержимое этой строки с содержимым строки в основной памяти, которая в данный момент анализируется на соответствие строки кэш-памяти. Если слова, запрашиваемого центральным процессором при доступе, не существует в кэш-памяти (бит достоверности, выбранный по адресу равен 0), то сначала центральный процессор проверяет, был ли сектор A, содержащий это слово, помещен ранее в кэш-память. Если он отсутствует, то один из секторов кэш-памяти заменяется на этот сектор.

Если все сектора кэш-памяти используются, то выбирается один какой-нибудь сектор, и при необходимости только некоторые строки этого сектора возвращаются в основную память, а этот сектор можно использовать дальше.

Методы организации кэш-памяти напрямую связаны с условиями выборки и хранения информации в ней – см. в табл. 2.1.

Таблица 2.1. Условия сохранения и обновления информации

| Режим работы | Наличие копии ячейки ОП в кэш-памяти | Информация в ячейке кэш-памяти | Информация в ячейке ОП |
|---|---|---|---|
| Чтение | Копия есть | Не изменяется | Не изменяется |
| | Копии нет | Обновляется (создается копия) | Не изменяется |
| Сквозная запись | Копия есть | Обновляется | Обновляется |
| | Копии нет | Не изменяется | Обновляется |
| Обратная запись | Копия есть | Обновляется | Не изменяется |
| | Копии нет | Обновляется | Не изменяется |

Если процессору требуется информация из некоторой ячейки основной памяти, а копия этой ячейки уже есть в кэш-памяти, то вместо оригинала считывается копия. В этом случае информация ни в кэш-памяти, ни в основной памяти не изменяется.

При записи строк существует несколько методов обновления старой информации. Эти методы называются стратегией обновления строк основной памяти. Если результат обновления строк кэш-памяти не возвращается в основную память, то содержимое основной памяти становится неадекватным вычислительному процессу. Чтобы избежать таких ошибок, предусмотрены различные методы обновления основной памяти.

2. Оперативный уровень.

*Оперативная память* (system memory, ОП) - имеет среднюю емкость. Работает на частоте системной шины. Время доступа к оперативной памяти со-



ставляет меньше к жестким магнитным дискам. Доступ процессора к оперативной памяти происходит через кэш 2-го уровня. Некоторые подсистемы компьютера способны обращаться к оперативной памяти напрямую, минуя процессор.

Базовые типы ОП:

- DRAM (Dynamic Random Access Memory) - динамическое запоминающие устройство случайного доступа. Единичная ячейка памяти напоминает конденсатор с диодной конструкцией. Наличие или отсутствие заряда конденсатора соответствует единице или нулю. Основной вид, применяемый для оперативной памяти, видеопамяти, а также различных буферов и кэшей более медленных устройств. По сравнению со SRAM заметно более дешевая, хотя и более медленная по двум причинам - емкость заряжается не мгновенно, и, кроме того, имеет ток утечки, что делает необходимой периодическую подзарядку.

- SRAM (Static RAM) - статическая память, разновидность памяти, единицей хранения информации в которой является состояние «открыто-закрыто» в транзисторной сборке. Используется преимущественно в качестве кэш-памяти 2-го уровня. Ячейка SRAM более сложна по сравнению с ячейкой DRAM, поэтому более высокое быстродействие SRAM компенсируется высокой ценой. Несмотря на низкое энергопотребление, является энергозависимой, то есть при отключении питания информация теряется.

*Примечание.* ОП является наиболее дефицитным и наиболее важным ресурсом в вычислительных машинах и системах. Проблема управления ОП усложняется при переходе к мультипрограммным системам, т. к. в них ОП используют одновременно несколько вычислительных процессов. Для идентификации переменных и команд используют символьные имена (метки), виртуальные адреса и физические адреса.

*Методы управления ОП:* методы распределения без использования виртуальной памяти – ВП (фиксированными, динамическими или перемещаемыми разделами); методы распределения памяти с использованием дискового про-

странства – виртуальной памяти ВП (страничное, сегментное, странично-сегментное распределение, свопинг).

*Методы управления без использования ВП – распределение памяти фиксированными разделами.* Наиболее простой способ. Вся ОП делится на определенное число разделов фиксированной величины. Очередной процесс (или задача) поступившая на выполнение, становится в общую очередь или в очередь к подходящему по размеру разделу памяти (Когда раздел освобождается, очередной процесс (программа) подгружается в ОП).

В этом случае подсистема управления памятью выполняет задачи:

- сравнения размера поступившей на выполнение программы с размерами свободных разделов памяти;

- выбора подходящего раздела;

- загрузка программы и настройка адресов.

При очевидном преимуществе – простоте реализации этот способ распределения имеет очевидный минус: жесткость. Во-первых, одна программа занимает весь раздел целиком. Это ведет к тому что, во-первых, неэкономно расходуется память; во-вторых, коэффициент мультипрограммирования ограничен числом разделов. С другой стороны, даже если суммарный объем свободной ОП машины позволяет выполнять некоторую программу, разбиение памяти на разделы не позволяет сделать этого.

*Методы управления без использования ВП – распределение памяти динамическими (переменной величины) разделами.* При таком способе распределения в начале работы ВМ вся ОП свободна. Каждой поступающей на выполнение задаче выделяется необходимый ей объем ОЗУ. Если достаточный объем памяти отсутствует, задача не принимается на выполнение и стоит в очереди. После завершения задачи память освобождается, и на это место может быть загружена другая задача. Т.е., в произвольный момент времени ОП представляет собой случайную последовательность свободных и занятых областей памяти. Правила выбора соответствующего раздела при загрузки задачи: первый попавшийся достаточного размера, раздел с наименьшим достаточным размером.



Подсистема управления памятью выполняет задачи:

- ведение таблиц свободных и занятых областей, в которых указывается начальные адреса и размеры участков памяти;

- анализ запроса (при поступлении новой задачи);

- просмотр таблицы свободных областей (с целью выбора раздела для размещения вновь поступившей задачи);

- загрузка задачи в выделенный ей раздел;

- корректирование таблиц свободных и занятых областей (как после загрузки очередной задачи в ОП, так и после завершения задачи).

По сравнению с методом распределения памяти фиксированными разделами данный метод более гибок, но ему присущ серьезный недостаток – фрагментация памяти. Фрагментация – это наличие многих несмежных областей памяти, настолько маленьких по размеру, что ни в одну из них нельзя поместить ни одну из пришедших на выполнение программ, хотя суммарный объем таких фрагментов может составить значительную величину.

*Методы управления без использования ВП – распределение памяти с перемещаемыми разделами.* Борьба с фрагментацией может быть осуществлена путем перемещения всех занятых участков в сторону старших либо в сторону младших адресов так, чтобы все свободные участки памяти составляли единую область. В дополнение к функциям, которые выполняет ОС при распределении памяти переменными разделами, в данном случае она должна еще время от времени копировать содержимое разделов из одного места памяти в другое, корректируя таблицы свободных и занятых областей. Такая процедура называется сжатием и выполняется ОС в дополнение к функциям, которые ОС выполняет при распределении ОП переменными разделами.

Сжатие может выполняться либо при каждом завершении задачи, либо только тогда, когда для вновь поступившей задачи нет свободного раздела достаточного размера. В первом случае требуется меньше вычислительной работы при корректировке таблиц, а во втором - реже выполняется процедура сжатия. Хотя процедура сжатия и приводит к более эффективному использованию

памяти, она может потребовать значительного времени, что может свести на нет преимущества данного метода.

*Методы управления с использованием ВП – организация виртуальной памяти.* Виртуальная память возникла как средство решения проблемы размещения в ОП программ, размер которых значительно превышает имеющуюся в наличии свободную память.

Виртуальным называют такой ресурс, который для пользователя представляется обладающим теми свойствами, которыми он в действительности не обладает. Виртуальная память (ВП) – это совокупность программно-аппаратных средств, позволяющих пользователям писать программы, размер которых превосходит имеющуюся ОП.

Для этого ВП решает следующие задачи:

- размещает данные в ЗУ разного типа, например, часть программы в ОП, а часть на жестком диске;

- перемещает данные по мере необходимости между ЗУ разного типа, например, подгружает нужную часть программы с диска в ОП;

- преобразует виртуальные адреса в физические.

Все эти действия выполняются без участия пользователя.

*Методы управления с использованием ВП – страничное распределение.* Виртуальное адресное пространство каждого процесса делится на части фиксированного размера, называемые *виртуальными страницами*. Размер виртуального адресного пространства в общем случае не является кратным размеру страницы, поэтому последняя страница каждого процесса дополняется фиктивной областью. Вся ОП ВМ делится на части такого же размера, называемые физическими страницами (или блоками). Размер страницы обычно выбирается равным степени двойки, т.к. это позволяет упростить механизм преобразования адресов. Часть виртуальных страниц процесса при его загрузке помещается в ОП, а часть на жесткий диск.

ОС при загрузке процессора формирует для него отдельную информационную структуру – таблицу страниц, в которой устанавливается соответствие



виртуальных страниц с определенными номерами физическим страницам с определенными номерами. Также в таблице страниц содержится управляющая информация: признак модификации страницы, признак невыгружаемости (выгрузка некоторых страниц может быть запрещена), признак обращения к странице (используется для подсчета обращений к странице за определенный период времени), а также некоторые другие данные создаваемые и используемые механизмом ВП.

При каждом обращении к памяти происходит чтение из таблицы страниц информации о виртуальной странице, к которой произошло обращение. Если данная виртуальная страница находится в ОП, выполняется преобразование виртуального адреса в физический. Если же нужная виртуальная страница отгружена на диск, то генерируется т.н. случай страничного прерывания. Процесс переводится в состояние ожидания, и активизируется другой процесс, из очереди готовых. Параллельно с этим программа обработки страничного прерывания находит на диске требуемую страницу и попытается загрузить ее в ОП. Если в памяти имеется свободная фиксированная страница, то соответствующая виртуальная страница подгружается в ОП, если же свободных страниц нет, то реализуется процедура выталкивания из ОП какой-нибудь страницы. Критерии выбора выталкиваемой виртуальная страница могут быть следующими:

- дольше всего не использовавшаяся страница;

- первая попавшаяся (случайная) страница;

- страница, к которой было меньше всего обращений.

После того, как выбрана страница, которая должна быть удалена из ОП, анализируется ее признак модификации. Если выталкиваемая страница с момента загрузки процесса была модифицирована, то ее модификация должна быть переписана на диск, если же нет, то соответствующая физическая страница делается свободной, а виртуальная страница просто уничтожается.

В некоторых системах используется механизм рабочего множества страниц для каждого процесса. Это рабочее множество – перечень наиболее часто

используемых страниц, которые постоянно находятся в ОП и выгрузка которых запрещена.

На производительность системы со страничной организацией памяти влияют временные затраты, связанные с обработкой страничных прерываний и преобразованием виртуального адреса в физический. Чем чаще возникает страничное прерывание, тем больше времени тратится на перемещение страниц. Чтобы уменьшить частоту страничных прерываний, нужно увеличить размер страницы. Увеличение размера страницы уменьшает размер таблицы страниц, а значит, и уменьшает затраты памяти. Но, с другой стороны, чем больше размер виртуальной страницы, тем больше памяти занимает фиктивная область в конце последней виртуальной страницы каждой программы.

*Методы управления с использованием ВП – сегментное распределение.* При страничной организации памяти – пространство процесса делится на механически равные части. Это не позволяет дифференцировать способы доступа к разным частям программы (сегментам), а это свойство часто бывает полезным. Например, можно запретить обращаться с операцией записи и чтения в сегмент кода программы, а для сегмента данных разрешить только чтение. Кроме того, разбиение программы на сегменты, а не на страницы делает возможным разделение одного сегмента несколькими процессами.

В случае сегментного распределения виртуальное адресное пространство процесса делится на сегменты, размер которых определяется программистом с учетом смыслового значения содержащейся в них информации. Отдельным сегментом м.б. подпрограммы, массив данных и др. Иногда сегментация программы может выполняться по умолчанию компилятором.

При загрузке процесса одни сегменты записываются в ОП, другие остаются на диске. Сегменты одной программы могут занимать несмежные участки ОП. Во время загрузки ОС создает для процесса таблицу сегментов (аналогичную таблице страниц), в которой для любого сегмента указывается начальный физический адрес сегмента в ОП, размер, правила доступа, признак модификации, признак обращения к данному сегменту за последний интервал времени и



некоторая другая информация. Если виртуальная адресация нескольких процессов включают один и тот же сегмент, то в таблицах сегментов этих процессов делаются ссылки на один и тот же участок ОП, в который этот сегмент загружается в единственном экземпляре.

Система с сегментной организацией функционирует аналогично системе со страничной организацией. При каждом обращении к ОП из таблицы сегментов считывается информация о сегменте, к которому производится обращение. Если сегмент находится в ОП, то осуществляется преобразование виртуального адреса в физический. Если же сегмент находится на диске, то возникает случай страничного прерывания; процесс переводится в состояние ожидания, и активизируется другой процесс. Аналогичным образом при необходимости освобождения ОП некоторые сегменты выгружаются. Вместе с тем, при обращении к памяти проверяется, разрешен ли доступ требуемого типа к данному сегменту.

К недостаткам сегментной организации можно отнести более медленную работу и фрагментацию на уровне сегментов.

*Методы управления с использованием ВП – странично-сегментное распределение.* Данный метод сочетает в себе достоинства обоих подходов, представляя собой их комбинацию. Всё виртуальное адресное пространство процесса делится на сегменты, а каждый сегмент – на страницы. ОП при этом делится на физические страницы. Загрузка процесса в ОП осуществляется постранично. ОС при загрузке процесса создает таблицу сегментов процесса. Для каждого сегмента создается своя таблица страниц, структура которой полностью аналогична структуре таблицы страниц при страничном распределении. Адрес таблицы сегментов загружается в специальный регистр процессора, когда активизируется соответствующий процесс. В таблице страниц находятся адреса таблиц страниц для всех сегментов данного процесса.

*Методы управления с использованием ВП – свопинг.* Разновидностью виртуальной памяти является свопинг. Для того чтобы задача могла начать выполняться, она должна быть загружена в ОП, объем которой ограничен.

Чтобы увеличить уровень мультипрограммирования, и был предложен метод организации вычислительного процесса, называемый *свопингом*. В соответствии с этим методом некоторые процессы (задачи), обычно находящиеся в состоянии ожидания, целиком могут отгружаться (откачиваться) на диск, а на их место подгружаться другие. При этом программа-планировщик ОС не исключает их из своего рассмотрения и при наступлении условий, в которых возможно выполнять некоторую задачу, находящуюся в области свопинга на диске, эта задача перемещается в ОП. Существуют различные алгоритмы отгрузки процессов на диск и подкачивания других процессов, а также различные способы выделения оперативной и дисковой памяти загружаемому процессу.

3. Внешний уровень.

*Системы внешней памяти* - это hard disc, то есть жесткий диск (HDD), floppy disc (FDD), то есть гибкий диск или дискета, CD-ROM и некоторые другие.

*Жесткий диск,* или винчестер (название пошло от совпадения обозначения первого жесткого диска с обозначением винтовки XIX века фирмы Winchester) предназначен для хранения той информации, которая более или менее часто используется в работе: программы операционной системы, компиляторов, сервисных программ, прикладных программ пользователя, текстовых документов, файлов базы данных и пр. Винчестер обеспечивает более быстрый доступ к данным, чем дискета, и значительно превосходит ее в емкости и надежности. *Жесткий диск* - это одна или более жестких пластинок, покрытых материалом, позволяющим делать магнитную запись компьютерных данных. По его поверхности двигаются считывающее-записывающие головки. Между ними и поверхностью диска есть воздушный зазор, предохраняющий диск от повреждений и загрязнений.

*Гибкие магнитные диски.* Дискеты используются для обмена программами и данными между компьютерами, для хранения программ, не используемых в работе, или запасных копий данных на случай их разрушения в винчестере.



Можно использовать их для обычной работы на компьютере, однако это резко замедляет исполнение программ. Floppy disc - это круглое плоское изделие из майлара (полиэтилен-перифталат), покрытое оксидом железа; субстанцией, содержащей крошечные частицы, которые удерживают магнитное поле. Дискета упакована в защитную пластиковую оболочку. Данные считываются и записываются с помощью считывающее-записывающие головки, которая сменяет магнитное ориентирование частиц. Направление в одну сторону представляет собой двоичную «1», а в противоположную ей - двоичный «0». На настоящий момент практически исчезли из использования.

*CD-ROM* (CD-R) - это аббревиатура, обозначающая Compact Disc Read Only Memory. С них можно только считывать информацию, так как она наносится на поверхность диска в виде желобков. Хотя кроме этого незначительного минуса, у них есть много плюсов. Во-первых, они обладают большой вместительностью - свыше 600 Мбайт. Во-вторых, использование лазерной оптики значительно ускоряет процесс считывания данных по сравнению с магнитными средствами. Существую также CD-диски для чтения и записи информации (пишущие CD или CD-RW), различные DVD диски.

*Примечание:* другие представители внешней памяти – см. презентации.

*Организация обмена данными*

Возможности использования ВМ в значительной степени зависят от состава и технических характеристик периферийных устройств. Функции, выполняемые периферийными устройствами, весьма разнообразны. *Внешнее запоминающее устройство* предназначены для энергонезависимого хранения больших объемов им формации. *Устройства ввода-вывода* используются для ввода в ВМ и вывода из нее различной информации: числовой, текстовой, графической, аудио. *Периферийные устройства* различаются принципом действия, форматами и скоростью передачи информации через их внешние выводы, набором управляющих сигналов. Каждое периферийное устройство работает в своем темпе, не синхронизированном с работой процессора. Запросы со стороны периферийных устройств на установление связи и обмен данными могут поступать в произ-

вольные моменты времени. Для организации обмена требуются специальные электронные средства согласования форматов и синхронизации процессов.

В соответствии с принципами, оговоренными ранее при организации ВМ используются *принципы модульности* и *магистральности*. Модульность предполагает построение системы из ми дулей со стандартным интерфейсом. Принцип магистральности означает использование для информационного обмена между устройствами ВМ магистралей (шин), к которым подключаются модули. Принципы организации информационного обмена по шинам включают в себя:

1. Принцип подчиненности предполагает, что одно из обменивающихся информацией по шине устройств является ведущим, управляющим процессом обмена, а другое (или несколько других) – ведомым

2. Принцип квитирования базируется на использовании специального сигнала квитанции, формируемого ведомым устройством и используемого ведущим как разрешение завершения цикла обмена. Это позволяет организовать обмен между модулями с различным быстродействием.

3. Принцип унификации характеристик модулей состоит в обеспечении информационной, электрической и конструктивной совместимости интерфейсов модулей вычислительной системы.

*Информационная совместимость* предполагает единый состав управляющих сигналов в линиях шины, форматов адресов и данных, а также протоколов обмена.

*Электрическая совместимость* это согласованность параметров электрических сигналов, используемых для информационного обмена по шине, нагрузочной способности источников сигнала и входных токов приемников, паразитных параметров линии связи.

*Конструктивная совместимость* это унификация модулей, шины и корпуса ВМ по конструктивным параметрам: размерам, типам соединителей, мест их установки и т. п.



Отметим, что для организации обмена используются специальные командные сигналы (например стробы чтения и записи) при этом обмен по магистралям МП бывает синхронный и асинхронный.

Обмен данными между периферийными устройствами и вычислительным ядром системы.

Проблемы организации обмена данными между периферийными устройствами и вычислительным ядром (процессором и ОП) связаны с асинхронным характером процессов т.е. случайным их появлением. В ВМ используют три способа организации обмена, каждый из которых по-разному решает отмеченные проблемы:

- программно-управляемая передача, инициируемая процессором (бывает асинхронная и синхронная);
- передача информации с прерыванием программы, активизируемая запросу прерывания от периферийного устройства;
- передача информации в режиме прямого доступа к памяти.

При программно-управляемой передаче и передачах данных с прерыванием программы и обмен осуществляется под управлением процессора. Операции ввода-вывода при таких обменах инициируются либо текущей командой программы, либо запросом прерывания от периферийного устройства. В режиме прямого доступа к памяти передача информации выполняется без участия процессора под управлением специализированного управляющего блока – контроллера прямого доступа к памяти.

Програмно-управляемая передача. Программно-управляемый обмен бывает синхронным и асинхронным. Инициатором обмена всегда выступает процессор, реализующий требуемые операции ввода-вывода с помощью соответствующих команд. *Синхронная передача* применяется при взаимодействии с быстродействующими периферийными устройствами, для обмена с которыми не требуется дополнительной синхронизации (такие устройства ввода-вывода всегда готовы к обмену информацией). Этот способ передачи реализуется при минимальных затратах аппаратных и программных средств. *Асинхронный об-*

*мен* является более универсальным и более сложным способом программно-управляемого обмена. Он используется при работе с периферийными устройствами, быстродействие которых ниже быстродействия процессора. В некоторые моменты времени такие периферийные устройства могут оказаться не готовыми к обмену. Поэтому для выполнения программно-управляемого обмена в общем случае необходимо использовать специальные средства, синхронизирующие процесс приема-передачи. Эти средства содержатся в адаптере (контроллере) периферийных устройств. Адаптер (контроллер) подключается к шине ВМ и служит посредником между периферийными устройствами и вычислительным ядром.

Любой адаптер содержит регистр данных (порт данных). *В режиме ввода* данные из периферийных устройств поступают в порт и хранятся в нем до момента пересылки по шине в вычислительное ядро. *В режиме вывода* данные записываются в порт процессором и хранятся там до передачи с помощью адаптера в устройство вывода. Большинство адаптеров периферийных устройств, кроме регистра данных, через который собственно и осуществляется обмен информацией с процессором, содержат в своем составе специальные регистры управления и состояния. Названные регистры, подключаемые к шине и доступные процессору для чтения или записи, обеспечивают взаимодействие процессора и адаптера периферийного устройства, необходимое для управления процессом обмена. *Регистр состояния* отражает факт подключения периферийного устройства и его работоспособность. Один из разрядов этого регистра может использоваться в качестве флага готовности READY, который устанавливается периферийным устройством при готовности к приему-передаче данных через шину. Процессор может проверить состояние флага готовности и при наличии сигнала готовности выполнить требуемый обмен. Проверка готовности к обмену может осуществляться программно или аппаратно. В большинстве случаев современные МП для проверки готовности используют аппаратные средства. Они имеют специальный вход готовности READY, который по шине управления соединен с выходом одноименного триггера готовности периферийного



устройства. Перед выполнением любой операции обмена процессор опрашивает вход готовности и по результатам проверки либо осуществляет внешний обмен (READY = 1), либо переходит в состояниее ожидания, пока периферийное устройство не будет готово к обмену (READY = 0).

Программно-управляемая передача является идеальным (самым быстрым) способом обмена данными между периферийным устройством и процессором. *Недостатком* такого способа обмена являются вынужденные непроизводительные затраты времени процессора на ожидание готовности периферийного устройства к обмену. Такой способ передачи допустим, если быстродействие системы не является критическим фактором и МП в момент выполнения передачи не выполняет каких либо других действий. К недостаткам такого обмена информации так же можно отнести ситуацию, когда по каким-либо причинам (например, из-за возникшей неисправности) сигнал готовности вообще не может быть сформирован периферийным устройством. В такой ситуации процессор не сможет выйти из режима ожидания, и его работа будет заблокирована.

Для исключения подобных ситуаций в современных вычислительных системах используют специальный программный прием, называемый *тайм-аутом*. (Тайм-аут – это средство восстановления работоспособности ВМ при не поступлении сигнала готовности в течение заданного интервала времени.) Предотвратить потери времени в циклах ожидания и существенно улучшить программное взаимодействие процессора с периферией, особенно при одновременной работе со многими периферийными устройствами, позволяет использование прерывания.

<u>Передача информации с(по) прерыванием(ями)</u> Тип обмена данными, при котором для выполнения операций ввода-вывода производят прерывание программы, называется *передача данных с прерыванием программы*. Важным отличием обмена данными с прерыванием программы от синхронного и асинхронного обменов является то, что в нем инициатором обмена является не процессор, а внешнее устройство, запросившее обмен. Возможность прерывания программ – важное архитектурное свойство ВМ, позволяющее повышать про-

изводительность процессора при наличии нескольких протекающих параллельно во времени процессов, требующих в произвольные моменты времени управления и обслуживания со стороны процессора.

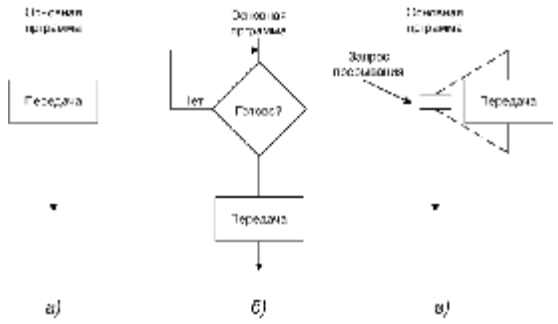
Реализация обменов с прерыванием программы по сравнению с программно-управляемым вводом-выводом требует более сложной аппаратной и программной поддержки. Периферийное устройство, поддерживающее такой обмен, должно содержать специальную схему, формирующую *сигнал запроса прерывания* INT, который по соответствующей линии шины управления поступает на одноименный вход процессора. При использовании обменов с прерыванием программы процессор не тратит время на слежение за состоянием флагов готовности своих устройств ввода-вывода, а работает по своей основной программе. При готовности к обмену активизирующееся устройство ввода-вывода формирует запрос прерывания и посылает его процессору. Последний, обнаружив сигнал запроса, завершает выполнение операций, которые нельзя прервать (например, чтобы избежать потери данных, заканчивает исполнение текущей команды) и выполняет определенную последовательность действий.

Прежде чем прервать основную программу и перейти к обработчику прерывания, процессор запоминает в стеке свое текущее состояние, по крайней мере, адрес следующей команды прерываемой программы (адрес возврата), идентифицирует источник прерывания и только после этого переходит к подпрограмме обработки прерывания. По завершению подпрограммы обработки прерывания процессор восстанавливает состояние на момент прерывания и возвращает управление прерванной программы. Более подробно организация работы ВМ при обработке запроса прерывания рассматривается в курсе МПСУ.

Заметим несколько важных моментов.

а) Алгоритм организации ввода вывода при программно-управляемом обмене и обмене с прерыванием программы представлен на рис. 2.5.






а) Синхронный; б) Асинхронный; в) С прерыванием

*Рис. 2.5.* Организация обмена

б) Эффективность обменов с прерыванием программы определятся типом периферийного устройства, участвующих в обмене. При работе с относительно медленными устройствами ввода-вывода потери времени на переключение процессора при прерываниях относительно невелики, поэтому организация ввода-вывода с использованием системы прерываний для таких устройств достаточно эффективна. В случае быстродействующих периферийных устройств, при передачах блоков, на транзитные пересылки слов из порта в регистр процессора, а затем из этого регистра в память тратится сравнительно много времени. При высокой скорости обмена (200 Кбайт/с и выше), характерной, например, для обменов накопителей на жестких магнитных дисках с ОП, интервалы между передачами слов невелики (5 мк. сек. и менее). За это время практически невозможно осуществить контекстное переключение процессора и выполнить саму процедуру обмена, при этом на все время программно-управляемой передачи процессор вынужден отвлекаться от выполнения основной программы. В ряде случаев это оказывается крайне нежелательным. В этих случаях используют прямой доступ к памяти.

<u>Передача информации в режиме прямого доступа к памяти.</u> *Прямым доступом к памяти* (ПДП, UDMA) называют способ обмена данными, обеспечивающий установление связи и передачу данных между оперативной памятью и периферийным устройством автономно от центрального процессора. При нали-

чии кэш-памяти памяти (внутренней или внешней) режим ПДП позволяет осуществлять обмен данными между ОП и периферийным устройством параллельно с выполнением процессором основной программы. Использование режима ПДП разгружает процессор от обслуживания операций ввода-вывода и способствует повышению общей производительности ВМ. Прямой доступ к памяти требует более сложного интерфейса, по сравнению с программно-управляемой передачей. Для управления обменом данными в режиме ПДП применяют специальные управляющие контроллеры, которые в зависимости от вариантов аппаратной реализации подразделяются на специализированные процессоры ввода-вывода и контроллеры ПДП. В отличие от других способов обмена данными в режиме ПДП управляющий контроллер после получения запроса на организацию передачи от периферийного устройства принимает на себя функции управления системной шиной, при этом сам процессор путем перевода своих тристабильных выходных буферов шины в состояние высокого сопротивления отключается от системной шины.

## *2. Персональные компьютеры: особенности архитектуры и применения*

Блоки и устройства, входящие в состав ПК показаны на презентации. В целом, ПК состоит из центрального системного блока, к которому через внешние разъемы подключаются дополнительные блоки: клавиатура дисплей, принтер и т.п. Основой системного блока является материнская плата, состоящая из: микропроцессора, системных чипсетов, генератора тактовых импульсов, модулей ОЗУ, ПЗУ, микросхемы CMOS-памяти; адаптеров клавиатуры и различных дисков, контроллеров прерываний, таймеров и т.п.

Отдельным классом компьютерных систем являются портативные ПК, ноутбуки, планшеты и смартфоны. Здесь же остановимся в основном на настольных персональных компьютерах, как на наиболее распространенном классе вычислительных устройств.



Перечислим некоторые особенности отдельных устройств и элементов, а потом рассмотрим одну из основных частей ПК – материнскую плату.

1. Микропроцессор

Это центральное устройство, предназначенное для управления работой всех блоков машины и для выполнения различных операций над информацией.

Некоторые особенности.

*Разрядность шины данных* микропроцессора определяет количество разрядов, над которыми одновременно могут выполняться операции; *разрядность шины адреса* МП определяет его адресное пространство.

*Адресное пространство* – это максимальное количество ячеек основной памяти, которое может быть непосредственно адресовано микропроцессором.

*Рабочая тактовая частота* МП во многом определяет его внутреннее быстродействие, поскольку каждая команда выполняется за определенное количество тактов. Быстродействие (производительность) ПК зависит также и от тактовой частоты шины системной платы, с которой работает (может работать) МП.

*Кэш-память*, может иметь несколько уровней:

L1 – память 1-го уровня, находящаяся внутри основной микросхемы (ядра). МП и работающая всегда на полной частоте МП (впервые кэш L1 был введен в МП i486 и у МП 1386SLC). Чаще применяется для ускорения работ программ;

L2 – память 2-го уровня, кристалл, размещаемый на плате МП и связанный с ядром внутренней микропроцессорной шиной (впервые введен в МП Pentium II). Память L2 может работать на полной или половинной частоте МП. Эффективность этой кэш-памяти зависит и от пропускной способности микропроцессорной шины. Чаще используется для взаимодействия с внешними устройствами.

L3 – память 3-го уровня используется в многопроцессорных и многоядерных решениях, является общей памятью всех ядер (процессоров) системы. Медленнее чем L2, но быстрее чем оперативная память.

*Состав инструкций* – перечень, вид и тип команд, автоматически исполняемых МП. От типа команд зависит классификационная группа МП (CISC, RISC, VLIW, MISC и т. д.). Перечень и вид команд определяют непосредственно те процедуры, которые могут выполняться над данными в МП, и те категории данных, над которыми применимы эти процедуры. Дополнительные инструкции в небольших количествах вводились во многих МП (286, 486, Pentium Pro и т. д.). Существенное изменение состава инструкций произошло в МП i386 (этот состав далее принят за базовый), и совместимость по инструкциям для 32-разрядных систем обозначается x86.

*Конструктив* подразумевает те физические разъемные соединения, в которые устанавливается МП, и которые определяют пригодность материнской платы для установки МП. Разные разъемы имеют разную конструкцию (Slot – щелевой разъем, Socket – разъем-гнездо), разное количество контактов, на которые подаются различные сигналы и рабочие напряжения.

*Рабочее(ие) напряжение(ия)* также является фактором пригодности материнской платы для установки конкретного МП.

*Конвейерное выполнение команд* – это одновременное выполнение разных тактов последовательных команд в разных частях МП при непосредственной передаче результатов из одной части МП в другую. Конвейерное выполнение команд увеличивает эффективное быстродействие ПК в 2-5 раз (впервые применено 80286). В современных условиях идея конвейера обогатилась таким решениями как блоки предсказания ветвлений и переходов, динамическое выполнение команд и т.п.

*Многозадачной работы* (многопрограммность) и сопутствующая ей защита памяти. Современные микропроцессоры имеют два режима работы: *реальный* (однозадачный, Real Address Mode), в котором возможно выполнение только одной программы и непосредственно адресоваться могут только 1024 + 64 Кбайт основной памяти компьютера, а остальная память (расширенная) доступна лишь при подключении специальных драйверов (поддерживается операционной системой); и *защищенный* (многозадачный, Protected Virtual Address



Mode), обеспечивающий выполнение сразу нескольких программ, непосредственную адресацию и прямой доступ (без дополнительных драйверов) к расширенной основной памяти. Предоставляется непосредственный доступ к памяти емкостью 16 Мбайт для МП 286; 4 Гбайт для процессоров 386, 486, Celeron; 100 Гбайт для МП Pentium Xeon и 64 Гбайт для остальных процессоров Pentium, а при страничной организации памяти – к 16 Тбайт виртуальной памяти для каждой задачи. В этом режиме осуществляется автоматическое распределение памяти между выполняемыми программами и соответствующая ее защита от обращений со стороны чужих программ. Защищенный режим поддерживается операционными системами Windows, OS/2, UNIX.

*Система виртуальных машин* является дальнейшим развитием режима многозадачной работы, при котором каждая задача может выполняться под управлением своей операционной системы, то есть практически в одном МП моделируется как бы несколько компьютеров, работающих параллельно и имеющих разные операционные системы.

*Динамическое исполнение* (dynamic execution) это в первую очередь, означает наличие многоступенчатой суперконвейерной структуры (superpipelining), предсказания ветвлений программы при условных передачах управления (multiple branch prediction) и исполнение команд по предполагаемому пути ветвления (speculative execution). Если процессор может заранее предсказывать направление перехода (ветвления), то производительность его работы значительно повысится за счет оптимизации загрузки вычислительных конвейеров. Если путь ветвления предсказан неверно, процессор должен сбросить полученные результаты, очистить конвейеры и загрузить нужные команды заново, что требует достаточно большого числа тактов.

*MMX технология*– MultiMedia eXtention (Pentium II МП Pentium MMX) означает содержание дополнительных команд, ориентированных на обработку аудио- и видеоинформации, увеличение кэш-память, дополнительное введение многоразрядных регистров.

*Многопоточность* (Hyper-threading) это возможность на одном физическом ядре реализовать несколько потоков выполнения инструкций, т.е. по сути виртуально иметь несколько ядер.

*Многоядерность* это наличие в одной микросхеме процессора по сути нескольких процессоров – процессорных ядер, объединенных с помощью различных средств.

*Технология изготовления* (производства) – это технологическая норма, по которым выполнена микросхема. Технологическая норма связана с размерами транзистора, толщиной и длинной затвора, а так же с расстоянием между элементами. В настоящее время применяются нормы 22 нм.

2. Системная шина и системный контроллер

Это основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой. Включает шину адреса, данных, управления, питания (см. выше).

*Особенность:* все блоки, а точнее их порты ввода-вывода, через соответствующие унифицированные разъемы (стыки) подключаются к шине единообразно: непосредственно или через контроллеры (адаптеры). Управление системной шиной осуществляется микропроцессором либо непосредственно, либо, что чаще, через дополнительную микросхему контроллера шины, формирующую основные сигналы управления. Обмен информацией между внешними устройствами и системной шиной выполняется с использованием специальных кодов. Современные шины поддерживают как последовательный, так и параллельный режим.

В настоящее связь между устройствами ПК обеспечивается с помощью многоуровневой системы шин. Так например процессор и память могут обменивается информацией через контроллер (так называемый «северный мост») с помощью шины FSB по которой процессор и подключается к нему. К данному контроллеру так же могут подсоединяться достаточно быстродействующие устройства – например, по шине PCI Express такие устройства как видеоадаптер. «Северный мост» соединяется с другим контроллером – «южным мостом»



(оба моста образуют так называемый системный чипсет – набор логики материнской платы – см. ниже). К «южному мосту» подключаются достаточно медленные устройства по своим – периферийным шинам – см. ниже.

В некоторых случаях (современные процессоры) когда контроллер памяти входит в состав самого процессора «южный мост» отсутствует, а сама FSB в таких системах отсутствует принципиально.

3. Основная память

Помимо сверхоперативной памяти (см. выше) и внешней памяти (см. ниже) в персональных компьютерах существует средний уровень та называемая основная память.

Данная память предназначена для хранения и оперативного обмена информацией со всеми элементами системы. Память содержит два вида запоминающих устройств: постоянное запоминающее устройство **(**ПЗУ**)** и оперативное запоминающее устройство (ОЗУ) – см. выше.

Особенность: кроме основной памяти обычно реализованной в виде ОЗУ на системной плате ПК имеется и энергонезависимая память CMOS RAM (Complementary Metal-Oxide Semiconductor RAM), постоянно питающаяся от своего аккумулятора; в ней хранится информация об аппаратной конфигурации ПК (обо всей аппаратуре, имеющейся в компьютере), которая проверяется при каждом включении системы. Помимо этого на материнской плате имеется несколько ПЗУ хранящих служебную информацию – см. ниже.

4. Внешняя память

Относится к внешним устройствам ПК и используется для долговременного хранения любой информации, которая может когда-либо потребоваться для решения задач. Современное состояние и яркие представители данного типа устройств представлены на презентации.

5. Источник питания

Персональный компьютер питается от вторичного источника электропитания постоянного тока, преобразующего сетевое напряжение до требуемых значений. Помимо вторичного блока питания ПК снабжаются блоками беспе-

ребойного питания (аккумуляторные батареи) защищающие его от исчезновения или скачков напряжения, а так же осуществляющего стабилизацию входного сетевого напряжения.

Блок питания располагается в самом корпусе ПК и обычно (согласно спецификации АТХ 2.х), должен обеспечивать выходные напряжения ±5, ±12, +3,3 Вольт, а также +5 Вольт дежурного режима.

Подключение устройств, в том числе и материнской платы к блоку питания осуществляется с помощью специальных разъемов (с разным количеством контактов), провода в которых имеют уникальную цветовую маркировку, указывающую на тип передаваемого сигнала (напряжения).

6. Таймер

Таймер или внутри машинные электронные часы реального времени, обеспечивают при необходимости автоматический съем текущего момента времени (год, месяц, часы, минуты, секунды и доли секунд). Таймер подключается к автономному источнику питания – аккумулятору (см. выше), и при отключении машины от электросети продолжает работать.

7. Внешние устройства (периферийные)

Внешние устройства ПК или так называемая периферия обеспечивают взаимодействие машины с окружающей средой: пользователями, объектами управления и другими компьютерами. К внешним устройствам относятся: внешние запоминающие устройства (ВЗУ) или внешняя память ПК; диалоговые средства пользователя; устройства ввода информации; устройства вывода информации; средства связи и телекоммуникаций.

Особенность:

- *Диалоговые средства пользователя* включают в свой состав: устройства речевого ввода-вывода или та называемые средства мультимедиа (акустические системы, средства распознавания речи и т.п.); в некоторых случаях сюда относят видеомонитор (видеотерминал, дисплей и т.п.).

- *Устройства ввода* включают: клавиатуру; графические планшеты (дигитайзеры) – устройства для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера); сканеры; устройства целеуказания (мышь, джойстик и т.п.); сенсорные экраны.



- *Устройства вывода* информации: принтеры; графопостроители, плоттеры; в некоторых случаях сюда относят видеомонитор (видеотерминал, дисплей и т.п.).

- *Устройства связи и телекоммуникации* используются для связи с приборами и другими средствами (согласовали интерфейсов, адаптеры, цифроаналоговые и аналого-цифровые преобразователи и т. п.) и для подключения ПК к каналам связи, к другим компьютерам и вычислительным сетям (сетевые интерфейсные платы и карты – сетевые адаптеры, мультиплексоры передачи данных, модемы: модуляторы-демодуляторы).

Многие из названных выше устройств относятся к условно выделенной группе средств *мультимедиа.*

8. Дополнительные интегральные микросхемы

- *Математический сопроцессор* (устар.) широко использовался для ускоренного выполнения операций над двоичными числами с фиксированной и плавающей запятой, над двоично-кодированными десятичными числами, для вычисления некоторых трансцендентных, в том числе тригонометрических функций. Математический сопроцессор имел свою систему команд и работал параллельно (совмещенно во времени) с основным МП, но под управлением последнего. Ускорение выполнения операций происходило в десятки раз. Современные модели МП, начиная с МП 80486 DX, включают сопроцессор в свою структуру.

- *Контроллер прямого доступа к памяти* (DMA – Direct Memory Access) обеспечивает обмен данными между внешними устройствами и оперативной памятью без участия микропроцессора, что существенно повышает эффективное быстродействие ПК. Входит в состав системного набора микросхем

- *Сопроцессор ввода-вывода* (устар.) за счет параллельной работы с МП существенно ускоряет выполнение процедур ввода-вывода при обслуживании нескольких внешних устройств (дисплея, принтера, НЖМД, НГМД и т. д.);

освобождает МП от обработки процедур ввода-вывода, в том числе реализует и режим прямого доступа к памяти. В настоящее время функции сопроцессора вода-вывода выполняют специализированные контроллеры.

- *Контроллер прерываний* обслуживает процедуры прерываний. *Прерывание* временный приостанов выполнения одной программы с целью оперативного выполнения другой, в данный момент более важной (приоритетной) программы. Контроллер принимает запрос на прерывание от внешних устройств, определяет уровень приоритета этого запроса и выдает сигнал прерывания в МП. Микропроцессор, получив этот сигнал, приостанавливает выполнение текущей программы и переходит к выполнению специальной программы обслуживания того прерывания, которое запросило внешнее устройство. После завершения программы обслуживания восстанавливается выполнение прерванной программы. Контроллер прерываний является программируемым.

## Системная плата

На системной или материнской плате непосредственно расположены:

- разъем для подключения микропроцессора;
- набор системных микросхем (чипсет, chipset), обеспечивающих работу микропроцессора и других узлов машины;
- микросхема постоянного запоминающего устройства, содержащего программы базовой системы ввода-вывода (Basic Input-Output System – BIOS, либо ее альтернативы UEFI);
- микросхема энергонезависимой памяти (питается от автономного, расположенного на плате, аккумулятора), по технологии изготовления называемая CMOS;
- микросхемы дополнительной памяти;
- разъемы для подключения модулей оперативной памяти;
- наборы микросхем и разъемы для системных, локальных и периферийных интерфейсов;
- микросхемы мультимедийных устройств и т. д.



Тип системной платы определяет, прежде всего, тип (серия) *базового микропроцессора*, поддержка которого гарантируется и на который они рассчитаны.

Поскольку микропроцессоры фирмы Intel и процессоры некоторых других фирм, в частности фирмы AMD, используют электрически разные интерфейсы и геометрически разные разъемы, то материнские платы делят по виду разъема процессора. Socket 1(2,..,479)7, Slot 1(2), Slot A(B), Socket A(B,H,…,G3) и другие - процессорные разъемы, установленные на материнской плате. Спецификация разъема определяет как сам конструктив разъема, так и назначение контактов, электрические параметры, определяет порядок взаимодействия с шинами данных, особенности взаимодействия с оперативной памятью и т. п.

Важным параметром системной платы является *тактовая частота* (FSB) на которой она работает: современные платы имеют рабочие частоты 100, 133, 150, 200, 266, 400, 800 МГц и более. Этот параметр особенно влияет на производительность ПК, аналогично с архисекретными решениями, применяемыми в процессоре и других элементах.

Существуют различные базовые типоразмеры (форм-факторы) материнских плат. Форм-фактор определяет не только внешние размеры системных плат, но и ряд специфических параметров, характеризующих функциональные и эксплуатационные свойства. Например: AT, ATX,…, WTX.

От типа, используемого на системной плате набора *системных микросхем* (чипсета) зависят многие важные характеристики ПК. Чипсет определяют во многом тактовую частоту шин, обеспечивают надлежащую работу микропроцессора, системной шины, интерфейсов взаимодействия с оперативной памятью и другими компонентами ПК. В частности, они содержат в себе контроллеры прерываний, прямого доступа к памяти, микросхемы управления памятью и шиной – все те компоненты, которые в оригинальной IBM PC были собраны на отдельных микросхемах. Обычно в одну из микросхем набора входят также часы реального времени с CMOS-памятью и иногда – контроллер клавиатуры, однако эти блоки могут присутствовать и в виде отдельных микросхем. В по-

следних разработках в состав микросхем наборов для интегрированных плат стали включаться и контроллеры внешних устройств.

Тип набора в основном определяет функциональные возможности платы:

- типы поддерживаемых процессоров;
- структуру и объем кэш-памяти;
- возможные сочетания типов и объемов модулей памяти;
- поддержку режимов энергосбережения;
- возможность программной настройки параметров и т. п.

Современные системные наборы состоят из двух базовых микросхем с условными именами *северный мост* (North bridge) и *южный мост* (South bridge). Северный мост обеспечивает управление четырьмя компонентами: шиной процессора (FSB) и внутренней (системной) шиной, поэтому его иногда называют 4-портовым контроллером. Южный мост имеет в своем составе контроллеры (адаптеры) дисководов, клавиатуры, мыши, управляет интерфейсными шинами IDE/ATA/SATA, SCSI, USB, IEEE, PCI, LPC; его также называют функциональным контроллером.

Типы чипсетов и их модификации различаются типами поддерживаемых микропроцессоров и памяти DRAM (SDRAM, DRDRAM, DDR SDRAM), частотой процессорной шины между мостами (133, 266, 400, 533, 800 МГц), вариантам схем мостов, версиями поддерживаемых протоколов UDMA, AGP, Ethernet, USB, наличием встроенного видеоадаптера и др.

Системная плата имеет разъемы *(слоты) для* установки определенных модулей оперативной памяти. Для модулей ОЗУ используются 30-, 72- и 168-, 184-, 240-контактные разъемы и они сгруппированы в 2-4 слотах. Слоты с 30 и 72 контактами рассчитаны на модули памяти типа SIMM, которые в новых машинах уже не используются; 168-контактные слоты предназначены для модулей DIMM. В настоящее время в основном используется память DDR с 184-контактным разъемом и более. С точки зрения поддерживаемой оперативной памяти, важными характеристиками являются также число слотов памяти, рабочая тактовая частота памяти (см. ниже) и общая емкость поддерживаемой памяти.



На системной плате (в новых моделях обычно в составе микросхем чипсета) располагается микросхема постоянного запоминающего устройства (ПЗУ), содержащая программы BIOS, необходимые для управления многими компонентами компьютера. BIOS доступна постоянно, независимо от работоспособности внешних компонентов, таких, например, как системные загрузочные диски. В современных материнских платах используются, как правило, микросхемы Flash BIOS, программы в которых могут перезаписываться при помощи специальных средств, что облегчает модернизацию этих программ при появлении новых устройств, которым нужно обеспечить поддержку (например, новых типов микросхем оперативной памяти). В настоящее время осуществляется постепенный переход на новый тип BIOS – UEFI.

Важным компонентом, размещенным на системной плате, является микросхема CMOS-памяти. Она питается от своего локального аккумулятора (батарейки) и поэтому является энергонезависимой (сохраняет информацию при отключении компьютера от сети). Название CMOS произошло от названия технологии исполнения ее элементов. Память хранит информацию о параметрах многих устройств, входящих в ПК. Информация в ней может изменяться по мере необходимости, то есть память отслеживает текущую конфигурацию компьютера, на что не способна микросхема BIOS. Поэтому при загрузке компьютера BIOS берет необходимую для своей работы информацию об изменяемых компонентов ПК именно из этой памяти. Так, из CMOS-памяти считывается информация об установленном МП, о типах и емкости оперативной и всех видах дисковой памяти, о работоспособности устройств компьютера и т. д. Четкое отслеживание времени (в том числе и календаря), даже в отключенном от энергосети состоянии, также связано с тем, что информация о времени хранится в CMOS-памяти.

На системных платах располагаются также *перемычки* (джамперы) и *DIP-переключатели*, используемые для конфигурирования различных компонентов.

В современных компонентах пытаются сократить количество механических переключений, перекладывая их на автоматические программно-управляемые электронные схемы. Компоненты, которые после установки конфигурируются автоматически, относят к категории PnP-устройств (PnP – Plug and Play, Включай и работай).

*Системный и периферийный интерфейс*

*Интерфейс* (interface) – совокупность средств сопряжения и связи, обеспечивающая эффективное взаимодействие систем или их частей. Как было отмечено выше *системная шина* это многоуровневая система, как и все компьютерные системы и технологии с точки зрения организации системного и периферийного интерфейса данная архитектура претерпела множество изменений.

*Шина ISA* (Industry Standard Architecture – промышленная стандартная архитектура) представлена в двух версиях: для IBM PC/XT (1981 год) и для PC AT (1984 год). Она использовалась в первом компьютере IBM PC, и тогда имела неофициальное название IBM PC/XT, и позволяла добавлять в систему различные устройства. Часто возникает путаница относительно ее характеристик, поскольку в новых компьютерах эта шина не применяется, а фирма IBM до 1987 года не публиковала ее полного описания и временных диаграмм сигналов. В настоящее время устарела.

*Шина PC/XT* – содержала 8-разрядную шину данных и 20-разрядную шину адреса; была рассчитана на тактовую частоту 4,77 МГц; имела 4 линии для аппаратных прерываний и 4 канала для прямого доступа в память (каналы DMA – Direct Memory Access). Шина адреса ограничивала адресное пространство микропроцессора величиной 1 Мбайт. Использовалась с МП 8086, 8088.

*Шина PC/AT* – содержала 16-разрядную шину данных и 24-разрядную шину адреса; рабочая тактовая частота до 8 МГц; могла применяться и в МП с тактовой частотой 16 МГц, так как контроллер шины способен делить частоту пополам; имела 7 линий для аппаратных прерываний и 4 канала DMA. Использовалась с МП 80286. Совместима и с МП с тактовой частотой больше 66 МГц



(коэффициент деления увеличен); увеличено количество линий аппаратных прерываний с 4 до 15 и каналов прямого доступа к памяти (DMA) с 4 до 7. Благодаря 24-разрядной шине адреса адресное пространство увеличилось с 1 Мбайт до 16 Мбайт. Теоретическая пропускная способность шины данных равна 16 Мбайт/с, но реально она ниже, около 5,5 Мбайт/с, в виду ряда особенностей ее использования.

*Шина EISA* (Extended ISA) – имела 32-разрядную шину данных и 32-разрядную шина адреса; создана в 1989 году как функциональное и конструктивное расширение ISA. Адресное пространство шины было 4 Гбайт, работала на частоте 8-10 МГц. Теоретическая пропускная способность шины – 33 Мбайт/с, причем скорость обмена по каналу МП – кэш – ОП определяется параметрами микросхем памяти; увеличено число разъемов расширений – теоретически может подключаться до 15 устройств (практически до 10). Улучшена система прерываний, поддерживается *Bus Mastering* – режим единоличного управления шиной со стороны любого из устройств на шине, имеет *систему арбитража* для управления доступом устройств к шине. Обеспечивалось автоматическое конфигурирование системы и управление DMA. Шина поддерживает многопроцессорную архитектуру вычислительных систем. Шина EISA весьма дорогая и находила применение в скоростных системах, сетевых серверах и рабочих станциях.

*Шина MCA* (Micro Channel Architecture) – 32-разрядная шина, созданная фирмой IBM в 1987 году для машин PS/2, пропускная способность 76 Мбайт/с, рабочая частота 10-20 МГц. По своим прочим характеристикам близка к шине EISA, но не совместима ни с ISA, ни с EISA. Поскольку компьютеры PS/2 не получили широкого распространения, в первую очередь ввиду отсутствия наработанного обилия прикладных программ, шина MCA также использовалась не очень широко.

*Шина VLB* (VL-bus, VESA Local Bus) представлена в 1992 году ассоциацией стандартов видео электроники (VESA – торговая марка Video Electronics Standards Association) и поэтому часто ее называют шиной VESA. Шина VLB,

по существу, является расширением внутренней шины МП для связи с видеоадаптером и реже с жестким диском, платами мультимедиа, сетевым адаптером. Разрядность шины для данных – 32 бита, для адреса – 30 бит;, реальная скорость передачи данных по VLB – 80 Мбайт/с, теоретически достижимая – 132 Мбайт/с (в версии 2 –400 Мбайт/с). В настоящий момент не используется.

*Шина PCI* (Peripheral Component Interconnect, соединение внешних компонентов) – один из самых распространенных и универсальных интерфейсов для подключения различных устройств. Разработана в 1993 году фирмой Intel. Шина PCI является намного более универсальной, чем VLB; допускает подключение до 10 устройств; имеет свой адаптер, позволяющий ей настраиваться на работу с любым МП от 80486 до Pentium и страше. Тактовая частота базовой версии шины PCI – 33 МГц, разрядность – 32 разряда для данных и 32 разряда для адреса с возможностью расширения до 64 бит, теоретическая пропускная способность 132 Мбайт/с, а в 64-битовом варианте – 264 Мбайт/с. Модификация 2.1 локальной шины PCI работает на тактовой частоте до 66 МГц и при разрядности 64 имеет пропускную способность до 528 Мбайт/с. Осуществлена поддержка режимов Plug and Play, Bus Mastering и автоконфигурирования адаптеров. В настоящее время реализованы и другие версии и скорости передачи данных.

*Шина AGP (*Accelerated Graphics Port – ускоренный графический порт) – интерфейс для подключения видеоадаптера к отдельной магистрали AGP, имеющей выход непосредственно на системную память. Разработана шина на основе стандарта PCI v2.1. Шина AGP может работать с частотой системной шины 133 МГц и обеспечивала высокую скорость передачи графических данных. Пиковая пропускная способность в режиме четырехкратного умножения AGP4x (передаются 4 блока данных за один такт) имеет величину 1066 Мбайт/с, а в режиме восьмикратного умножения AGP8x – 2112 Мбайт/с. По сравнению с шиной PCI, в шине AGP устранена мультиплексированность линий адреса и данных (в PCI для удешевления конструкции адрес и данные передаются по одним и тем же линиям) и усилена конвейеризация операций чтения-записи, что позволяет устранить влияние задержек в модулях памяти на скорость выполнения этих операций.



*Шина PCI-E* пришла на смену шине AGP, фактически шиной не является, т.к. представляет собой соединение типа «точка-точка». Используется программная модель шины PCI. В отличие от параллельной шины PCI является последовательной, двунаправленной. Пропускная способность 2,5 Гбит/с, что с учетом дуплексности и избыточность 8b/10b (8 бит в десяти), для соединения x1 составит:2,5 x· 2 x· 0,8 = 4 Гбит/с (где 2,5 - битрейт, Гбит/с; 2 - учет дуплексности; 0,8 - учет избыточности 8b/10b для 1.0 и 2.0; 0,985 — 128b/130b для 3.0).

В отличие от рассмотренных выше вариантов подразумевает использование в качестве *локальной шины*, а не шины *расширения*.

Помимо рассмотренных выше шин расширения и локальных шин в компьютерных системах присутствует достаточно большое число периферийных шин, использующихся для подключения различных устройств. Рассмотрим их более подробно. К таким шинам относят шины *IDE* (Integrated Drive Electronics), *ATA* (AT Attachment – подключаемый к AT) и его вариантов, *EIDE* (Enhanced IDE), *SCSI* (Small Computer System Interface) используются чаще всего в качестве интерфейса только для внешних запоминающих устройств.

*Интерфейс ATA*, широко известный и под именем Integrated Drive Electronics (IDE), предложен в 1988 году пользователям ПК IBM PC AT. Первоначально разрабатывался для подключения к шине ISA. Он ограничивает емкость одного накопителя 504 Мбайт (по причине организации адресного пространства в традиционной BIOS головка-цилиндр-сектор: 16 головок x 1024 цилиндра x 63 сектора x 512 байт в секторе = 504 Кбайт = 528 482 304 байт) и обеспечивает теоретическую скорость передачи данных 5-10 Мбайт/с. Модификации:

*Fast ATA-2* или Enhanced IDE (EIDE – расширенный IDE), использующий как традиционную (но расширенную) адресацию по номерам головки, цилиндра и сектора, так и адресацию логических блоков (Logic Block Address – LBA), поддерживает емкость диска до 2500 Мбайт и ско-

рость обмена до 16,7 Мбайт/с. К адаптеру EIDE, поддерживающему стандарт AT API, может подключаться до четырех накопителей, в том числе и CD-ROM, и НКМЛ.

*ATAPI* (ATA Package Interface) – стандарт, созданный с тем, чтобы напрямую подключать к интерфейсу ATA не только жесткие диски, но и дисководы CDROM, стримеры, сканеры и т. д. Версии интерфейса ATA-3 и Ultra ATA обслуживают диски большей емкости, имеют скорость обмена до 33 Мбайт/с, поддерживают технологию SMART (Self-Monitoring Analysis and Report Technology технологию самостоятельного следящего анализа и отчета), позволяющую устройствам сообщать о своих неисправностях, и ряд других сервисов. Версии интерфейса ATA/ATAPI-5, ATA/ATAPI-6, ATA/ATAPI-7 по протоколам UDMA/66, UDMA/100 и UDMA/133 обеспечивают пиковую пропускную способность 66, 100 и 133 Мбайт/с, соответственно.

*UDMA* (Ultra Direct Memory Access) – режим прямого доступа к памяти. Обычный метод обмена с жестким диском IDE – это программный ввод-вывод, PIO (Programmed Input/Output), при котором процессор, используя команды ввода-вывода, считывает или записывает данные в буфер жесткого диска, что отнимает какую-то часть процессорного времени. Ввод-вывод путем прямого доступа к памяти идет под управлением самого жесткого диска или его контроллера в паузах между обращениями процессора к памяти, что экономит процессорное время, но несколько снижает максимальную скорость обмена. На материнских платах реализованы два канала IDE, к каждому из которых возможно подключение до двух устройств.

*SATA* (Serial ATA) - последовательный интерфейс обмена данными с накопителями информации. SATA является развитием параллельного интерфейса ATA (IDE), который после появления SATA был переименован в PATA (Parallel ATA). SATA использует 7-контактный разъём вместо 40-контактного разъёма у PATA. Каждому устройству полагается отдельный кабель. Спецификация SATA Revision 1.0 (SATA/150 до 1,5 Гбит/с) была представлена 7 января



2003 года; Пропускная способность SATA/150 незначительно выше пропускной способности шины Ultra ATA (UDMA/133). Спецификация SATA Revision 3.0 (SATA III или SATA 3.0) представлена в июле 2008 и предусматривает пропускную способность до 6 Гбит/с (750 Мбайт/с для данных с учетом 8b/10b кодирования).

*SCSI* (Small Computer System Interface) является сложным и мощным интерфейсом. Спецификация была стандартизирована в 1986. Это универсальные периферийные интерфейсы для любых классов внешних устройств. Фактически SCSI является упрощенным вариантом системной шины компьютера, поддерживающим до восьми устройств. Такая организация требует от устройств наличия определенных контроллеров – например, в жестких дисках SCSI все функции кодирования-декодирования, поиска сектора, коррекции ошибок и т. п. возлагаются на встроенную электронику, а внешний SCSI-контроллер выполняет функции обмена данными между устройством и компьютером – часто в автономном режиме, без участия центрального процессора. Существует много различных спецификаций данного интерфейса, отличающихся пиковой пропускной способностью, максимальным числом подключаемых устройств, предельной длиной кабеля. Ultra-320 SCSI, также известен под названием Fast Ultra-320, это развитие интерфейса Ultra-3 с удвоенной скоростью передачи данных (до 320 Мбайт/сек). Ultra-640 SCSI, предложен в начале 2003 года, пропускная способность (640 Мбайт/сек).

*RS-232* – интерфейс обмена данными по последовательному коммуникационному порту (COM-порту). Управление работой COM-портов (число которых ограничено четырьмя) осуществляется специальной микросхемой UART16550A расположенной на материнской плате. Физически разъем COM-порта может быть 25- или 9-контактным. С помощью данного интерфейса осуществлялась работа и подключение таких устройств, как внешний модем, мышь и т.д. В настоящее время устарел.

*RS-485* – стандарт физического уровня для асинхронного интерфейса. Физическая среда – витая пара. Используется для построения различных сетей. Длина линии – 1,2 км. Скорость 100 кбит/с — 10 Мбит/с.

*RS-422,* ANSI/TIA/EIA-422-B - американский стандарт, его международный эквивалент. В отличие от RS-485, которая обеспечивает многоточечную структуру, EIA-422/V.11 не позволяет иметь несколько отправителей, но позволяет иметь несколько получателей. Опубликован в мае 1994 года, на сегодняшний день находится в ревизии «B». Подтверждён Telecommunications Industry Association в 2005 году. Достоинством стандарта является скорость передачи данных до 10 мегабод для 12-метрового кабеля.

*IEEE 1284 (Institute of Electrical and Electronic Engineers 1284* – стандарт Института инженеров по электротехнике и электронике 1284) более известен как LTP – стандарт, описывающий спецификации параллельных скоростных интерфейсов SPP (Standard Parallel Port – стандартный параллельный порт), EPP (Enhanced Parallel Port – улучшенный параллельный порт), ECP (Extended Capabilities Port – порт с расширенными возможностями), как правило, используемых для подключения через параллельные порты компьютера (LPT-порты) таких устройств, как принтеры, внешние запоминающие устройства, сканеры, цифровые камеры. Со стороны LPT-порта установлен стандартный разъем DB-25 (25 контактов), а со стороны устройства используется разъем типа Centronics. В настоящее время стандарт не развивается, т.к. . последняя спецификация совпала с появлением USB.

*USB* (*Universal Serial Bus)* –универсальная последовательная шина. Она появилась в 1995 году и призвана заменить такие устаревшие интерфейсы, как RS-232 (COM-порт) и параллельный интерфейс IEEE 1284 (LPT-порт), то есть прийти на смену последовательным и параллельным клавиатурным и мышиным портам – все устройства подключаются к одному разъему, допускающему установку многочисленных устройств с легкостью технологии Plug and Play. К шине USB можно одновременно подключить до 127 устройств. Обмен по интерфейсу – пакетный, скорость обмена – 12 Мбит/с. В 2001 году появилась следующая спецификация интерфейса USB 2.0 (начальный стандарт теперь называется USB 1.1), обеспечивающая пропускную способность 480 Мбит/с. Поддерживается также дополнительный подканал со скоростью обмена данными в 1,5 Мбит/с для медленных устройств (клавиатуры, мыши, модема). Шина USB реализует как синхронный (нужный, например, при проведении телеконференций), так и асинхронный режимы передачи данных. Последняя версия USB 3.0 – появилась в 2008 году; скорость передачи возросла до 600 Мбайт/с.



*IEEE 1394 (Institute of Electrical and Electronic Engineers 1394* – стандарт Института инженеров по электротехнике и электронике) – последовательный интерфейс, предназначенный для подключения внутренних компонентов компьютера и внешних устройств. IEEE 1394 известен также под именем Fire Wire – «огненный провод» (хотя это название – собственность Apple и на компьютерах других производителей не применяется, как и цифровой интерфейс iLink компании Sony, также объединенный в стандарте IEEE 1394). Цифровой последовательный интерфейс Fire Wire характеризуется высокой надежностью и качеством передачи данных, его протокол поддерживает гарантированную передачу критичной по времени информации, обеспечивая прохождение видео- и аудио сигналов в реальном масштабе времени без заметных искажений. К одному контроллеру возможно подключение до 63 устройств на один порт с помощью единого шестижильного кабеля. Пропускная способность интерфейса составляет 100-400, 1600 Мбит/с. Этот интерфейс будет использоваться для подключения жестких дисков, дисководов CD-ROM и DVD-ROM, а также высокоскоростных внешних устройств, таких как цифровые видеокамеры, видеомагнитофоны и т.д.

*PCMCIA (Personal Computer Memory Card International Association* – ассоциация производителей плат памяти для персональных компьютеров) – внешняя шина компьютеров класса ноутбуков. Другое название модуля PCMCIA – PC Card. Шина имеет разрядность 16/26 (данные/адрес, адресное пространство - 64 Мбайт), поддерживает автоконфигурирование, возможно подключение и отключение устройств в процессе работы компьютера. Конструктив – миниатюр-

ный 68-контактный разъем. Контакты питания сделаны более длинными, что позволяет вставлять и вынимать карту при включенном питании компьютера.

*ACPI (Advanced Configuration Power Interface* – расширенный интерфейс конфигурирования и питания) – интерфейс, представляющий собой единую систему управления питанием для всех компонентов компьютера. Поддерживается новейшими модификациями BIOS материнских плат.

Помимо перечисленных выше шин в персональных компьютерах используется еще несколько интерфейсов. Рассмотрим те, которые касаются передачи видео и звука.

*VGA* это компонентный аналоговый видеоинтерфейс, используемый в мониторах и видеоадаптерах. Выпущен IBM в 1987 году. В настоящее время интерфейс уходит в прошлое.

*DVI* это стандарт на интерфейс и соответствующий разъем, предназначенный для передачи видеоизображения на цифровые устройства отображения, такие как жидкокристаллические мониторы, телевизоры и проекторы. Существуют как аналоговые, так и цифровые модификации. Пришел на смену VGA.

*HDMI* (High Definition Multimedia Interface) - интерфейс для мультимедиа высокой четкости, позволяющий передавать цифровые видеоданные высокого разрешения и многоканальные цифровые аудио сигналы с защитой от копирования.

Начало этого тысячелетия вообще знаменуется кардинальным обновлением интерфейсов ПК. Фирма Intel и ее сподвижники почти все порты (COM, LPT, PS/2) и параллельные интерфейсы (ISA, PCI, IDE-ATA) меняют на скоростные последовательные. Все больше системных плат, где есть только поддержка последовательных протоколов передачи данных, таки как USB, SATA и т.п.

## *Память персональных компьютеров*

1. Микропроцессорная память и кэш-память. Данный вопрос был детально рассмотрен выше.



2. Основная память.

2.1. Общие замечания по организации. Как было отмечено выше основная память содержит оперативное RAM и постоянное ROM запоминающие устройства. Оперативная память может составляться из микросхем динамического (Dynamic Random Access Memory – DRAM) или статического (Static Random Access Memory – SRAM) типа. SRAM строиться на триггерах, DRAM на своеобразных конденсаторах (см. выше). ROM в основном используется для хранения программ и информации, необходимой для нормальной работы ПК (см. выше)

2.2. Физическая структура основной памяти. В ПК используется матричная организация памяти. При матричной организации адрес ячейки, поступающий в регистр адреса, например по 20-разрядным *кодовым шинам адреса*, делится на две 10-разрядные части, поступающие соответственно в *Рег. адр. X* и *Рег. адр. Y*. Из этих регистров коды полуадресов поступают в дешифраторы *дешифратор X* и *дешифратор Y*, каждый из которых в соответствии с полученным адресом выбирает одну из 1024 шин. По выбранным шинам подаются сигналы записи-считывания в ячейку памяти, находящуюся на пересечении этих шин. Таким образом адресуется $1024^2$ ячеек.

Считываемая или записываемая информация поступает в *регистр данных (Рег. данных)*, непосредственно связанный с кодовыми шинами данных. Управляющие сигналы, определяющие, какую операцию следует выполнить, поступают по кодовым шинам инструкций. *Куб памяти* содержит набор запоминающих элементов – собственно ячеек памяти.

2.3. Типы модулей оперативной памяти (DRAM)

2.3.1. Различаются по типу корпусов самих микросхем памяти. Корпус *DIP* (Dual In-line Package – с двухрядным расположением выводов) – одиночная микросхема памяти, используется только в составе более укрупненных модулей. *SIP* (Single In-line Package – корпус с однорядным расположением выводов) – микросхема с одним рядом выводов, устанавливаемая вертикально. *SIPP* (Single In-line Pinned Package – корпус с однорядным расположением прово-

лочных выводов). Более современные варианты исполнения базируются на микросхемах с корпусами *SOIC* (Small-Outline Integrated Circuit) - предназначен для поверхностного монтажа, имеет форму прямоугольника с двумя рядами выводов по длинным сторонам; *TSOP* (Thin Small-Outline Package) – данные корпуса очень тонкие, низкопрофильные, являются разновидностью SOP микросхем; *BGA* ( Ball grid array - массив шариков) - тип корпуса поверхностно-монтируемых интегральных микросхем, выводы представляют собой шарики из припоя, нанесенные на контактные площадки с обратной стороны микросхемы, микросхему располагают на печатной плате, согласно маркировке первого контакта на микросхеме и на плате. Затем микросхему нагревают с помощью паяльной станции или инфракрасного источника, так что шарики начинают плавиться.

2.3.2. Различают по типу корпусов модулей памяти.

*SIMM* (Single In-line Memory Module) представляет собой печатную плату с односторонним краевым разъемом типа слот и установленными на ней совместимыми микросхемами памяти типа DIP. Микросхемы SIMM бывают двух разных типов: короткие на 30 контактов (длина 75 мм) и длинные на 72 контакта (длина 100 мм). Модули SIMM имеют емкость от 256 Кбайт, 1, 4, 8, 16, 32 и 64 Мбайт. Модули SIMM выпускаются с контролем и без контроля по четности и с эмуляцией контроля по четности. Память SIMM отличается также низким быстродействием – обычно она имеет время обращения 60 и 70 нс. Сейчас устарели.

*DIMM* (Dual In-line Memory Module) – более современные модули, имеющие 168-контактные разъемы (длина модуля 130 мм); могут устанавливаться только на те типы системных плат, которые имеют соответствующие разъемы. Появление DIMM стимулировалось выпуском процессоров Pentium, имеющих 64-битовую шину данных. В настоящее время используют разъемы на 184 контакта (для DDR SDRAM) и 240 контакта (для DDR2 и DDR3). Выпускают так же корпуса более компактной формы, например SO-DIMM.



*RIMM* (Rambus In-line Memory Module) – это технология основана на Direct Rambus DRAM, что потребовало нового конструктивного исполнения для модулей памяти. Модули RIMM требуют интенсивного охлаждения. Это связано со значительным энергопотреблением и, соответственно, тепловыделением, что обусловлено высоким быстродействием данных модулей памяти (время обращения 5 не и ниже). Хотя внешне модули RIMM напоминают модули DIMM, они имеют меньшее число контактов и с обеих сторон закрыты специальными металлическими экранами, которые защищают модули RIMM, работающие на больших частотах, экранируя их чувствительные электронные схемы от внешних электромагнитных наводок. В настоящее время менее распространены.

2.4. Технологии работы оперативной памяти.

*FPM DRAM* (Fast Page Mode DRAM) – динамическая память с быстрым страничным доступом, активно использовалась с микропроцессорами 80386 и 80486. Память со страничным доступом отличается от обычной динамической памяти тем, что после выбора строки матрицы и удержании RAS допускает многократную установку адреса столбца, стробируемого CAS. Это позволяет ускорить блочные передачи, когда весь блок данных или его часть находятся внутри одной строки матрицы, называемой в этой системе страницей. Существуют две разновидности FPM DRAM, различающиеся временем обращения: 60 и 70 нс. Ввиду своей медлительности они не эффективны в системах с процессорами уровня Pentium II и выше. Модули FPM DRAM в основном выпускались в конструктиве SIMM.

*EDO RAM* (EDO – Extended Data Out, расширенное время удержания данных на выходе) фактически представляет собой обычные микросхемы FPM, к которым добавлен набор регистров-защелок, благодаря чему данные на выходе могут удерживаться в течение следующего запроса к микросхеме. При страничном обмене такие микросхемы работают в режиме простого конвейера: удерживают на выходе содержимое последней выбранной ячейки, в то время как на их входы уже подается адрес следующей выбираемой ячейки. Это поз-

воляет примерно на 15% по сравнению с FPM ускорить процесс считывания последовательных массивов данных. При случайной адресации такая память никакого выигрыша в быстродействии не дает. Память типа RAM EDO имеет минимальное время обращения 45 нс и максимальную скорость передачи данных по каналу процессор-память 264 Мбайт/с. Модули RAM EDO выпускались в конструктивах SIMM и DIMM. Существовали так же варианты типа *BEDO DRAM* (Burst Extended Data Output, EDO с блочным доступом), с более увеличенными скоростями передачи.

*SDRAM* (Synchronous DRAM – синхронная динамическая память), память с синхронным доступом, увеличивает производительность системы за счет синхронизации скорости работы ОЗУ со скоростью работы шины процессора. SDRAM также осуществляет конвейерную обработку информации, выполняя внутреннее разделение массива памяти на два независимых банка, что позволяет совмещать выборку из одного банка с установкой адреса в другом банке. SDRAM также поддерживает блочный обмен. Основная выгода от использования SDRAM состоит в поддержке последовательного доступа в синхронном режиме, где удается исключить дополнительные такты ожидания. Память SDRAM может устойчиво функционировать на высоких частотах: выпускаются модули, рассчитанные на работу при частотах 100 МГц (спецификация PC100) и выше. Модули памяти именуемые ESDRAM имеют встроенный кэш.

*DDR SDRAM* (Double Data Rate SDRAM - SDRAM 2). Вариант памяти SDRAM, осуществляющий передачу информации по обоим фронтам тактового сигнала. Это позволяет удвоить пропускную способность по сравнению с традиционной памятью SDRAM (до 1,6 Гбайт/с при частоте шины 100 МГц). Кроме того, DDR SDRAM может работать на более высокой частоте. Модули обозначаемые DDR200, DDR266, DDR333, DDR400 работают соответственно на частотах 200, 266, 333, 400 МГц.

*Примечание*: в маркировке памяти часто встречается обозначение PCXXX. Модули DDR имеют маркировку PC. Но в отличие от SDRAM, где PC обозначало частоту работы (например, PC133 – память предназначена для работы на частоте 133МГц), показатель PC в модулях DDR указывает на максимально достижимую пропускную способностью, измеряемую в мегабайтах в секунду. В этом случае частота умножается на ширину шины, по которой передаются данные, т.е. на 8 (64 бита), и еще на два. Таким образом, PC1600 – это память DDR SDRAM, работающая на частоте 100 МГц (100x8x2=1600). PC2100 – частота 133 МГц.



*DRDRAM* (Direct Rambus DRAM) – перспективный тип оперативной памяти, позволивший значительно увеличить производительность компьютеров. Высокое быстродействие памяти Direct RDRAM достигается рядом особенностей, не встречающихся в других типах. В частности, применением собственной двухбайтовой шины Rambus с частотой 800 МГц, обеспечивающей пиковую пропускную способность до 1,6 Гбайт/с. Контроллер памяти Direct RDRAM управляет шиной Rambus и обеспечивает преобразование ее протокола с частотой 800 МГц в стандартный 64-разрядный интерфейс с частотой шины до 200 МГц. Была дороже, чем варианты DDR, что способствовало ее проигрышу последней на рынке.

*DDR2 SDRAM* – конструктивно новый тип оперативной памяти DDR2 SDRAM был выпущен в 2004 году. За счет технических изменений показывает более высокое быстродействие и предназначен для использования на современных компьютерах. Память может работать с тактовой частотой шины 200, 266, 333, 337, 400, 533, 575 и 600 МГц. При этом эффективная частота передачи данных соответственно будет 400, 533, 667, 675, 800, 1066, 1150 и 1200 МГц.

*DDR3 SDRAM* – этот тип памяти основан на технологиях DDR2 SDRAM со вдвое увеличенной частотой передачи данных по шине памяти. Отличается пониженным энергопотреблением по сравнению с предшественниками. Частота полосы пропускания лежит в пределах от 800 до 2400 МГц (рекорд частоты — более 3000 МГц), что обеспечивает большую пропускную способность по сравнению со всеми предшественниками.

*DDR4 SDRAM* (double-data-rate four synchronous dynamic random access memory) - новый тип оперативной памяти, являющийся эволюционным разви-

тием предыдущих поколений DDR (DDR, DDR2, DDR3). Отличается повышенными частотными характеристиками и пониженным напряжением. Основное отличие DDR4 заключается в удвоенном до 16 числе банков, что позволило вдвое увеличить скорость передачи — до 3,2 Гбит/с. Пропускная способность памяти DDR4 достигает 34,1 ГБ/с (в случае максимальной эффективной частоты 4266 МГц, определенной спецификациями). Кроме того, повышена надежность работы за счет введения механизма контроля чётности на шинах адреса и команд. Будет поддерживать эффективные частоты от 1600 до 4266 МГц. Массовое производство 2013-2014 год.

Характеристики отдельных видов памяти представлены в таблице.

Табл. 2.1. Типы памяти

| Год выпуска | Тип памяти | Тактовая частота, МГц | Разрядность шины, бит | Пиковая пропускная способность, Мбайт/с |
|---|---|---|---|---|
| 1987 | FPM DRAM | 33 | 32 | 132 |
| 1995 | EDO DRAM | 50 | 32 | 200 |
| 1997 | SDRAM | 66 | 64 | 528 |
| 1998 | SDRAM | 100 | 64 | 800 |
| 2000 | SDRAM | 133 | 64 | 1064 |
| 2000 | SDRAM | 150 | 64 | 1200 |
| 1999 | D RDRAM. 1 канал | 600 | 16 | 1200 |
| 2000 | DRDRAM, 1 канал | 800 | 16 | 1600 |
| 2001 | DRDRAM. 2 канала | 600 | 16 | 2400 |
| 2001 | DRDRAM, 2 канала | 800 | 16 | 3200 |
| 2002 | DRDRAM. 2 канала | 1066 | 16 | 4300 |
| 2001 | DDR SDRAM, 2 канала | 100 | 64 | 1600 |
| 2001 | DDR SDRAM, 2 канала | 150 | 64 | 2400 |
| 2001 | DDR2 SDRAM, 2 канала | 100 | 64 | 3200 |
| 2003 | DDR2 SDRAM, 2 канала | 200 | 64 | 6400 |
| 2004 | DDR2 SDRAM, 2 канала | 266 | 64 | 8500 |
| 2005 | DDR2 SDRAM, 2 канала | 400 | 64 | 12800 |
| 2005 | DDR2 SDRAM, 2 канала | 800 | 64 | 25000 |
| 2007 | DDR2, SDRAM 2 канала | 1066 | 64 | 32000 |
| 2006 | DDR3 SDRAM, 2 канала | 800 | 64 | 50000 |
| 2007 | DDR3 SDRAM, 2 канала | 1066 | 64 | 64000 |

Существуют так же специальные виды оперативной памяти, аналогичные по работе DDR, и использующиеся например в графических ускорителях, например GDDR2, GDDR3, GDDR4, GDDR5. Они отличаются удвоенными частотами и специальными конструктивными решениями.



*Постоянные запоминающие устройства*

Постоянное запоминающее устройство (ПЗУ или ROM – *Read Only Memory,* память только для чтения) также строится на основе установленных на материнской плате модулей и используется для хранения неизменяемой информации: загрузочных программ операционной системы, программ тестирования устройств компьютера и некоторых драйверов базовой системы ввода-вывода BIOS и т. д. К ПЗУ принято относить энергонезависимые постоянные и полупостоянные запоминающие устройства, из которых оперативно можно только считывать информацию, запись информации в ПЗУ выполняется вне ПК в лабораторных условиях или при наличии специального программатора – в компьютере.

По технологии записи информации можно выделить ПЗУ следующих типов: микросхемы, программируемые только при изготовлении – классические или масочные ПЗУ или ROM; микросхемы, программируемые однократно в лабораторных условиях – программируемые ПЗУ (ППЗУ) или programmable ROM (PROM); микросхемы, программируемые многократно – перепрограммируемые ПЗУ или erasable PROM (EPROM). Среди них следует отметить электрически перепрограммируемые микросхемы EEPROM (Electrical Erasable PROM), в том числе флэш-память.

В настоящее время в ПК используются полупостоянные, перепрограммируемые запоминающие устройства – флэш-память. Модули, или карты, флэш-памяти могут устанавливаться прямо в разъемы материнской платы, либо выпускаться в различных корпусах.

Флэш-память бывает весьма полезной как для создания весьма быстродействующих, компактных, альтернативных НМД запоминающих устройств – твердотельных дисков, так и для замены ПЗУ, хранящего программы BIOS, позволяя обновлять и заменять эти программы на более новые версии при модернизации ПК.

*Внешние запоминающие устройства (общие замечания)*

Разные вид накопителей были рассмотрены выше; современные подходы при реализации данных устройств – см. ниже; дополнительный материал размещен на презентациях.

Здесь же остановимся на некоторых особенностях организации и использования внешних запоминающих устройств, используемых в ПК.

*Файловая система* (ФС). Это совокупность программ, обеспечивающая выполнение операций над файлами. В ОС для ПК используются десятки файловых систем: в DOS – FAT16, FAT32 и FAT12 для гибких дисков (FAT – File Allocation Table, таблица размещения файлов), для Windows 9x характерны FAT16 и FAT32, популярной в последних версиях Windows NTFS, для OS/2 - HPFS, для Linux - Ext2(3,4) и т. д.

*НЖМД*. НЖМД весьма разнообразны. Диаметр дисков чаще всего 3,5 дюйма (89 мм) или 2,5. Внешние дорожки диска длиннее внутренних. Поэтому в современных жестких дисках используется метод зонной записи. В этом случае все пространство диска делится на несколько зон, причем во внешних зонах секторов размещается больше данных, чем во внутренних.

Время доступа к информации на диске напрямую связано со скоростью вращения дисков. Стандартные скорости вращения – 3600, 4500, 5400 и 7200 оборотов/мин; при интерфейсе SCSI используются скорости до 10 000 и даже до 12 000 оборотов/мин. Для повышения скорости обмена данными процессора с дисками НЖМД следует кэшировать. Кэш-память для дисков имеет то же функциональное назначение, что и кэш для основной памяти, то есть служит быстродействующим буфером для кратковременного хранения информации, считываемой или записываемой на диск. Кэш-память может быть встроенной в дисковод, а может создаваться программным путем в оперативной памяти.

Для того чтобы получить на магнитном носителе структуру диска, включающую в себя дорожки и секторы, над ним должна быть выполнена процедура, называемая физическим, или низкоуровневым, форматированием (physical, или low-level formatting). В ходе выполнения этой процедуры контроллер запи-



сывает на носитель служебную информацию, которая определяет разметку цилиндров диска на секторы и нумерует их. Форматирование низкого уровня предусматривает и маркировку дефектных секторов для исключения обращения к ним в процессе эксплуатации диска.

Существует и технология SMART (Self-Monitoring Analysis and Reporting Technology) – технология самотестирования и анализа, осуществляющая автоматическую проверку целостности данных, состояния поверхности дисков, перенос информации с критических участков на нормальные и другие операции без участия пользователя. Кроме того, при появлении и нарастании серьезных ошибок SMART своевременно выдает сообщение о необходимости принятия мер по спасению данных.

В ПК имеется обычно один, реже несколько накопителей на жестких магнитных дисках. Программными средствами один физический диск может быть разделен на несколько «логических» дисков; тем самым имитируется несколько НМД на одном накопителе.

Внешние HDD относятся к категории переносных. Последнее время переносные накопители (их также называют внешними, мобильными, съемными, а портативные их варианты – карманными – Pocket HDD) получили широкое распространение. Питание переносных жестких дисков выполняется либо от клавиатуры, либо по шине USB.

Переносные жесткие диски весьма разнообразны: от обычных HDD в отдельных корпусах до стремительно набирающих популярность твердотельных дисков.

Устройства флэш-памяти

Флэш-диски (Flash Disks) – весьма популярный и очень перспективный класс энергонезависимых запоминающих устройств. Флэш-диски (твердотельные диски) являются модификацией HDD и представляют собой устройства для долговременного хранения информации с возможностью многократной перезаписи. Стирание и запись данных осуществляется так же, как у HDD – блоками (иногда называемыми по аналогии с магнитными дисками секторами, но более

правильно было бы их именовать кластерами). У флэш-дисков отсутствуют какие-либо подвижные части, да и форма у них совсем не круглая – чаще всего они представляют собой прямоугольные картриджи. Для хранения информации в них используются специализированные микросхемы памяти с металлизацией (металл-нитридные), выполненные по технологии Flash, изобретенной в начале 80-х годов в фирме Intel. Дисками их называют условно, поскольку флэш-диски полностью эмулируют функциональные возможности HDD. При работе указатели в микросхеме перемещаются на начальный адрес блока, затем байты данных передаются в последовательном порядке с использованием стробирующего сигнала. Стирание содержимого всего блока выполняется одномоментно отдельным сигналом (отсюда, вероятно, и название памяти flash – вспышка); тотальное стирание было специально организовано разработчиками, поскольку первоначально флэш-память применялась в военных приборах, и при обнаружении попыток несанкционированного доступа к ним необходимо было сразу уничтожать все данные – система автоматически генерировала внутренний сигнал стирания).

По существу, флэш-диски – это «полупостоянные» запоминающие устройства, стирание, считывание и запись информации в которых выполняется электрическими сигналами. Количество циклов перезаписи информации в одну и ту же ячейку у флэш-памяти ограничено. В современных устройствах имеются программные или аппаратные средства формирования виртуальных блоков, обеспечивающие запись информации поочередно в разные области флэш-памяти так, чтобы число циклов стирания и записи было равномерно распределено по всем блокам диска. Это существенно увеличивает срок службы флэш-памяти: ее работоспособность сохраняется сотни лет.

Емкость современных флэш-дисков, изготовленных на основе многоуровневых ячеек (MultyLevel Cell, MLC) на базе логических схем NAND (≪Не-И≫, штрих Шеффера), достигает многих гигабайтов при крайне миниатюрных размерах. Флэш-диски обладают высочайшей надежностью – среднее время наработки на отказ (Mean Time Between Failures – MTBF) у них составляет, как



правило, более миллиона часов; они устойчивы к механическим ускорениям и вибрациям, работают в широком диапазоне температур (от -40 до +85 °С). Во время выполнения операций чтения-записи флэш-диски обычно потребляют не более 200 мВт электроэнергии и, естественно, не шумят.

Флэш-диски в настоящее время выпускаются многими фирмами, с различными интерфейсами и в разных конструктивных исполнениях. Они могут быть не только внешними дисками ПК, но и устанавливаться внутри системного блока. В качестве фиксированной памяти используются флэш-карты, выполненные в виде печатных плат, предназначенных для непосредственной установки в разъемы системной платы компьютера. Они способны работать с системными и локальными интерфейсами ПК (ISA, PCI и др.). Значительно чаще флэш-память используется в качестве альтернативных HDD твердотельных дисков. В этом случае востребованы периферийные интерфейсы ATA (IDE), Serial ATA, USB, IEEE 1394 и др. Широкое применение флэш-диски нашли в цифровых фото и видеокамерах. На CES 2013 Kingston показала "флэшку" DataTraveler HyperX Predator объемом 1 терабайт с поддержкой стандарта USB 3.0. Показатели скорости чтения/записи у нее составляют 240 и 160 мегабайт в секунду соответственно.

## Дисковые массивы RAID

В машинах-серверах баз данных и в суперкомпьютерах часто применяются *дисковые массивы RAID* (Redundant Array of Inexpensive Disks – массив недорогих дисков с избыточностью), в которых несколько запоминающих устройств на жестких дисках объединены в один большой накопитель, обслуживаемый специальным RAID-контроллером. Отличительной особенностью RAID-массивов является то, что в них используются основанные на введении информационной избыточности методы обеспечения достоверности информации, существенно повышающие надежность работы системы (при обнаружении искаженной информации она автоматически корректируется, а неисправный накопитель в режиме Plug and Play замещается исправным).

В качестве концепции компоновки дисковых массивов RAID была впервые представлена в 1987 году инженерами из калифорнийского университета в Беркли, которые описали пять уровней конфигурации RAID (RAID 1-5). Позже к ним были добавлены RAID 0 и RAID 6.

- 0-й уровень осуществляет расщепление дисков (disk stripping), записывая данные в виде дорожек поочередно на каждом диске массива без контроля четности. Это единственный уровень, не обеспечивающий устойчивость к отказам;

- 1-й уровень подразумевает два диска, второй из которых является точной копией (зеркальной) первого. Метод RAID 1 подходит для системных и загрузочных разделов;

- 2-й уровень использует несколько дисков специально для хранения контрольных сумм и обеспечивает самый сложный функционально и самый эффективный метод исправления ошибок;

- 3-й уровень включает четыре диска: три являются информационными, а последний хранит контрольные суммы, предназначенные для исправления ошибок в первых трех;

- 4-й и 5-й уровни используют диски, на каждом из которых хранятся свои собственные контрольные суммы;

- 6-й уровень – RAID 5, дополненный резервными дисковыми контроллерами, вентиляторами, шинами и др.

Дисковые массивы второго (RAID6, RAID 7) и третьего (RAID 10, RAID 30, RAID 50) поколений используют различные сочетания базовой компоновки.

Современные дисковые массивы могут объединять 160 и более физических дисков любой емкости, формирующих логические диски; имеют внутренний кэш и разъемы для подключения внешних интерфейсов типа SCSI или Fibre Channel.

Гибкие магнитные диски, диски высокой плотности, оптические диски – см. презентации.



**Зачетные вопросы, рассматриваемые в теме**

1. Организация микропроцессоров: понятие микропроцессора, его основные особенности, преимущества использования; функциональная структура МП.

2. Организация микропроцессоров: предназначение, характеристика, состав и принцип работы операционного блока.

3. Организация микропроцессоров: предназначение, характеристика, состав и принцип работы блока управления и интерфейсного блока.

4. Особенности организации процессоров при использовании внутренних регистров.

5. Организация памяти ВМ: общая структура и характеристики каждого уровня.

6. Организация памяти ВМ: организация оперативного, сверх оперативного уровней; методы управления оперативной памятью.

7. Системы внешней памяти и особенности организации кэш-памяти. Методы повышения пропускной способности оперативной памяти.

8. Организация обмена данными в ВМ – общие сведения. Обмен данными между периферийными устройствами и вычислительным ядром системы.

9. Функциональная и структурная организация, характеристики современных ПК.

10. Системная плата, внутримашинный и шинный интерфейсы, основные используемые шины и элементы современных ПК.

11. Современные персональные компьютеры, особенности архитектуры и применения: запоминающие устройства и внешняя периферия.

**Примечания**

При разработке данной темы использовались издания [5,8], содержание которых было дополнено актуальной информацией полученной из открытых источников.

При изучении настоящей темы так же рекомендуется обратиться к соответствующим презентациям по дисциплине, где представлены в качестве сбавочных данных некоторые сведения по имеющимся в настоящее время достижениям в области компьютерных, сетевых и информационных технологий.



# ТЕМА № 3. СОВРЕМЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

*Вопросы, рассматриваемые в данной теме*:

1. Однопроцессорные и многопроцессорные системы. Таксономия М. Флина. Однопроцессорные конфигурации.
2. Централизованные и распределенные системы обработки данных. Вычислительные системы параллельной обработки. Многопроцессорные и многоядерные конфигурации.
3. Вычислительные системы – состояние, производительность, направления развития.

## Теоретические сведения

В рамках данной темы рассматривается третья часть главы 1 «История развития, архитектура и базовые понятия компьютерных технологий».

### *1. Однопроцессорные и многопроцессорные системы. Таксономия М. Флина. Однопроцессорные конфигурации.*

С точки зрения способов использования процессоров в настоящее время выделилось две тенденции: применение однопроцессорных и многопроцессорных конфигураций. Причем первая группа, по сути, разилась до технологий, где используются многоядерные конфигурации. Такие системы с одной стороны являются многопроцессорными решениями, а с другой стороны отличаются от классический мультипроцессоров.

При этом отдельная группа систем основанных на использовании микроконтроллеров – изначально однопроцессорных систем, с малым функционалом, предназначенных для решения конкретных задач, стала основой новым решениям. Здесь по-прежнему в одном корпусе имеется функционально завершен-

ная система, но теперь с функционалом и характеристиками, превосходящими обычные ПК (например, всевозможные гаджеты на архитектуре ARM).

Исторически первыми появились однопроцессорные архитектуры. Классическим примером однопроцессорной архитектуры является архитектура фон Неймана со строго последовательным выполнением команд: процессор по очереди выбирает команды программы и также по очереди обрабатывает данные (программа и данные хранятся в единственной последовательно адресуемой памяти).

По мере развития вычислительной техники архитектура фон Неймана обогатилась сначала конвейером команд, а затем многофункциональной обработкой, и по таксономии* М.Флина получила обобщенное название компьютера с одним потоком команд и одним потоком данных.

Заметим, что данная классификация является лишь еще одним способом привести к единой системе разнообразный компьютерный мир, и конечно не совершенна. Более того на рис. 3.1 нет четкого деления систем на однопроцессорные и многопроцессорные решения, что было бы логичным. Это обстоятельство продиктовано тем фактом, что как будет показано ниже даже в случае одиночного потока команд – процессорных элементов, выполняющих обработку, может быть несколько. Поэтому здесь и далее будем считать что процессор может быть достаточно сложен по архитектуре, содержать несколько процессорных элементов или ядер. В случае многопроцессорной конфигурации процессоров как самостоятельных единиц может быть несколько.

Здесь ключевым понятием являются так называемые потоки команд и данных. *Поток команд* - это последовательность команд, выполняемых ЭВМ (системой), *поток данных* - последовательность данных (исходная информация и промежуточные результаты решения задачи), обрабатываемых под управлением потока команд.

---

* теория классификации и систематизации сложноорганизованных областей действительности, имеющих обычно иерархическое строение *©2001 "Большая Российская энцикл."*



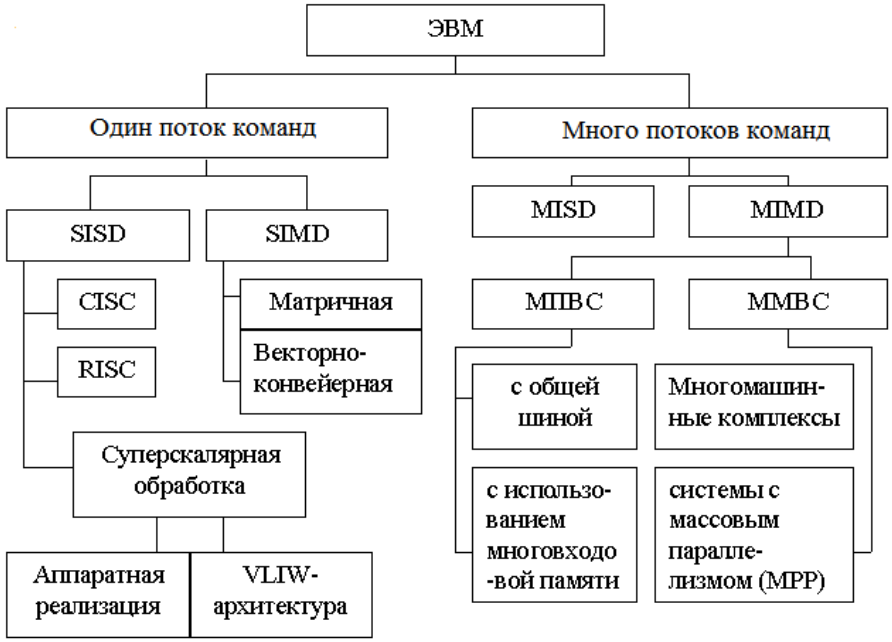



*Рис. 3.1.* Классификация вычислительно техники

## *SISD-компьютеры*

*SISD* (Single Instruction Single Data) или ОКОД - один поток команд, один поток данных.

*Рис. 3.2.* SISD-архитектура

SISD компьютеры это обычные, «традиционные» последовательные машины, в которых в каждый момент времени выполняется лишь одна операция над одним элементом данных (числовым или каким-либо другим значением). При работе такой системы в мультипрограммном режиме, когда совместно решаются несколько задач (программы и исходные данные по каждой из них хра-

нятся в оперативной памяти), обеспечивается параллельная работа устройств системы, происходит разделение времени и оборудования между совместно выполняемыми программами. Но в каждый данный момент операционное устройство (АЛУ), поскольку оно является единственным, занимается обработкой информации по какой-то одной команде, т. е. одновременное преобразование информации в АЛУ по нескольким командам, принадлежащим разным участкам одной и той же программы или разным программам, невозможно. Основная масса первых ЭВМ функционировала в соответствии с принципом фон Неймана и имела структура класса SISD. Данная структура породила CISC, RISC и архитектуру с суперскалярной обработкой.

Компьютеры с CISC архитектурой

*Компьютеры с CISC* (Complex Instruction Set Computer) архитектурой имеют комплексную (полную) систему команд, под управлением которой выполняются всевозможные операции типа «память-память», «память-регистр», «регистр-память», «регистр-регистр». Данная архитектура характеризуется:

- большим числом команд (более 200);
- переменной длиной команд (от 1 до 11 байт и более);
- значительным числом способов адресации и форматов команд;
- сложностью команд и многотактностью их выполнения;
- наличием микропрограммного управления, что снижает быстродействие и усложняет процессор.

Обмен с памятью в процессе выполнения команды делает практически невозможной глубокую конвейеризацию арифметики, что ограничивает его производительность.

Большинство первых компьютеров типа IBM PC относились к CISC архитектуре, например, компьютеры с микропроцессорами 8080, 80486, 80586 (товарная марка Pentium).

Компьютеры с RISC архитектурой

*Компьютеры с RISC* (Reduced Instruction Set Computer) архитектурой содержат набор простых, часто употребляемых в программах команд.



Основными являются операции типа «регистр-регистр».

Данная архитектура характеризуется:

- сокращенным числом команд;
- большинство команд выполняется за один машинный цикл;
- постоянная длина команд;
- небольшое количество способов адресации и форматов команд;
- нет необходимости в использовании микропрограммного управления;
- большим числом регистров процессора.

Компьютеры с RISC-архитектурой «обязаны» иметь преимущество в производительности по сравнению с CISC компьютерами, за которое приходится расплачиваться наличием в программах дополнительных команд обмена регистров процессора с оперативной памятью.

*Примечание*. Такие системы в свое время нашли широкое применение в микроконтроллерах, о которых будет сказано позднее. Заметим так же что достижения CISC и RISC впоследствии стали использоваться вместе, для построения современных процессоров. При этом подход RISC оказался очень перспективным, т.к. четкие указания в виде простых команд позволяют более эффективно использовать возможности всей системы.

Компьютеры с суперскалярной обработкой

Еще одной разновидностью однопотоковой архитектуры *является суперскалярная обработка*. Смысл этого термина заключается в том, что в аппаратуру процессора закладываются средства, позволяющие одновременно выполнять две или более скалярные операции, т.е. команды обработки пары чисел. Суперскалярная архитектура базируется на многофункциональном параллелизме и позволяет увеличить производительность компьютера пропорционально числу одновременно выполняемых операций. Способы реализации суперскалярной обработки могут быть разными.

*Аппаратная реализация суперскалярной обработки п*рименяется как в CISC, так и в RISC - процессорах и заключается в чисто аппаратном механизме выборки из буфера инструкций (или кэша инструкций) несвязанных команд и

параллельном запуске их на исполнение. Этот метод хорош тем, что он «прозрачен» для программиста, составление программ для подобных процессоров не требует никаких специальных усилий, ответственность за параллельное выполнение операций возлагается в основном на аппаратные средства.

*VLIW-архитектуры суперскалярной обработки.* Второй способ реализации суперскалярной обработки заключается в кардинальной перестройке всего процесса трансляции и исполнения программ. Уже на этапе подготовки программы компилятор группирует несвязанные операции в пакеты, содержимое которых строго соответствует структуре процессора. Например, если процессор содержит функционально независимые устройства (сложения, умножения, сдвига и деления), то максимум, что компилятор может «уложить» в один пакет - это четыре разнотипные операции; (сложение, умножение, сдвиг и деление). Сформированные пакеты операций преобразуются компилятором в командные слова, которые по сравнению с обычными инструкциями выглядят очень большими. Отсюда и название этих суперкоманд и соответствующей им архитектуры - VLIW (Very Large Instruction Word - очень широкое командное слово). По идее, затраты на формирование суперкоманд должны окупаться скоростью их выполнения и простотой аппаратуры процессора, с которого снята вся «интеллектуальная» работа по поиску параллелизма несвязанных операций. Однако практическое внедрение VLIW-архитектуры затрудняется значительными проблемами эффективной компиляции.

*Примечание:* Технология VLIW в большей степени требует наличия нескольких процессорных модулей, возможно АЛУ; формально это не многоядерная архитектура, а дальнейшее развитие RISC подхода, где есть четкие указания, что делать каждому модулю. Между тем VLIW использовали такие процессоры как Intel Itanium, Эльбрус. Такая технология так же предложена ATI для своих графических процессоров (начиная с HD2000). В настоящее время фактически используется в процессорах Radeon линейки HD6000 (40 нм, часто-



та ядра 880 МГц, число процессоров Steam 1536, 2.7 TFLOPS, 2 Гб с GDDR5 и частотой 1375 МГц) [1]. Сама карта сейчас устарела.

Архитектуры класса SISD охватывают те уровни программного параллелизма, которые связаны с одиночным потоком данных. Они реализуются многофункциональной обработкой и конвейером команд.

Параллелизм циклов и итераций тесно связан с понятием множественности потоков данных и реализуется векторной обработкой. В таксономии компьютерных архитектур М. Флина выделена специальная группа однопроцессорных систем с параллельной обработкой потоков данных – SIMD.

## *SIMD-компьютеры*

*SIMD* (Single Instruction Stream - Multiple Data Stream) или ОКМД - один поток команд и множество потоков данных. SIMD компьютеры состоят из одного командного процессора (управляющего модуля), называемого контроллером, и нескольких модулей обработки данных, называемых процессорными элементами.

*Управляющий модуль* принимает, анализирует и выполняет команды. Если в команде встречаются данные, контроллер рассылает на все процессорные элементы команду, и эта команда выполняется на нескольких или на всех процессорных элементах.

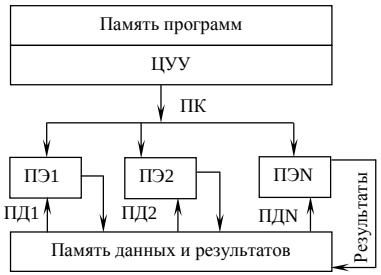



*Рис. 3.3.* SIMD-архитектура

[1] По материалам *Компьютер Билд,*. – №01(153). – 2012. – С.8.

Все *процессорные элементы* идентичны и каждый из них представляет собой совокупность управляюще-обрабатывающего органа (быстродействующего решателя) и процессорной памяти небольшой емкости. Процессорные элементы выполняют операции параллельно над разными потоками данных (ПД) под управлением общего потока команд (ПК), вследствие чего такие ЭВМ называются системами с общим потоком команд. В любой момент в каждом процессорном элементе выполняется одна и та же команда, но обрабатываются различные данные. Реализуется синхронный параллельный вычислительный процесс.

Одним из преимуществ данной архитектуры считается то, что в этом случае более эффективно реализована логика вычислений. До половины логических инструкций обычного процессора связано с управлением выполнением машинных команд, а остальная их часть относится к работе с внутренней памятью процессора и выполнению арифметических операций. В SIMD компьютере управление выполняется контроллером, а «арифметика» отдана процессорным элементам. Возможны два способа построения компьютеров этого класса. Это матричная структура ЭВМ и векторно-конвейерная обработка.

Матричная архитектура

Суть *матричной архитектуры* заключается в том, что имеется множество процессорных элементов, исполняющих одну и ту же команду над различными элементами вектора (потоков данных), объединенных коммутатором. Каждый процессорный элемент включает схемы местного управления, операционную часть, схемы связи и собственную оперативную память. Изменение производительности матричной системы достигается за счет изменения числа процессорных элементов.

Основные их преимущества - высокая производительность и экономичность. Недостатки матричных систем, ограничивающие области их применения, заключаются в жесткости синхронного управления матрицей процессор-



ных элементов и сложности программирования обмена данными между процессорными элементами через коммутатор.

Они применяются главным образом для реализации алгоритмов, допускающих параллельную обработку многих потоков данных по одной и той же программе (одномерное и двумерное прямое и обратное преобразования Фурье, решение систем дифференциальных уравнений в частных производных, операций над векторами и матрицами и др.). Матричные системы довольно часто используются совместно с универсальными однопроцессорными ЭВМ. Примером векторных супер-ЭВМ с матричной структурой является знаменитая в свое время система ILLIAC-IV.

*Векторно-конвейерная архитектура*

В отличие от матричной, *векторно-конвейерная структура* компьютера содержит конвейер операций, на котором обрабатываются параллельно элементы векторов и полученные результаты последовательно записываются в единую память. При этом отпадает необходимость в коммутаторе процессорных элементов, служащем камнем преткновения в матричных компьютерах.

Векторно-конвейерную структуру имеют однопроцессорные супер-ЭВМ серии VP фирмы Fujitsu; серии S компании Hitachi; C90, M90, T90 фирмы Cray Research; Cray-3, Cray-4 фирмы Cray Computer и т.д.

Общим для всех векторных суперкомпьютеров является наличие в системе команд векторных операций, допускающих работу с векторами определенной длины, допустим, 64 элемента по 8 байт. В таких компьютерах операции с векторами обычно выполняются над векторными регистрами.

MMX технология

Еще одним примером SIMD-архитектуры является *технология MMX*, которая существенно улучшила архитектуру микропроцессоров фирмы Intel. Технология MMX представляет собой компромиссное решение, объединяющее пути, используемые в классическом процессоре CISC-архитектуры (Pentium), в компьютерах с параллельной SIMD-архитектурой, с добавлением ряда простых (RISC) команд параллельной обработки данных. Она разработана для ускоре-

ния выполнения мультимедийных и коммуникационных программ с добавлением новых типов данных и новых инструкций. Технология в полной мере использует параллелизм SIMD-архитектуры и сохраняет полную совместимость с существующими операционными системами и приложениями для SISD.

### *2. Централизованные и распределенные системы обработки данных. Вычислительные системы параллельной обработки. Многопроцессорные и многоядерные конфигурации.*

Основной задачей развития компьютерных технологий является удовлетворение потребностей человека в обработке все большего количества информации. Не затрагивая вопросы имеющихся проблем связанных с улучшением производительности самих компьютерных систем (которые были озвучены выше, а способы по решению представлены ниже), эргономики, организации человеко-ориентированного интерфейса здесь обсудим другой аспект, связанный с обработкой данных. А именно то, каким образом может быть построена вычислительная система, где решаются задачи обработки – в едином центре, или на периферии – в отдельных ее узлах, связь между которыми происходит по специализированным каналам.

Заметим, что данный вопрос опирается на тему 3 настоящего пособия, но может быть рассмотрен и без нее, т.к. здесь ставятся другие цели. В рамках данного пункта основной задачей является знакомство с общими понятиями централизованных и распределенных систем обработки данных. При этом будет взят конкретный пример – построение системы для задач управления технологическим процессом. Этот пример позволит обсудить еще одну область применения систем на основе микропроцессоров – применение во встраиваемых приложениях. Полученные же здесь знания позволят успешно самостоятельно проанализировать и такие области использования систем обработки данных как реализация центров обработки данных – ЦОД, масштабных систем сбора и анализа информации и т.п.



Все микропроцессорные системы (в состав которых входит микропроцессор) можно разделить на три класса:

- вычислительны машины (ВМ) и системы класса ПК, ноутбук или подобные предназначенные для решения задач обычного пользователя (обладают средними характеристиками, ориентированы на мультимедиа и организацию удобного человека-машинного интерфейса);

- вычислительные системы (ВС), представляющие собой сложные структуры и предназначенные для обработки большого объема специализированной информации по сложным алгоритмам (обладают большими характеристиками в какой то конкретной области, другие параметры могут существенно отставать от класса ПК), например серверы, суперЭВМ, кластеры и т.п.;

- микроконтроллерные системы, представляющие собой системы, использующиеся в быту, во встраиваемых приложениях, в военной индустрии, авиации и т.п. (большая часть параметров намного уступает в характеристике классу ПК, но простота и надежность работы, дешевая структура, реализация всей архитектуры в одном корпусе – делает их незаменимыми для многих задач)

Первая группа достаточно хорошо изучена, знакомство со второй уже произошло, и продолжиться в третьей теме, а вот третья группа и будет рассмотрена здесь.

<u>Управление техническим объектом</u>

Основой для реализации алгоритмов управления в технических системах является универсальность средств вычислительной техники, поскольку они обеспечивают возможность решения любой алгоритмически разрешимой задачи.

В состав типовой системы управления (рис. 3.4) входят:

- *объект управления*, содержащий собственно объект или процесс, исполнительные устройства и систему датчиков контролируемых параметров объекта;

- *блок устройств сопряжения с объектом*, обеспечивающий согласование сигналов (как управляющих, так и информационных) системы и объекта

управления; пульт управления, предоставляющий оператору возможность формировать параметры процесса управления и вносить при необходимости в него коррективы;

- *микропроцессорная система управления* или цифровой регулятор.

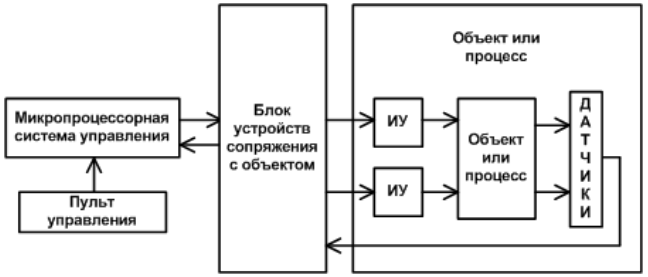



*Рис. 3.3.* Обобщенная структура типовой системы управления

Конкретная реализация цифрового регулятора определяется типом используемой системы обработки и зависит от сложности объекта управления. Более сложные системы требуют использования сложных регуляторов, которые могут быть построены как на основе ВМ общего назначения, например класса ПК в промышленном исполнении, так и на основе ВС. Следует отметить, что во многих случаях функциональные возможности ВМ или ВС для решения задач управления оказываются избыточными и большинство цифровых регуляторов чаще всего реализуют в виде микроконтроллерных систем.

Цифровой регулятор получает необходимую информацию о текущем состоянии объекта от измерительных преобразователей – датчиков. Управляющие воздействия поступают на объект через исполнительные устройства. Для связи цифрового регулятора с датчиками и исполнительными устройствами используются специальные электронные схемы – устройства сопряжения с объектом УСО. В связи с тем что обмен информацией между цифровым регулятором и УСО осуществляется с помощью цифровых и аналоговых сигналов, в составе УСО имеются цифроаналоговые и аналого-цифровые преобразователи, буферные усилители сигналов, элементы гальванической развязки электрических цепей. При реализации цифрового регулятора на основе ВМ его связь с УСО



осуществляется через порты ввода-вывода. Для организации информационного взаимодействия управляющих программ ВМ с портами УСО используются специальные системные программы – драйверы.

Сложные системы управления включают в контур управления не только ВМ, но и человека и являются не автоматическими, а автоматизированными. При этом функции, возлагаемые на ВМ, усложняются. Требуется обеспечить интерфейс между человеком и ВМ: индикацию состояния управляемого процесса в графической, символьной или числовой форме представления информации, ввод управляющих воздействий от человека, корректирующих ход управляемого процесса. Во всех случаях в системах существуют несколько параллельно протекающих процессов. Требования ко времени выполнения у различных процессов различны. Большинство процессов периодически повторяются (но с обновленными данными). Поскольку требования к времени выполнения различных процессов различны, а для их выполнения используются общие ресурсы системы, необходимо четкое упорядочение выполнения процессов. Для этого установлены абсолютные приоритеты. Существенной особенностью работы цифровых регуляторов в системах управления, как на базе ВМ, так и при микропроцессорной реализации, является то, что все операции, связанные с управляемым процессом, должны выполняться в реальном масштабе времени при постоянном информационном взаимодействии регулятора с объектом управления.

В основу организации управления заложена модель реально протекающего дискретного процесса. Процесс в модели представляется в дискретном времени. Реальное непрерывное время делится на кванты т, длительность кванта должна быть выбрана соответствующей инерционности объекта. Последовательность квантов, отражающая течение времени, нумеруется: т = 0, 1, 2, ... Для управления объектом при работе с моделью минимально необходимыми являются 3 компонента: идентификация текущего состояния модели $q(i)$ на основе показаний датчиков; расчеты, связанные с определением следующего состоя-

ния $q(i + 1)$; формирование управляющих воздействий на исполнительные устройства, обеспечивающих переход в состояние $q(i + 1)$.

Для реализации подсчета времени в сек. и мин. (реального времени) в цифровых регуляторах используют таймер (на рис. 3.3 не показан). По сигналу от таймера, происходит прерывание программы, выполняемой в фоновом режиме, и осуществляется переход к программам идентификации состояния объекта, расчета следующего состояния, формирования управляющих воздействий и вывода их на объект. Выполнение этих программ должно завершиться раньше, чем закончится квант. В оставшуюся часть кванта времени выполняются программы, имеющие более низкий приоритет. Процессы периодически повторяются. Привязка инициализации указанных процессов к сигналам от таймера и обеспечивает работу системы управления объектом в реальном времени.

Для управления сложными распределенными в пространстве объектами организуют аппаратно-программные управляющие комплексы с иерархической (двух- или трехуровневой) структурой. На верхнем уровне находятся ВМ в промышленном исполнении, выполняющие функции пультов управления. На нижних уровнях находятся ВМ, непосредственно осуществляющие сбор информации от датчиков, ее обработку и формирование управляющих воздействий. ВМ нижнего уровня содержат УСО, связанные через систему кабельных соединений с датчиками и исполнительными устройствами. Обмен информацией в этой части системы осуществляется с помощью цифровых и аналоговых сигналов. Для передачи отдельных сигналов требуются отдельные жилы кабеля. При такой организации информационного взаимодействия с объектом, особенно с объектами большой протяженности возникает ряд проблем. Эти проблемы связаны с обеспечением помехоустойчивости кабельных каналов связи, относительно высокой стоимостью кабельных соединений и их громоздкостью. Для преодоления этих трудностей часто используют встраиваемые системы на базе микроконтроллеров (МК). С их помощью осуществляется первичная обработка сигналов в непосредственной близости от объекта, и реализуются алгоритмы локального управления без выхода на ВМ. При этом углубляется иерар-



хия организации системы управления и перераспределяются функции между уровнями. Обмен информацией между МК и ВМ более высокого уровня осуществляется с использованием передачи сообщений, а не отельных сигналов и, как правило, выполняется с использованием стандартных каналов связи, благодаря чему существенно упрощается система кабельных соединений.

*Примечание*. Структура предложенная на рис. 3.3 может быть использована для любых объектов: от управления микроволновкой и телевизора, до сложного технологического процесса производства тепловой и электрической энергии.

Рассмотрим понятие микроконтроллера более подробно.

Организация микроконтроллерных систем

Микропроцессорные средства появились как результат развития технологии производства СБИС. Использование этих средств позволило существенно расширить сферы применения вычислительной техники. В частности они оказали влияние на развитие таких важных отраслей промышленности, как приборостроительная, радиотехническая, электронная, машиностроительная и др. Уже первые внедрения МП определили два основных направления их применения: использование МП в качестве центральных процессоров малых (микро и мини) ВМ и реализацию на базе МП встраиваемых систем управления различными объектами.

Особенности архитектуры и тенденции развития современных МП для ВМ рассмотрены ранее. Здесь рассматриваются принципы построения, организация и основные характеристики средств вычислительной техники для встраиваемых систем управления объектами самой различной природы. Здесь и далее под *встраиваемой системой управления (контроллером)* будем понимать систему управления, пространственно приближенную к датчикам и исполнительным устройствам и конструктивно интегрированную в оборудование – механизм, технологическую установку, робот и пр. (*централизованная система*). Такая система управления может быть реализована на основе одноплатной ВМ с необходимым набором интерфейсов, обеспечивающих функции связи с объ-

ектом и взаимодействие с оператором. Важным компонентом такой системы является интерфейс с системой управления более высокого уровня. Благодаря его наличию возможно решение задач комплексной автоматизации с использованием заданного числа единиц технологического оборудования, объединенного в единую *распределенную систему*.

Микропроцессорные реализации цифровых регуляторов имеют ряд специфических характеристик. В первую очередь, это необходимость разработки для каждой решаемой задачи собственного ПО. Действительно, из-за многообразия задач управления и ограниченности ресурсов микропроцессорных систем при решении конкретных задач использование стандартного ПО, в отличие от ВМ общего пользования, практически исключается. Кроме того, «уникальность» ПО любой задачи, не изменяемого после его отладки в течение всего времени эксплуатации системы, позволяет использовать для хранения ПО энергонезависимую память. К отличительным характеристикам микропроцессорных реализаций цифровых регуляторов можно отнести:

- относительно невысокие требования к вычислительным ресурсам (в большинстве случаев требуется реализация сравнительно простых алгоритмов управления);

- многообразие источников и приемников обрабатываемой информации (входные и выходные сигналы для цифровых регуляторов могут быть как дискретными, так и аналоговыми и для их обработки в составе микропроцессорной системы часто предусматривают устройства, преобразующие аналоговые си талы в цифровую форму и обратно);

- работа микропроцессорных систем управления в «реальном времени» (необходимо обеспечивать совершенно конкретную, рассчитываемую на этапе разработки скорость реакции системы управления на внешние события, потому что задержки, превышающие расчетные, в таких системах недопустимы, так как могут приводить к потере управляемости и, как следствие, к повреждению или разрушению объекта управления, что потребовало включения в состав



микропроцессорных систем управления таких устройств, как таймеры, контроллеры прерываний и др.);

- повышенные требования к надежности и стойкости к воздействиям окружающей среды (встраиваемые системы управления должны работать без сбоя и ошибок в необслуживаемом режиме и, как правило, в сложных условиях эксплуатации).

По мере совершенствования цифровых технологий у разработчиков СБИС появилась возможность разместить основные подсистемы цифровых регуляторов (процессор, память, периферийные устройства) на одном кристалле. Первая реализация подобной системы, получившей название однокристальной ВМ или МК, была представлена фирмой Intel в 1978 г. в виде устройства 8048.

*Микроконтроллером* называется программируемое однокристальное вычислительное устройство с встроенным набором средств для ввода и вывода, применяемое для решения задач управления и первичной обработки данных в технических системах.

Встраиваемые средства на базе МК относительно просты, имеют низкую стоимость, малые размеры. На них обычно возлагают следующие функции:

- сбор, первичную обработку и преобразование сигналов датчиков объекта в сообщения стандартного формата, передачу их в ВМ;

- прием, накопление и трансляцию команд управления, передаваемых от ВМ;

- самодиагностику и первичную диагностику объекта.

В зависимости от сложности задач встраиваемые системы управления бывают одноуровневыми и многоуровневыми. В последнем случае на нижнем уровне управления решаются задачи локального управления отдельными подсистемами, а на более высоких – общие задачи управления системой в целом, мониторинг и управление параметрами с использованием SCADA систем. Одноуровневые системы управления, как правило, реализуют на базе МК, а при построении двух- и более уровневых систем на верхних уровнях управления предполагается использование ВМ в промышленном исполнении, обеспечива-

ющих интерфейс с человеком-оператором. Существуют микроконтроллерные системы следующих типов – автономные (без ВМ), локальные (автономные с ВМ) и сетевой конфигурации (рис. 3.4).

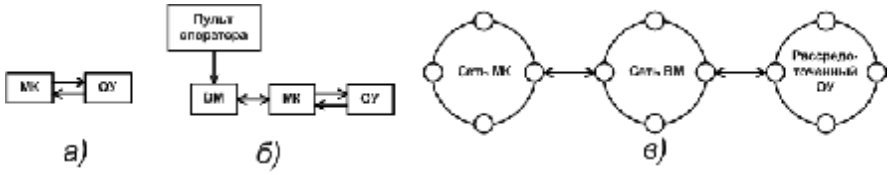



*Рис. 3.4.* Типы микроконтроллерных систем: *а* – автономная; *б* – локальная; *в* – сетевой конфигурации; МК – микроконтроллер; ОУ – объект управления; ВМ – вычислительная машина

Автономные системы управления являются самыми простыми и дешевыми, но их возможности ограничены. Более сложные локальные системы обладают широкими возможностями при управлении объектами различной природы и сложности. Наиболее сложными системами управления являются системы сетевой конфигурации с распределенным многоуровневым управлением и распределенным (рассредоточенным) объектом управления.

Типовая структура микроконтроллера. Общие сведения.

Существует большое разнообразие МК. По общим архитектурным принципам их объединяют в семейства. Все МК семейства имеют одинаковое ядро, характеризующееся совокупностью таких понятий, как система команд, организация памяти программ и памяти данных, базовый набор внутренних периферийных устройств и система прерываний. Например, существуют сотни модификаций ядра семейства MCS-51, разработанного фирмой Intel. Наиболее важной особенностью семейства является программная совместимость на уровне двоичных кодов всех входящих в него МК. Это позволяет разработчикам систем заменять одни МК другими без потери наработок своего ПО.



Отличия между отдельными представителями одного семейства в основном заключаются в составе периферийных устройств и объеме памяти программ и данных. Сегодня в мире выпускаются тысячи типов МК. В группе лидеров такие компании, как Atmel, Intel, Motorola, Texas Instruments, Toshiba, Microchip Technology Inc., Zilog, Mitsubishi Electronics, Hitachi Semiconductor, Philips Semiconductors, Siemens и др.

МК (рис. 3.5) содержит в своем составе процессорное ядро с развитой системой команд, память программ IROM, регистровый файл данных RRAM, а также набор программируемых интерфейсных схем, выполняющих ряд важных функций внутри МК и обеспечивающих связь с внешней средой. Большинство из перечисленных блоков являются обязательными компонентами любого МК, некоторые, например IROM, в структуре конкретного МК могут отсутствовать.

Состав и назначение интегрированных на кристалл периферийных устройств определяется областью преимущественного применения МК и может широко варьироваться в зависимости от типа МК. С целью повышения гибкости использования аппаратных средств, интегрированных на кристалле, и внешних выводов кристалла эти средства делаются многофункциональными с программной настройкой на тот или иной режим работы.

При инициализации МК информация о типе настройки заносится в специальные регистры и в процессе дальнейшей работы МК обычно остается неизменной. Это позволяет упростить оперативное управление встроенными средствами МК и расширить возможности их использования. В большинстве случаев в число устройств встроенной периферии входят таймеры (счетчики), последовательные и параллельные порты ввода-вывода, контроллер прерывания, многоканальный аналого-цифровой преобразователь (АЦП), сторожевой таймер и пр.

Успех применения МК способствовал развитию и усложнению их структур. Достижения микроэлектроники позволили не только усложнить структуру систем реализованных на кристалле, но и существенно повысить их быстродействие. Сейчас микроконтроллеры на ARM-ядре по производительности превышают некоторые ПК.

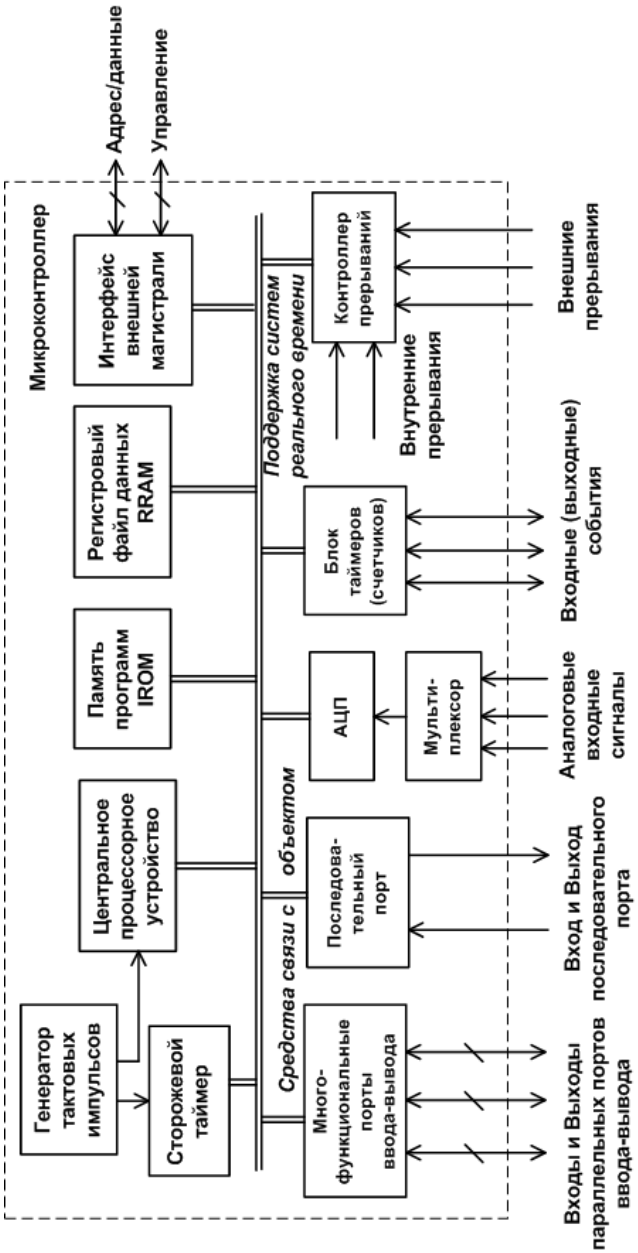



*Рис. 3.5.* Структура микроконтроллера на примере MCS-51



*Параллельная обработка данных как архитектурный способ повышения производительности*

*Производительность* – важнейший показатель ВС. Увеличение требований к производительности постоянно опережает возможности вычислительных систем. Поэтому задачи повышения производительности находятся в центре внимания исследователей и разработчиков средств и систем вычислительной техники.

Имеются три метода, обеспечивающих увеличение производительности систем: совершенствование элементной базы, структурные и математические методы. *Совершенствование элементной базы* позволяет увеличивать тактовую частоту работы устройств и систем, интегрировать в одном кристалле процессор, память и часть устройств ввода-вывода, увеличивать разрядность шин, памяти и операционных устройств, использовать внутри кристалла более совершенные структуры устройств. *Структурные методы* направлены на использование методов параллельной обработки данных, что требует такой организации ВС, когда параллельно (одновременно) выполняется большое число преобразований. *Математические методы* связаны с созданием новых вычислительных методов решения классов задач, допускающих распараллеливание вычислительных процессов. Для реализации потенциально достижимого уровня производительности требуются согласованные усилия математиков и разработчиков аппаратного обеспечения, системного ПО, интегральных микросхем.

Параллельная обработка может быть организована на нескольких уровнях организации ВС, но в первую очередь она связана с применением архитектур с несколькими процессорами. Можно отметить четыре основных преимущества параллельных архитектур:

1. Применение мультипроцессорных архитектур – это единственный способ построения наиболее высокопроизводительных ВС. Построив даже самый мощный процессор для данного уровня развития вычислительной техники, можно еще более увеличить производительность, объединив в единую систему несколько процессоров.

2. Отношение стоимости к производительности при требовании высокой производительности у мультипроцессорных систем ниже, чем у однопроцессорных. Очевидно, что повышение производительности в рамках одной архитектуры, как для процессора, так и для ВС, построенной на его основе, связано с повышением стоимости. Функциональную зависимость между этими показателями устанавливает так называемый закон Гроша (справедлив только в определенных пределах):

$$V=1/T=kC^2$$

где $V$ – производительность; $T$ – время решения задачи; $k$ – константа зависящая от архитектуры, элементной базы, класса задач; $C$ – стоимость.

В целом, как для однопроцессорных систем так и для многопроцессорных с момента когда возможности архитектуры исчерпаны повышение производительности за счет вложения средств наращивание вычислительной возможностей малоэффективно. Для начального этапа (малой производительности) – однопроцессорные системы ниже по стоимости; для следующего этапа – стоимости примерно равны; на последнем этапе наращивание производительности многопроцессорных систем более дешево.

3. Мультипроцессорная система в связи с модульной организацией имеет более широкие возможности для изменения производительности, что позволяет обеспечить широкий диапазон требований в рамках одной архитектуры. Это свойство систем называют *масштабируемость*.

4. Мультипроцессорная система позволяет обеспечить большую отказоустойчивость за счет использования структурной избыточности.

Несмотря на отмеченные преимущества мультипроцессорных систем, их распространение сопряжено с определенными трудностями, свойственными всем распределенным системам параллельной обработки данных. Рассмотрим *основные трудности*.

1. Для параллельных систем характерны новые, не известные ранее типы ошибок:

- дедлоки (взаимные блокировки процессов);



- ливлоки (невыполнение некоторых альтернатив);

- голодание (невозможность получить), ресурс;

- несправедливость при планировании ресурса.

2. Сложности понимания и анализа параллельных процессов. Мышление людей имеет последовательный характер, так как мысли выражаются словами.

3. Недостаточная разработанность теоретических моделей. Теория параллелизма развивается в нескольких направлениях:

- создание вычислительных моделей применительно к задачам синтеза и анализа (сети Петри, системы переходов, реактивные системы и др.);

- разработка языков спецификации требований (предикаты, темпоральная логика и др.);

- разработка методов верификации (анализ достижимости, логический вывод и др.).

4. Недостаточная разработанность технологии параллельного программирования. При этом актуально решение нескольких задач. Необходимо создание ОС, языков программирования, компиляторов и прикладных программ, наиболее полно использующих возможности параллельных архитектур. Желательна совместимость и стыковка этих параллельных программных средств с уже существующими средствами. Одна из актуальных задач – синхронизация и управление параллельными процессами.

В настоящее время актуально увеличивать производительность мощных ВС, которые являются общим инструментом для многих пользователей. В тех случаях когда нет необходимости выполнять несколько программ на одной и той же ВМ, производительность можно повысить, выполняя их на отдельных машинах. Повышение производительности в пределах одной ВС имеет смысл, если создаются условия для понижения стоимости обработки за счет совместного использования аппаратных и программных ресурсов: обеспечение доступа к совместно используемой информации из каждой программы, обеспечение обмена данными между программами и т.п.

Существуют два способа организации параллельной обработки данных. *Многоэлементная обработка* осуществляется на нескольких параллельно работающих операционных устройствах (элементах структуры системы). Каждый элемент выполняет свою работу, осуществляя обработку соответствующей порции данных от начала до конца. Использование такого способа параллельной обработки в современных процессорах связано с понятием суперскалярной архитектуры процессора. *При многофазной (многостадийной) обработке* процесс обработки данных разбивается на несколько стадий (фаз), выполняемых последовательно. Между фазами имеются буферы для хранения промежуточных результатов. После выполнения 1-й фазы результат запоминается в буфере и начинается обработка 2-й фазы. Средства выполнения 1-й фазы освобождаются и на их вход поступает следующая порция данных. Таким образом, параллельно решается несколько задач. Этот способ соответствует конвейерной обработке.

Параллельно выполняемые единицы обработки могут быть большими (программы) и маленькими (команды, микрооперации). Разделение обработки на единицы различной величины, соответствующие многоуровневой организации процессов, называют гранулярностью обработки. Способы распараллеливания с учетом гранулярности обработки положены в основу классификации систем параллельной обработки.

Классификация систем параллельной обработки

К настоящему времени разработано большое число различных архитектур систем параллельной обработки данных. Существуют различные подходы к их классификации. Одной из первых появилась классификация, предложенная М.Флинном в 1966 г.

Первый и второй класс ОКОД и ОКМД были рассмотрены выше. Здесь же остановимся именно на системах, в которых используется параллельная обработка, а именно системах где есть множественный поток команд.



МКОД (множественный поток команд, одиночный поток данных) или MISD (Multiple Instruction stream/ Single Datastream). К нему относят конвейерные системы, в которых данные обрабатываются за несколько стадий. На отдельных стадиях обработка выполняется различными операционными устройствами под управлением соответствующих команд.

МКМД (множественный поток команд, множественный поток данных) или MIMD (Multiple Instruction stream/ Multiple Data stream). К этому классу относятся мультипроцессорные системы, в которых отдельные порции данных обрабатываются различными операционными устройствами, каждое из которых выполняет свою программу.

Приведенный подход Флина порождает неоднозначность классификации существующих архитектур. Например, поток векторных данных можно рассматривать и как одиночный и как множественный. Кроме того, приведенными классами неоднозначно отражается конвейеризация, которая применяется на нескольких уровнях организации вычислительных процессов. Недостатком является и большое многообразие систем в классе MIMD. Поэтому были предложены другие подходы к классификации Р.Хокни, Т. Фенгом и др.

MISD компьютеры

*MISD* (Multiple Instruction Stream - Single Data Stream) или МКОД - множество потоков команд и один поток данных. MISD компьютеры представляет собой, как правило, регулярную структуру в виде цепочки последовательно соединенных процессоров П1, П2, ..., ПN, образующих процессорный конвейер (рис. 3.6).

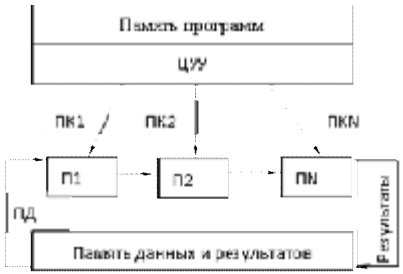



*Рис. 3.6.* MISD-архитектура

В такой системе реализуется принцип конвейерной (магистральной) обработки, который основан на разбиении всего процесса на последовательно выполняемые этапы, причем каждый этап выполняется на отдельном процессоре. Одинарный поток *исходных данных* для решения задачи поступает на вход процессорного конвейера. Каждый процессор решает свою часть задачи, и результаты решения в качестве исходных данных передает на вход последующего процессора. К каждому процессору подводится свой поток команд, т. е. наблюдается множественный поток команд *ПК*1, *ПК*2, ..., *ПК*N.

Вычислительных машин такого класса практически нет и трудно привести пример их успешной реализации. Один из немногих - систолический массив процессоров, в котором процессоры находятся в узлах регулярной решетки, роль ребер которой играют межпроцессорные соединения. Все процессорные элементы управляются общим тактовым генератором. В каждом цикле работы каждый процессорный элемент получает данные от своих соседей, выполняет одну команду и передает результат соседям.

В дальнейшем для *MISD* нашлась еще одна адекватная организация вычислительной системы - распределенная мультипроцессорная система с общими данными. Наиболее простая и самая распространенная система этого класса - обычная локальная сеть персональных компьютеров, работающая с единой базой данных, когда много процессоров обрабатывают один поток данных. Впрочем, тут есть одна тонкость. Как только в такой сети все пользователи переключаются на обработку собственных данных, недоступных для других абонентов сети, MISD - система превращается в систему с множеством потоков команд и множеством потоков данных, соответствующую MIMD-архитектуре.

MIMD компьютеры

*MIMD* (Multiple Instruction Stream - Multiple Data Stream) или МКМД - множество потоков команд и множество потоков данных. Эта категория архитектур вычислительных машин наиболее богата, если иметь в виду примеры ее успешных реализаций. В нее попадают симметричные параллельные вычислительные системы, рабочие станции с несколькими процессорами, кластеры ра-



бочих станций и т.д. Уже довольно давно появились компьютеры с несколькими независимыми процессорами, но вначале на таких компьютерах был реализован только параллелизм заданий, то есть на разных процессорах одновременно выполнялись разные и независимые программы.

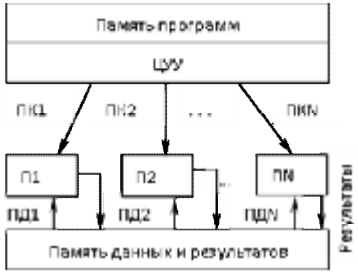



*Рис. 22.3.* MIMD-архитектура

Так как только MIMD-архитектура включает все уровни параллелизма от конвейера операций до независимых заданий и программ, то любая вычислительная система этого класса в частных приложениях может выступать как SISD и SIMD-система. Например, если многопроцессорный комплекс выполняет одну-единственную программу без каких-либо признаков векторного параллелизма данных, то в этом конкретном случае он функционирует как обычный SISD-компьютер, и весь его потенциал остается невостребованным. Таким образом, употребляя термин «MIMD», надо иметь в виду не только много процессоров, но и множество вычислительных процессов, одновременно выполняемых в системе. MIMD-системы по способу взаимодействия процессоров (рис. 22.3.) делятся на системы с сильной и слабой связью.

Многопроцессорные вычислительные системы (сильно связанные)

Сильно-связанные вычислительные системы или многопроцессорные вычислительные системы (МПВС) основаны на объединении процессоров на общем поле оперативной памяти. Управление обеспечивается одной общей операционной системой. При этом достигаются более быстрый обмен информацией между процессорами, чем между ЭВМ в многомашинных вычислительных

системах (комплексах), и более высокая суммарная производительность системы. Иногда их называют «истинными» мультипроцессорами.

В таких системах, как правило, число параллельных процессов невелико и управляет ими централизованная операционная система. Процессы обмениваются информацией через общую оперативную память. При этом возникают задержки из-за межпроцессорных конфликтов. При создании больших мультипроцессорных ЭВМ (мэйнфреймов, суперЭВМ) предпринимаются огромные усилия по увеличению пропускной способности оперативной памяти. В результате аппаратные затраты возрастают чуть ли не в квадратичной зависимости, а производительность системы упорно «не желает» увеличиваться пропорционально числу процессоров. Так, сложнейшие средства снижения межпроцессорных конфликтов в оперативной памяти суперкомпьютеров серии CRAY X-MP/Y-MP позволяют получить коэффициент ускорения не более 3,5 для четырехпроцессорной конфигурации системы.

По топологии межмодульных функциональных и управляющих связей и организации работы с общей памятью выделяются два типа многопроцессорных систем МКМД: с общей шиной; с использованием многовходовой памяти.

Сильно связанные системы с общей шиной. В МПВС с общей шиной все функциональные модули (процессоры П1, П2, ..., ПМ, модули памяти МП1, МП2, .... МПК, устройства ввода-вывода УВВ1, УВВ2, .... УВВМ) подсоединены к одной общей шине межмодульных связей, ширина которой может быть от одного бита до нескольких байтов. Между модулями системы нет постоянных связей, информация между ними передается в режиме разделения времени. В каждый данный момент времени по шине передается лишь один пакет информации от какого-то одного источника. Другие источники информации должны ожидать, пока не освободится системная шина.

Такие системы получили название SMP – симметричные мультипроцессорные системы (англ. Symmetric Multiprocessing, сокращенно SMP) т.к. все процессоры имеют равные права и одинаковую систему адресации.

Основные *преимущества системы* с общей шиной:



- система характеризуется относительно низкой функциональной сложностью и невысокой стоимостью;

- в системе легко осуществляется реконфигурация структуры путем добавления или удаления функциональных модулей.

*Недостатки таких систем*:

- ограничение производительности системы пропускной способностью общей шины;

- ухудшение общей производительности системы при ее расширении путем добавления модулей;

- потери производительности системы, связанные с разрешением конфликтов, которые возникают в случае, когда несколько модулей одновременно претендуют на занятие общей шины для передачи информации; эти потери можно снизить, если, кроме общей памяти, доступной всем процессорам, каждый из них снабдить местной (локальной, процессорной) памятью для хранения информации, участвующей в ближайшем ряду вычислений; при наличии местной памяти частота обращения процессоров к общей памяти уменьшается, следовательно, уменьшается частота возникновения конфликтов и потери времени на их разрешение;

- отказ общей шины приводит к выходу из строя всей системы.

Организация связей между элементами системы на основе общей шины является одним из распространенных способов построения не только многопроцессорных, но и многомашинных вычислительных комплексов небольшой мощности.

Архитектура SMP стала своего рода стандартом для многопроцессорных серверов (например, HP9000 и DEC Alpha Server AXP).

Сильно связанные системы с многовходовыми модулями ОП. В МПВС с многовходовыми модулями ОП взаимные соединения выполняются с помощью индивидуальных шин, подключающих каждый процессор и каждое устройство ввода-вывода к отдельному входу оперативной памяти. Для этого необходимо, чтобы модули ОП имели по несколько входов и снабжались управляющими

схемами для разрешения конфликтов в случаях, когда два или более процессора или устройства ввода-вывода требуют доступа к одному и тому же модулю памяти в пределах одного временного цикла. Число подключаемых элементов системы к одному модулю памяти ограничивается числом его входов.

При построении общего поля ОП МПВС целесообразной оказывается реализация метода расслоения оперативной памяти, при которой ячейки с соседними адресами оказываются расположенными в соседних модулях. Обязательными условиями применения этого метода являются модульность структуры ОП и наличие для каждого модуля памяти автономного блока управления памятью.

*Преимущества МПВС* с многовходовыми модулями ОП:

- скорость передачи информации значительно выше, чем в МПВС с общей шиной;

- система может работать и в режиме однопроцессорной конфигурации.

*Недостатки* таких систем следующие:

- большое число линий связи и разъемов, усложняющих конструкцию системы и снижающих ее надежность;

- оперативная память, составленная из многовходовых модулей, является дорогостоящей.

Принципы построения МПВС с многовходовыми модулями ОП используются в мэйнфреймах.

Многомашинные вычислительные системы ММВС (слабосвязанные)

В системах типа МКМД реализуется асинхронный вычислительный процесс, при котором каждый процессор системы выполняет свою программу (или свой участок сложной программы) с собственными данными. В таких системах происходит постоянное распараллеливание вычислений. Две основных причины создания этого типа ВС – дублирование важных блоков вычислений или модулей ВС и повышение производительности систем.



Слабосвязанные системы: многомашинные комплексы. Вычислительные системы со слабой связью или распределенные вычислительные системы, как правило, представляются *многомашинными комплексами*, в которых отдельные компьютеры объединяются либо с помощью сетевых средств, либо с помощью общей внешней памяти (обычно – дисковые накопители большой емкости). Каждая ЭВМ системы имеет свою оперативную память и работает под управлением своей операционной системы. Каждая машина использует другую как канал или устройство ввода-вывода. Обмен информацией между машинами происходит в результате взаимодействия их операционных систем.

ММВС строится из логически независимых компонентов: процессоров, устройств оперативной памяти, каналов ввода-вывода, ВЗУ, устройств управления ВЗУ, устройств ввода-вывода, устройств управления УВВ. Логическая независимость процессоров системы определяется возможностью их независимого функционирования. Для остальных компонентов эта независимость определяется возможностью их подсоединения к одному или к нескольким процессорам ММВС.

Связь между машинами (процессорами) ММВС может осуществляться на уровне любого из его логически независимых компонентов с помощью специальных мультисистемных средств или средств комплексирования. Такая связь должна быть достаточно гибкой и обеспечивать независимость функционирования различных модулей системы и их взаимодействие с различной скоростью, соответствующей скорости обмена информацией между элементами системы. Для этого на разных уровнях комплексирования применяются различные по тактовой частоте, разрядности, пропускной способности шины интерфейсов.

Очевидно, что система со всеми возможными уровнями связей будет наиболее совершенной, гибкой и надежной в функционировании. Но с другой стороны, система с полными связями получается сложной по своей структуре и организации функционирования. В каждом конкретном многомашинном комплексе не обязательно реализуются все уровни комплексирования.

Рассмотрим уровни связи вычислительной системы в порядке возрастания скорости обмена информацией.

*Межмашинная связь на уровне внешних устройств* используется главным образом для организации общего поля внешней памяти. Такая связь организуется через каналы ввода-вывода этих устройств и шинные интерфейсы. При этом обычно часть ВЗУ остается в индивидуальном пользовании отдельных машин. Преимуществом комплексирования на уровне ВЗУ является значительное увеличение объемов информации (данных и программ), одновременно доступных процессорам ММВС.

Наибольшее распространение получила *взаимодействие вычислительных средств на уровне канал-канал* через адаптер канал-канал. Адаптер подключается к двум каналам, причем функционально он рассматривается как устройство управления ввода-вывода для каждого из каналов, а каждая из связанных адаптером машин по отношению друг к другу является внешним устройством. В отличие от любого другого устройства управления внешними устройствами адаптер не управляет устройствами ввода-вывода, а только осуществляет связь между каналами и синхронизирует их работу. Адаптер обеспечивает быстрый обмен информацией между каналами, а следовательно, и между ОП взаимодействующих процессоров, если общее поле оперативной памяти не организовано.

*Взаимодействие на уровне ОП* осуществляется для создания общего поля оперативной памяти, что значительно ускоряет обмен информацией между процессорами и повышает возможности функционирования системы. Наибольшая оперативность обмена информацией достигается при реализации именно такого уровня связи. Построение ММВС с общим полем ОП связано с необходимостью применения многовходовых модулей памяти. При этом существенно усложняются структура и функции устройства управления общей оперативной памятью. С помощью УУ должны решаться такие задачи, как реализация установленной очередности обращения к модулям памяти со стороны процессоров, решение конфликтных ситуаций, возникающих при одновременном обращении к одному и тому же модулю памяти со стороны обоих процессоров, поддержание когерентности памяти. Наибольшие затруднения связаны с созданием программных средств, обеспечивающих функционирование системы с общим полем ОП.



*Взаимодействие на уровне процессоров* через интерфейс межпроцессорной связи осуществляется с целью синхронизации единого вычислительного процесса путем передачи между процессорами сигналов внешних прерываний и команд прямого управления.

Для оценки эффективности взаимодействия вычислительных средств системы на различных уровнях могут привлекаться такие показатели эффективности ВС, как время реакции системы на запросы с учетом их приоритетов, пропускная способность, время на решение заданного набора задач.

Слабосвязанные системы: системы массового параллелизма – ММР архитектура. Другим примером реализации слабосвязанной архитектуры являются *системы с массовым параллелизмом* (*MPP*), состоящие из десятков, сотен, а иногда и тысяч процессорных узлов. Строгой границы не существует, однако считается, что при числе процессоров 128 и более система относится к MPP-архитектуре. Большинство MPP-систем имеют как логически так и физически распределенную между процессорами память. Каждый узел такой системы содержит процессор и модуль памяти, в котором хранится процесс - совокупность команд, исходных и промежуточных данных вычислений, а также системные идентификаторы процесса. Узлы массово-параллельной системы объединяются коммутационными сетями самой различной формы - от простейшей двумерной решетки до гиперкуба или трехмерного тора. В отличие от архитектуры фон Неймана, передача данных между узлами коммутационной сети происходит по готовности данных процесса, а не под управлением некоторой программы. Отсюда еще одно название подобных систем - «системы с управлением потоком данных» (иногда просто «потоковые машины»).

*К достоинствам* данной архитектуры относится то, что она использует стандартные микропроцессоры и обладает неограниченным быстродействием (порядка TFLOPS).

*Однако есть и недостатки* – программирование коммутаций процессов является слабо автоматизированной и очень сложной процедурой. Так что для коммерческих задач и даже для подавляющего большинства инженерных приложений системы с массовым параллелизмом недоступны.

*Замечания.*

1. Сравнивая между собой ММВС и МПВС, можно отметить, что в МПВС достигается более высокая скорость обмена информацией между элементами системы и поэтому более высокая производительность, более высокая реакция на возникающие в системе и ее внешней среде нестандартные ситуации, более высокие надежность и живучесть (МПВС сохраняет работоспособность, пока работоспособны хотя бы по одному модулю каждого типа устройств). С другой стороны, построение ММВС из серийно выпускаемых ЭВМ с их стандартными операционными системами значительно проще, чем построение МПВС, требующих преодоления определенных трудностей, связанных главным образом с организацией общего поля оперативной памяти и созданием единой операционной системы. Разница организации MIMD-систем с сильной и слабой связью проявляются еще и при обработке приложений, отличающихся интенсивностью обменов между взаимодействующими процессами.

2. В любом случая относительно классификации МПВС, ММВС существует промежуточное объединение и разбиение этого класса по использованию ОП – с общей памятью и распределенной памятью.

3. Вычислительные сети являются дальнейшим развитием вычислительных систем распределенного типа. Они представляют собой новый, более совершенный этап в использовании средств вычислительной техники - переход к коллективному их использованию.

4. В настоящее время многоядерные системы являются дальнейшим развитием подхода по использованию многопроцессорных конфигураций. Современные параллельной обработки данных, например, кластеры, строятся уже на с применением многоядерных профессоров, по приведенным выше правилам.



### *3. Вычислительные системы – состояние, производительность, направления развития.*

В области высокопроизводительных систем наблюдается устойчивая тенденция снижения показателя стоимость (производительность). Наиболее мощными ВС являются суперкомпьютеры. Их архитектура, основанная на организации параллельной работы векторных процессоров, специализирована для эффективного решения определенных классов задач. Однако продукция основных компаний-поставщиков суперкомпьютеров, таких как, CRAY Research, NEC, SUN, Hewlett Packard и других остается непозволительно высокой для большинства российских предприятий и организаций. Кроме того, во многих странах действует ряд законов и положений, препятствующих экспорту в Россию изделий, относящихся к классу высокотехнологичных.

С другой стороны, все большую популярность приобретают системы, собранные из серийно производимых комплектующих. По производительности такие системы с большим числом процессоров уже не уступают суперкомпьютерам, а по стоимости и надежности часто лучше их. С учетом этого применительно к системам для высокопроизводительных вычислений изменяется и архитектура МП, а для организации информационного взаимодействия все чаще используются стандартные каналы связи и средства телекоммуникаций.

Развитие архитектуры ВС зависит от большого числа факторов. Существенными факторами являются: области применения и соответствующие требования к системам; компании-производители ВС; развитие технологии производства ИС и элементной базы; результаты научных исследований в области ВС и информационных технологий. Все эти факторы динамично изменяются, влияя на спектр архитектур используемых ВС. Для характеристики развития ВС периодически публикуются обзоры 500 наиболее мощных высокопроизводительных систем.

Системы, способные осуществлять как быстрый счет, так и быстрый обмен данными, требуются для точного прогнозирования погоды и климата, для развития современного промышленного дизайна, для создания новых фарма-

цевтических средств, для проведения научных исследований в различных областях знаний и для соблюдения других научных и прикладных интересов. Требуются такие системы и самим производителям ВС для выполнения соответствующих прикладных исследований и автоматизации проектирования. Области применения высокопроизводительных систем условно подразделяют на пять категорий: исследовательскую, промышленную, академическую, административную и область производителей.

На основе анализа ряда обзоров можно отметить две явные и устойчивые тенденции: наибольшее количество систем в промышленной области и наивысшая производительность – в исследовательской. Основными фирмами – производителями высокопроизводительных ВС являются: IBM, SGI, SUN, HP, Fujitsu, NEC, Hitachi, Intel, Compaq.

*По данным на 2007 год.*

Наибольшее число высокопроизводительных систем установлено в США – 55 %, в Европе – 31 %, в Японии – 11 %. В Европе наибольшее число систем установлено в Германии – 43 %.

Рассмотрим распределение высокопроизводительных систем но архитектурам. Системы MPP составляют 65%, системы SMP – 21 %, системы PVP (с векторными процессорами) – 12%, кластеры – 2%. Анализ динамики развития ВС показывает, что доля кластерных систем возрастает.

Распределение архитектур по суммарной производительности следующее: системы MPP – 77%, системы PVP – 14%, системы SMP – 8 %, кластеры – 1 %. Направление ВС с массовым параллелизмом MPP лидирует как по количеству, так и по суммарной производительности установленных систем.

*Данные на 2013 год.*

Информация представлена на презентациях. Отчетливо видно, что кластерные системы вышли на первое место по распространенности. При этом на второе место выходит Китай.



*Направления развития высокопроизводительных вычислительных систем*

Как было отмечено выше параллельная обработка это единственный способ увеличить производительность систем. Поэтому очевидно, что такие системы постоянно развивается.

Тенденции развития архитектур систем с общей памятью

Наиболее актуальный вопрос для систем SMP – организация быстрого доступа процессоров к ОП. Для уменьшения затрат времени на обмен данными с учетом свойства локальности программ и данных процессоры имеют кэш-память. Поскольку процессоры одновременно работают с данными, хранящимися в единой памяти, в архитектуре систем SMP обязательно должен быть механизм поддержки когерентности данных (в любой момент времени для каждого элемента данных во всей памяти системы SMP существует только одно его значение, хотя одновременно могут существовать несколько копий элемента данных, размещенных в разных видах памяти и обрабатываемых разными процессорами)

Для обеспечения высокого быстродействия в современных системах SMP когерентность реализуется аппаратными средствами. Типичные системы SMP для поддержки когерентности используют дополнительную шину слежения (snoopy bus). Для того, чтобы кэши всех процессоров были когерентными, каждый процессор «подглядывает» за шиной, осуществляя поиск тех операций считывания и записи между другими процессорами и ОП, которые влияют на содержимое собственных кэшей.

При увеличении числа процессоров узким местом становится шина, через которые процессоры соединяются с ОП. Для повышения производительности системы требуется увеличение пропускной способности шины.

Современное решение вышеуказанных проблем может быть проиллюстрировано на примере мультипроцессорной системы Power Scale, разработанной группой компаний Bull (рис. 2.4).

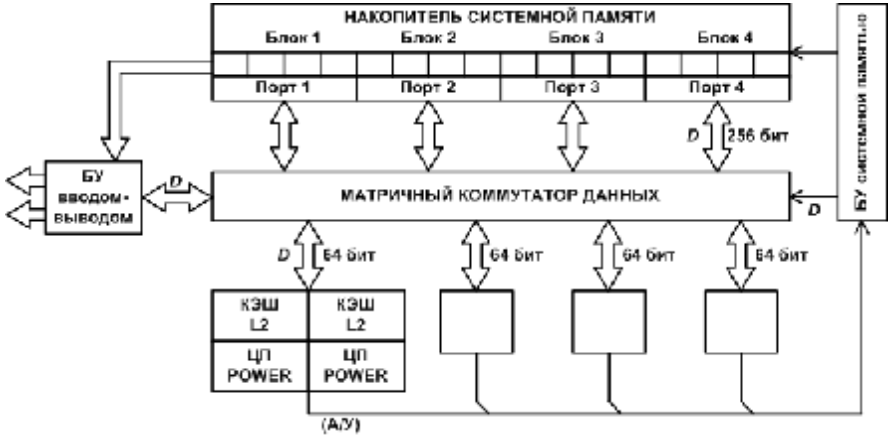



*Рис. 2.4.* Функциональная схема мультипроцессорной системы Power Scale (группа компаний Bull): БУ – блок управления; L2 – уровень кэш-памяти; ЦП – центральный процессор

Функциональная схема системы Power Scale содержит разделяемую ОП, включающую блок накопителя системной памяти и блок управления системной памятью; четыре процессорных блока, каждый из которых содержит два процессора Power PC с кэшами 2-го уровня. Основным средством доступа к разделяемой ОП является достаточно сложная «системная шина». Она включает шину адреса (управления) (А/У), реализованную классическим образом, и восемь наборов магистралей данных *D*, (четыре – от процессорных блоков и четыре – от блоков системной памяти), которые соединяются между собой посредством высокоскоростного матричного коммутатора данных. К коммутатору подключен также порт ввода-вывода, входящий в состав блока ввода-вывода. Используется также расслоение памяти (каждый из четырех блоков содержит по четыре слоя). Такая организация памяти позволяет обращаться к ней многим процессорам одновременно. Каждый процессорный блок имеет свой выделенный порт памяти для пересылки данных. При этом общая шина адреса и управления гарантирует, что на уровне системы все адреса являются когерентными. Управление информационным обменом организовано с учетом необходимости обеспечения когерентности кэшей и достижения максимальной производительности.



Тенденции развития архитектур систем с разделяемой памятью

В системах с симметричной мультипроцессорной архитектурой время обращения к общей памяти одинаково для всех процессоров. Отсюда происходит еще одно название для таких систем – UMA (Uniform Memory Architecture). Как уже отмечалось ранее, общая память и шина, связывающая процессоры с памятью становятся узким местом и сдерживают увеличение вычислительной мощности системы. Помимо рассмотренного подхода с коммутатором и «широкой» шиной данных (256 бит) получила развитие концепция построения распределенной разделяемой памяти – distributed shared memory (DSM). Это физически распределенная память с общим АП, доступ к которой осуществляется посредством команд «чтение» и «запись». Системы с распределенной разделяемой памятью, являющиеся развитием систем SMP, подразделяют на три класса:

- системы с архитектурой NUMA (Non-Uniform Memory Architecture);
- системы с архитектурой COMA (Cache-Only Memory Architecture);
- системы с рефлексивной памятью RM (Reflective Memory).

Выделенные классы отличаются организацией ОП и организацией информационного взаимодействия процессоров через разделяемую память.

*Рассмотрим принципиальные особенности архитектуры NUMA.* Целью разработчиков архитектуры NUMA было стремление объединить достоинства классов систем SMP (относительная простота разработки программ) и систем MPP (хорошая масштабируемость – возможность наращивания числа процессорных узлов в системе). Упрощению написания программ для систем с общей ОП способствует единое АП памяти для всех процессоров системы, что упрощает реализацию связей по данным между процессами. Создатели архитектуры NUMA в конце 70-х гг. XX в. предложили архитектуру, содержащую набор кластеров, соединенных межкластерной шиной. АП с использованием старших разрядов адреса делится между кластерами. Когда процессор обращается к памяти, он посылает адрес к своему контроллеру памяти. Резидентная память кластера связана с процессором через локальную шину. Контроллер анализирует старшие разряды адреса и по ним определяет, в каком модуле (кластере)

находится требуемая ячейка памяти. Если адрес локальный, то запрос выставляется на локальную шину, иначе – на межкластерную шину. Локальный запрос выполняется существенно быстрее, чем удаленный. Проблема развития этого подхода заключается и использовании каждым процессором кэш-памяти. Возникает упомянутая уже выше проблема когерентности кэшей – идентичности значений всех копий переменной в системе. Решение аппаратными средствами этой проблемы стало основой класса архитектур CC-NUMA (cache coherent NUMA).

У истоков создания архитектуры CC-NUMA стояла компания Sequent. По данным аналитиков данная архитектура позволяет на порядок повысить производительность и увеличить возможности масштабирования всех ресурсов системы. Распределенная разделяемая память в системах NUMA образуется как совокупность памятей вычислительных модулей, каждый из которых может включать несколько процессоров (1 – 4) с подсоединенным к ним блоком локальной памяти. Каждый такой модуль имеет архитектуру SMP. Распределение памяти по вычислительным модулям позволяет рационально использовать процессоры и блоки памяти и строить системы из стандартных модулей (например, использовать серийные системные платы ПК). Существенным архитектурным отличием CC-NUMA является использование аппаратуры для увязки механизма работы кэш-памяти вычислительного модуля с удаленными блоками памяти других вычислительных модулей.

При построении разделяемой памяти существенны следующие ее характеристики:

- аппаратный, программный либо аппаратно-программный способ реализации механизмов управления работой разделяемой памяти;

- размер минимальной разделяемой единицы данных: слово, кэш-строка, страница, сегмент, объект и т.д.;

- модель состоятельности памяти;

- механизм управления распределением памяти и размещением данных (централизованный или распределенный, статический или динамический).



При аппаратной реализации разделяемой памяти размер разделяемых единиц данных, как правило, соответствует кэш-строке. Это обусловлено привязкой к механизму когерентности кэш памяти МП. Аппаратные реализации имеют малую задержку при пересылках данных, но использование таких средств существенно повышает стоимость системы.

Системы с единой разделяемой памятью в основном являются симметричными мультипроцессорными системами SMP. Получившие распространение системы с распределенной разделяемой памятью, как уже отмечалось выше, делятся на типы архитектур NUMA, COMA и RM (системы с рефлексивной памятью). В архитектурах типов NUMA и COMA используется объявление «некогерентных» копий данных несостоятельными, в системах с рефлексивной памятью применяется модификация разделяемых данных. В системах SMP возможно как объявление несостоятельными модифицируемых данных, так и модификация разделяемых данных. С основными принципами организации систем COMA и RM можно познакомиться по литературе.

Все рассмотренные архитектуры систем с распределенной и разделяемой памятью имеют свои достоинства и недостатки. Эффективность использования их зависит от класса задач и других факторов.

## Развитие архитектур микропроцессоров высокопроизводительных вычислительных систем

Можно отметить следующие основные архитектурные способы повышения производительности МП:

- увеличение емкости памяти внутри кристалла (интеграция на кристалл функций управления ОП, так как в большинстве ВМ контроллер памяти реализован в чипсете);

- увеличение состава и числа функциональных устройств (интеграция общесистемных функций в процессоре размещением на кристалле процессора средств реализации сетевых и телекоммуникационных функций, что упрощает создание мультипроцессорных систем и объединение компьютеров в сеть).

Рассмотрим тенденции развития по каждому из отмеченных направлений.

*Организация внутрикристальной памяти.* Традиционно внутрикристальная память МП включает регистры процессора и встроенную кэш-память одного двух или трех уровней. В последних разработках процессоров помимо встроенных кэшей развивается технология встроенной динамической памяти DRAM (части ОП, а не буфера, каковым является кэш). Идея создания однокристального компьютера всегда была популярной. Корпорацией IBM достигнуты положительные результаты в размещении на кристалле с процессором встраиваемого блока памяти EDRAM (Embedded DRAM) достаточно большой емкости. В разработке IBM блок EDRAM емкостью 16 Мбайт занимает менее 20% площади кристалла и обеспечивает обмен с ядром процессора со скоростью свыше 50 Гбайт/с. При этом в МП использовалось программное управление режимом записи кэш-строк путем установки бита, переключающего режимы сквозной и обратной записи. Однако при промахе в кэш возрастает интервал времени между модификацией переменной в процессоре и записью ее в память. Это снижает производительность мультипроцессорных систем. Для преодоления данного недостатка используются специальные команды управления КЭШем, записи минуя кэш и т.п. На настоящий момент существуют разработки многопроцессорных систем с увеличенной по разрядности шиной имеющей высокое быстродействие, что позволяет отказаться от использования кэш-памяти и перейти к буферной памяти.

*Увеличение состава и числа функциональных устройств.* С ростом уровня интеграции возникает задача распределения между функциональными блоками системы дополнительных ресурсов на кристалле. Память – ресурс, непосредственно не проводящий вычисления. Размещение памяти на кристалле имеет целью уменьшить задержки, связанные с извлечением из памяти затребованных данных и команд. За счет этого увеличение емкости памяти на кристалле дает прирост производительности системы. Однако после достижения емкостью памяти некоторой величины, этот прирост производительности оказывается меньше, чем прирост производительности, который можно получить при использовании того же ресурса вентилей кристалла для дополнительного по-



строения функциональных преобразователей (суперскалярная архитектура, введение специальных блоков для мультимедийной обработки – MMX).

Основная проблема эффективного использования дополнительных функциональных устройств на кристалле – это организация их загрузки (параллельной работы). Управлять загрузкой устройств можно динамически, исследуя программный код на стадии исполнения, и статически на уровне компиляции программ. Первый способ используется в суперскалярных архитектурах процессоров, второй – в процессорах с длинным командным словом (см. VLIV-процессоры). Параллельное исполнение команд весьма проблематично и в процессоре предполагает использование следующих механизмов:

- переименование регистров с целью устранения ресурсных зависимостей;
- предсказание переходов;
- динамическое назначение команд на исполнение параллельно работающими устройствами, включая изменение порядка исполнения в сравнении с порядком следования в программе.

*Интеграция функций.* С увеличением уровня интеграции микросхем появилась возможность создания систем на одном кристалле – System On Chip. В кристалле интегрируются функции, для выполнения которых ранее требовались наборы микросхем, сетевые платы и другие средства. При интеграции функциональных устройств системы на одном кристалле с одной стороны уменьшаются задержки при их информационных обменах, что повышает производительность, с другой стороны уменьшается общее число микросхем системы, упрощается изготовление и монтаж плат, а это ведет к снижению стоимости систем и повышению их надежности. Таким образом, интеграция функций системы на кристалле улучшает все общетехнические показатели системы.

В частности, в кристалл процессора интегрируются стандартные интерфейсы сетевых и телекоммуникационных систем, что позволяет без дополнительных адаптеров соединять МП друг с другом, с ТВС. Так в процессорах Motorola MPC8260 поддерживается несколько телекоммуникационных прото-

колов, включающих 10/100 Мбит/с Ethernet, 155 Мбит/с ATM и др. Разработка систем с распределенной разделяемой памятью привела к интеграции в кристалл блока управления когерентностью многоуровневой памяти на кристалле и распределенной ВП, доступ к которой осуществляется через интегрированную в тот же кристалл коммуникационную среду (МП Alpha).

Направления развития мультипроцессорных систем
с распределенной памятью

В мультипроцессорных ВС с распределенной памятью (класс систем MPP) каждый процессор, как правило, работает со своей локальной памятью, а если программе требуется узнать значение переменной, расположенной в памяти другого процессорного узла, приводится в действие механизм передачи сообщений. Такой подход позволяет создавать системы, содержащие тысячи процессоров и достигать суммарной производительности в сотни миллиардов операций в секунду. Основные проблемы, связанные с созданием систем MPP: снижение затрат времени на передачу сообщений; необходимость учета при программировании топологии систем и специального распределения данных между процессорами с целью минимизации числа и объема пересылок данных между узлами системы.

Программы системы MPP, стремящиеся максимально полно использовать ее возможности, оказываются достаточно жестко привязанными к ее архитектуре. В результате число пакетов программ, разработанных для каждой из систем MPP, весьма ограничено.

Рассмотрим основные направления в развитии систем рассматриваемого класса, позволяющие преодолевать отмеченные трудности: использование концепции виртуальной памяти; введение стандартных протоколов интерфейса передачи сообщений *MPI* (упрощает техническую реализацию большого программного кода, но возникают сложности с использованием – т.к. практически осуществляется на машинном уровне). Определенным выходом из этого затруднения стали *ЯВУ*, основанные на параллелизме данных. К ним относятся языки Fortran-D, High Performance Fortran, mpC (вариант языка C).



*Производительность мультипроцессорных систем*
*при увеличении числа процессоров*

Участки вычислений, которые удалось распараллелить (при использовании идеологии МПВС, ММВС), выполняются с высокой скоростью, тогда как обычно сохраняющиеся участки последовательных вычислений выполняются намного медленнее. Все это приводит к большим колебаниям скорости вычислений в высокопроизводительных ВС (в 10–20 раз). Решение этой проблемы – нетривиальная задача.

*Вычислительные системы на кристалле*

Возможность реализации всех блоков ВС на одном кристалле весьма перспективна для разработчиков. Это позволяет улучшить все общетехнические показатели системы: снизить стоимость, повысить быстродействие и надежность.

В сложных дискретных системах общетехнические показатели существенно зависят от технологии реализации соединений между вентилями, элементами, узлами, блоками. Из опыта разработки дискретных систем известны следующие усредненные относительные оценки: стоимость соединения на кристалле в 10 раз ниже, чем на печатной плате и в 100 раз ниже, чем в межплатных соединениях. Соответственно и задержки в связях, существенно влияющие на быстродействие, на печатной плате в 10 раз больше, чем на кристалле. Надежность и помехозащищенность соединений на кристалле также существенно выше, чем на плате и в межплатных соединениях.

Программируемая логика – универсальное средство для разработки цифровых устройств и систем на базе кристалла СБИС. СБИС программируемой логики содержит множества унифицированных функциональных преобразователей и элементов памяти, а также средства программирования функций, выполняемых функциональными преобразователями, и связей между ними, а также связей с внешними выводами СБИС. Программирование выполняется пользователем с использованием сравнительно простых специализированных

устройств – программаторов, программно управляемых от стандартного компьютера. Использование СБИС программируемой логики в составе систем стало «островком аппаратной специализации в системе».

Возможность реализации системы на кристалле появилась в связи с развитием технологии производства СБИС, когда уровень интеграции превысил 1 млн вентилей на кристалле. Появилась возможность реализации на одном кристалле как унифицированных блоков: процессора, памяти, контроллеров шин, так и специализированных устройств, реализуемых средствами программируемой логики на той же СБИС.

Общая структура СБИС системы на кристалле – System On Programmable Chip (SOPC) содержит блоки: процессор, кэш-память, ОП ОЗУ, интерфейс шины для связи с ВП и другими устройствами, последовательный порт, а также поле программируемой логики, позволяющее реализовать специализированные блоки системы.

*Переход к нанотехнологии производства интегральных схем*

Полупроводниковые структуры современных ИС формируются с использованием планарной технологии. Интегральные микросхемы создаются на основе слоев из проводящих материалов, наносимых на поверхность кремниевой пластины. Существует множество материалов для формирования таких слоев. Различные сочетания материалов и методов их нанесения лежат в основе разных технологий создания ИС. Наибольшее распространение получили МОП-технологии и КМОП-технологии. В МОП-технологии используются три слоя, в которых размещаются проводники и элементы транзисторов: диффузионный слой, слой поликремния и слой металла. Эти слои выполняют различные функции.

Основной тенденцией при проектировании ИС является постоянное уменьшение минимального размера участка на кристалле, в пределах которого возможно успешное формирование элементов. При проектировании топологии слоев микросхем задаются не абсолютные размеры, а используются единицы,



кратные некоторому параметру размера *мю*, называемому проектной нормой. Этот параметр приблизительно равен максимальному значению смещения топологического элемента, которое может возникнуть при его формировании на пластине. Параметр и определяет ширину проводника и линейные размеры других компонент полупроводниковых структур. Так ширина проводника на пластине не может быть менее 2*мю*. Таким образом, пропорционально *мю* определяются все линейные размеры элементов на кристалле. Поэтому параметр *мю* непосредственно влияет на плотность размещения транзисторов на кристалле СБИС: снижение линейного размера на 30%, связанное с уменьшением *мю*, позволяет удвоить плотность размещения.

С размерами транзистора, а также с толщиной слоя диэлектрика, отделяющего затвор транзистора от канала, связана величина напряжения питания и энергии, выделяемой при переключении вентиля. Чем меньше размеры, тем меньше напряжение и энергия.

Таким образом, снижение проектной нормы *мю* в технологии изготовления ИС существенно влияет на три важнейших показателя: уровень интеграции (число вентилей на кристалле), быстродействие, энергопотребление.

В развитии производства ИС в зависимости от значения *мю* выделяют три этапа с соответствующими названиями технологии:

- микронная технологии (*мю* > 1 мкм);
- субмикронная технология (1 > *мю* > 0,1 мкм);
- нанотехнология (*мю* < 0,1).

Нанотехнология – это технология XXI в. Для снижения затрат площади кристалла на соединения увеличивают число слоев металлизации при изготовлении СБИС. В настоящее время их пять и более.

Одной из важнейших проблем, связанных с ростом уровня итерации, является проблема снижения энергопотребления кристалла. Она определяется проблемой теплоотвода от переключаемых вентилей. Повышение температуры выше определенного уровня (80–90 $^0$С) ведет к нарушению работы – переходов. Минимально допустимое напряжение питания *E* зависит от порога открывания

МОП-транзистора, а порог зависит от размеров транзистора и толщины слоя диэлектрика, отделяющего затвор транзистора от канала. По мере совершенствования технологии толщина слоя диэлектрика уменьшается. По мере снижения проектной нормы *мю* уменьшается напряжение питания *E*). При переходе к нанотехнологии темп уменьшения снижается. Поэтому проблема локального разогрева отдельных областей кристалла становится все более острой.

*Развитие параллельной обработки данных в системах на кристалле*

По данным фирмы Intel переход от однопроцессорной к мультипроцессорным системам на кристалле позволяет замедлить рост потребляемой мощности при увеличении ресурсов кристалла. Так для однопроцессорных структур на уровне развития технологии в 2002 г. увеличение производительности на 1% связано с увеличением потребляемой мощности на 3%. При мультипроцессорных структурах темп увеличения мощности снижается не менее, чем в три раза.

Мультипроцессорные системы на кристалле обозначают аббревиатурой CMP (Chip Multi-Processor).

Как и при организации мультипроцессорных систем на базе нескольких кристаллов МП (см. выше), в мультиплексорных системах на кристалле CMP в первую очередь получили развитие «симметричные» системы: Chip Multi-Processor Symmetric General Purpose cores - мультипроцессорные системы на кристалле с симметричными ядрами общего назначения (географически общий КЭШ в виде креста, в диагоналях которого процессорные ядра). Минимальное число ядер на кристалле - 2. Обмен информацией происходит через общий кэш. Достоинства таких систем, как и систем SMP: относительная простота разработки, интенсивное взаимодействие ядер, относительная простота программирования. Недостаток - это однотредовые системы (*тред* - это изолированный процесс с собственными командами и данными). Для многих задач при решении можно выделять хорошо распараллеливаемые треды. Особенностью однотредовых систем является слабая зависимость производительности от характера решаемых задач.



Подход к распараллеливанию процессов, основанный на «мультитредовости» получил развитие и в мультипроцессорных системах на кристалле. Такие системы получили обозначение SMT - Simultaneous Multi-Threading. В них организовано параллельное выполнение нескольких тредов с совместным использованием ресурсов на кристалле.

Далее в результате исследования проблем, связанных с наращиванием производительности, предложен класс асимметричных кластерных мультипроцессорных систем ACCMP (Asymmetric Cluster CMP). На кристалле размещается асимметричный кластер процессорных ядер с различной специализированной аппаратурой. Кластер может содержать, например, RISC-ядро процессора, DSP-ядра (специализированные процессоры для цифровой обработки сигналов), блоки памяти, средства коммуникаций: «линки», порты. Асимметричные архитектуры позволяют повысить отношение прироста производительности к приросту мощности в три раза в сравнении с симметричными архитектурами. Но при этом требуется разработка специализированных ядер. Для их повторного использования в различных разработках в фирмах создаются библиотеки IP-ядер.

Направление дальнейшего развития архитектур систем на кристалле - создание мультиядерных архитектур (Multi-Cores Architecture). Особенности и достоинства этого класса архитектур:

- большое число простых процессорных ядер на кристалле;
- большой кэш;
- использование при разработках архитектуры результатов создания и исследования систем SMP;
- эффективность взаимодействия ядер;
- высокое соотношение производительности к мощности;
- простота масштабирования производительности системы.

Основные проблемы, выявленные при разработке новых классов архитектур систем на кристалле и требующие исследования следующие:

- выбор способа соединения процессорных ядер на кристалле (шины, коммутаторы и др.), оптимизация соединений по соотношению производительность - мощность;

- синхронизация процессов в ядрах;

- влияние на производительность системы необходимости «подкачки» данных из внешнего системного окружения кристалла;

- пропускная способность каналов с памятью;

- ОС для работы с новым аппаратным обеспечением;

- метод решения того или иного класса задач для использования возможности параллельной обработки, присущей тому или иному классу архитектур;

- построение оптимизирующих компиляторов для новых классов архитектур и новых методов решения задач.

Решение этих проблем окажет существенное влияние на развитие средств вычислительной техники в XXI в. и на увеличение производительности. Рост производительности СОД оказывает влияние и на организацию взаимодействия ВС (машины) с человеком. В большинстве современных систем используется интерактивный режим взаимодействия человек–машина. При этом действия оператора на пульте вызывают прерывание вычислительного процесса и ввод команд и данных от человека в систему. Ответная информация о результатах выполнения команд по завершении необходимых процессов появляется на экране либо других устройствах вывода.

**Зачетные вопросы, рассматриваемые в теме**

1. Особенности организации современных ВМ: SISD компьютеры, CISC и RISC архитектура.

2. Особенности организации современных ВМ: основы организации суперскалярной обработки данных.

3. Особенности организации современных ВМ: SIMD компьютеры. Матричная, векторно-конвейерная архитектура, MMX технология.



4. Централизованные и распределенные системы обработки данных.

5. Типовая структура микроконтроллера, общие сведения.

6. Параллельная обработка данных как архитектурный способ повышения производительности.

7. Классификация систем параллельной обработки данных. Основные архитектуры многопроцессорных вычислительных систем.

8. Компьютеры класса MISD MIMD.

9. Многопроцессорные вычислительные системы (с общей шиной, с многовходовой памятью).

10. Многомашинные вычислительные системы: многомашинные комплексы, системы массового параллелизма.

11. Вычислительные системы – состояния производства, направления развития высокопроизводительных вычислительных систем, тенденции развития архитектур с общей и разделяемой памятью.

12. Вычислительные системы – развитие архитектур микропроцессоров, направления развития мультипроцессорных систем с распределенной памятью.

13. Общие оценки производительности мультипроцессорных систем при увеличении числа процессоров. Вычислительные системы на кристалле и нанотехнологии.

**Примечания**

При разработке данной темы первоисточником явились изданиях [7, 8], содержание которых было дополнено актуальной информацией полученной из открытых источников.

При изучении настоящей темы так же рекомендуется обратиться к соответствующим презентациям по дисциплине, где представлены в качестве сбавочных данных некоторые сведения по имеющимся в настоящее время достижениям в области компьютерных, сетевых и информационных технологий.

## ТЕМА № 4. ПРИНЦИПЫ ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

*Вопросы, рассматриваемые в данной теме*:

1. Телекоммуникационные вычислительные сети: основные понятия.
2. Организация и работа простейшей сети.
3. Параметры производительности сети.
4. Классификация вычислительных сетей.
5. Архитектурные принципы построения сетей.
6. Семиуровневая эталонная модель взаимодействия открытых систем.

### Теоретические сведения

В рамках данной темы рассматривается первая часть главы 2 «Сетевые технологии».

#### *1. Телекоммуникационные вычислительные сети: основные понятия.*

*Телекоммуникационная вычислительная сеть (информационная вычислительная сеть, компьютерная сеть), ТВС* – это сеть обмена и распределенной обработки информации, образуемая множеством взаимосвязанных абонентских систем и средствами связи.

Средства передачи и обработки информации в ТВС ориентированы на коллективное использование общесетевых ресурсов: аппаратных, программных, информационных. Обобщенная функциональная схема ТВС (рис. 3.1) включает в себя абонентские станции и телекоммуникационную систему. Узлами сети являются абонентские станции АС1,..., АСN.

*Абонентская станция* включает в себя ВМ (либо ВС), программный интерфейс и аппаратный интерфейс.

Для работы в составе сети помимо традиционных функций, свойственных одиночно работающей ВМ, требуется выполнять ряд дополнительных функций,



определяемых разработчиками сети. Часть из них выполняется программно, а часть – специализированными аппаратными средствами.

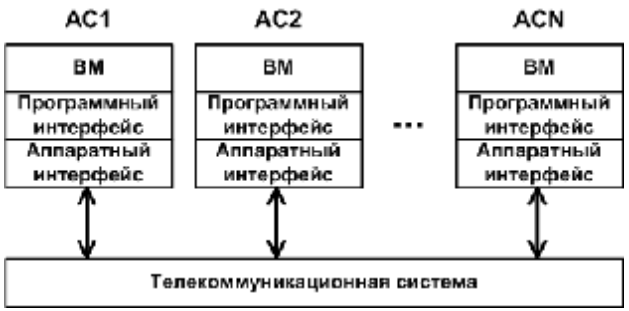



*Рис. 3.1.* Обобщенная функциональная схема телекоммуникационной ВС: АС1, АС2, ..., АСN - абонентские станции; ВМ - вычислительная машина

Программные средства, в абонентской станции, образуют программный интерфейс - часть системного ПО ВМ. Специализированные аппаратные средства образуют дополнительный блок в ВМ, называемый «сетевой адаптер» или «сетевая карта». Этот блок реализует аппаратный интерфейс. Таким образом, абонентской станцией может быть компьютер, содержащий сетевую карту (адаптер) и соответствующее ПО.

*Телекоммуникационная система* (или коммуникационная подсеть) включает физическую среду передачи информации, аппаратные и программные средства, обеспечивающие взаимодействие абонентских станций. Важнейшими функциями при этом являются: синхронизация взаимодействия абонентских систем при обмене информацией, коммутация соединений, маршрутизация сообщений.

Для ТВС большое значение имеют следующие требования: ВМ в сети связываются между собой автоматически; каждая ВМ в сети должна иметь возможность работать как в автономном режиме под управлением своей ОС, так и в качестве узла сети.

Использование ВМ и ВС не автономно, а в составе сети обеспечивает дополнительные возможности информационного обслуживания пользователей и

экономически целесообразно, поскольку позволяет наиболее дорогие ресурсы использовать коллективно.

### *2. Организация и работа простейшей сети.*

Простейшая сеть (рис. 3.2), содержит две ВМ, связанные между собой двумя однонаправленными проводными каналами связи. В конце каждого канала имеется буфер с дисциплиной FIFO. Каждая машина ВМ в такой сети может запросить данные из ОП другой машины: при этом посылается адрес запрашиваемых данных; в ответ запрашиваемая машина считывает данные по адресу, помещает их в буфер и пересылает ответ с данными первой машине. Таким образом, сообщения, которыми обмениваются ВМ, можно разделить на два типа: запрос о данных (содержащий адрес ОП), и ответ со считанными данными. Для обозначения каждого из двух типов сообщений требуется, по крайней мере, один дополнительный служебный бит, образующий заголовок сообщения (рис. 3.3), по анализу которого, и определяется ВМ какое действие необходимо совершить.

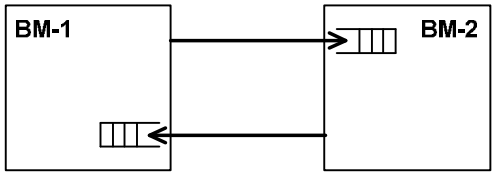



*Рис. 3.2.* Простейшая сеть

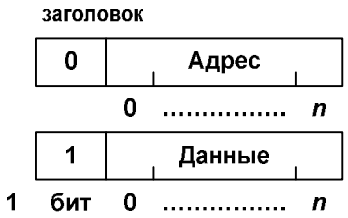



*Рис. 3.3.* Формат сообщений

Это пример организации обмена сообщениями фиксированного формата. В реальных сетях сообщения могут иметь переменную длину и содержат сотни слов. Кроме того, процесс обмена между машинами конкурирует за разделяемые ресурсы с другими процессами в ВМ. Основные функции, связанные с распределением ресурсов, а также с организацией взаимодействия процессов обработки данных и передач и приема сообщений по сети, реализуются программными средствами. Для организации выполнения этих функций требуется более полная служебная информация в сообщениях, что обусловливает расширение заголовка. *Сообщением* называют порцию данных, оформленную для передачи данных по сети в соответствии с принятыми соглашениями.



Весьма важными требованиями при организации сети являются требования к надежности и достоверности передачи сообщений, т.к. канал связи между ВМ в сети имеет много большую протяженность, подвержен воздействию помех. Сбои, происходящие при передаче данных, можно разделить на две категории: искажение данных и потерю сообщения. Для повышения достоверности используют помехоустойчивое кодирование (при подготовке к передаче) и декодирование (для выявления искажения); обе процедуры реализуются ПО. Для выявления потерь передаваемых сообщений используют принцип квитирования и контроля времени между отправкой сообщения и получением подтверждения от приемника о его получении. При посылке сообщения запускается таймер с запрограммированным в нем допустимым временем ожидания ответа; если он срабатывает раньше, чем приходит ответ – потеря сообщения, что вызывает прерывание вычислительного процесса в передающем узле, которое инициирует выполнение соответствующей программы.

Последовательность действий при передаче сообщения следующая:

- прикладная программа, которой требуется обмен с памятью другого компьютера, загружает копию передаваемых данных в буфер программы, управляющей обменом;

- системная программа, управляющая обменом, присоединяет к сообщению заголовок, выполняет кодирование (в простейшем случае вычисляет контрольную сумму и присоединяет ее к сообщению), запускает таймер, используемый для контроля времени ответа;

- системная программа, управляющая обменом, передает данные аппаратному интерфейсу сети и инициирует передачу сообщения по каналу связи.

Порядок действий при приеме следующий:

- поступление сообщения в аппаратный интерфейс сети вызывает инициирование обслуживания этого события системной программой (эта программа копирует полученные данные из аппаратного интерфейса в буфер системной программы);

- производится декодирование данных (в простейшем случае подсчет контрольной суммы и сравнение ее с принятой контрольной суммой) и при отсутствии искажения данных формируется сигнал подтверждения приема (квитирование), а если обнаружено искажение данных, сообщение уничтожается (передающая сторона определит это событие по срабатыванию таймера).

Реакция передающей стороны на подтверждение (квитанцию) от приемной стороны следующая:

- если подтверждение получено, копия сообщения удаляется из системного буфера;

- если подтверждение отсутствует и срабатывает таймер, инициируется повторная передача сообщения и запуск таймера.

В расширенном формате сообщений в рассматриваемой простейшей сети увеличен размер заголовка (00 - запрос данных, передается адрес; 01 – ответ, передаются данные; 10 - подтверждение запроса; 11 - подтверждение ответа), чтобы отразить увеличившееся разнообразие сообщений в канале связи, добавлен концевик с контрольной суммой (либо с остатком от деления на полином при использовании циклического кода).

Приведенную выше последовательность действий передающего и приемного узлов сети при передаче сообщения называют *протоколом* (соглашение о взаимодействии компьютеров или других коммуникационных компонентов в сети, включающее форматы данных и порядок действий).

### *3. Параметры производительности сети.*

Основные параметры показаны на рисунке 3.4.



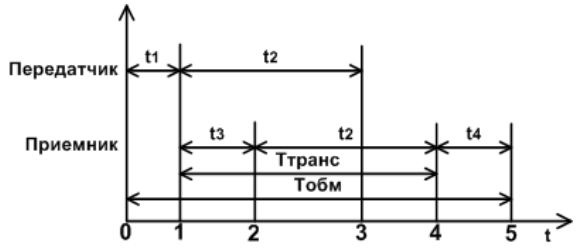



*Рис. 3.4.* Параметры, характеризующие задержки при передаче

По оси абсцисс откладывается время *t*. Цифры характеризуют события при передаче сообщений: 0 - начало передачи; 1 - появление 1-го бита в канале связи; 2 - достижение 1-м битом приемной стороны; 3 - появление последнего бита сообщения на выходе передающей стороны; 4 - достижение последним битом приемной стороны; 5 - завершение обмена. Параметры $\Delta t$, соответствуют интервалам времени, характеризующим различные виды задержек:

$\Delta t_1$ - задержка в передающем узле (интервал от начала передачи до появления 1-го бита в канале связи);

$\Delta t_2$ - время передачи данных (интервал от поступления в канал связи 1-го бита до поступления последнего бита);

$\Delta t_3$ - время продвижения сигналов по телекоммуникационной системе (в рассматриваемом простейшем случае по каналу связи);

$\Delta t_4$ - задержка в приемном узле (интервал от поступления из канала связи последнего бита сообщения до завершения обмена).

Помимо этого используют два интегральных показателя: $T_{транс}$ - время транспортировки сообщения (интервал времени, в течение которого сигналы сообщения присутствуют в телекоммуникационной системе); $T_{обм}$ - время обмена (интервал времени с момента начала передачи на передающей стороне до момента завершения приема на приемной стороне).

Оценка показателей $\Delta t_1$ и $\Delta t_2$ проводится на основе анализа процессов в передающем и приемном узлах. Показатель $\Delta t_3$ включает сумму задержек в канале связи и телекоммуникационном оборудовании (в рассмотренной простей-

шей сети - это задержка в канале связи). Показатель $\Delta t_2$ определяется информационным объемом сообщения $V_C$ (в битах или в байтах) и шириной полосы пропускания телекоммуникационной системы $W_{ТКС}$ (в бит/с либо в байт/с соответственно): $\Delta t_2 = V_C / W_{ТКС}$ . Для интегральной оценки производительности при обмене в сети используется также интегральный показатель эффективной полосы пропускания: $W_{ЭФФ} = V_C / T_{обм}$.

### *4. Классификация вычислительных сетей.*

В основу классификации сетей положены два показателя: число узлов в сети и их удаленность. Это влияет на коммуникационные средства, используемые для объединения в сеть ВМ, ВС и терминалов, а также на организацию телекоммуникационной сети. По этим признакам выделяют четыре вида сетей:

1. Глобальные сети (Wide Area Network - WAN): сети, объединяющие тысячи узлов, удаленность которых может достигать нескольких тысяч километров. Глобальные сети объединяют абонентов, которые могут находиться в разных странах и на разных континентах. В связи с большой сложностью такие сети имеют иерархическую структуру.

2. Региональные сети (Metropolitan Area Network - MAN): сети, объединяющие абонентов района, города, области. Удаленность абонентов обычно составляет десятки-сотни километров.

3. Локальные сети (Local Area Network - LAN): сети, объединяющие до нескольких сотен узлов, удаление которых не превышает нескольких километров. Обычно локальные ВС создают в пределах одного здания.

4. Системные сети (Storage or system Area Network - SAN): это высокопроизводительные ВС, объединяющие до нескольких сотен узлов, с длиной связей до 100 м, располагаемые чаще всего в специальном машинном зале.



### *5. Архитектурные принципы построения сетей.*

*Архитектура сети* (сетевая архитектура) - логическая и техническая организация ТВС, включающая совокупность сетевых аппаратных и программных средств, методов доступа и используемых протоколов.

В начальный период создания ТВС (70-е гг. XX в.), когда сетей было мало, для каждой сети разрабатывались свои протоколы и по-своему распределялись функции между подсистемами. Это затрудняло и удорожало создание сетей. Из всех систем обработки данных и информационного обслуживания ТВС являются самыми сложными и крупномасштабными, объединяющими большое число различных подсистем. Для преодоления влияния сложности на эффективность и стоимость создания и использования ТВС используется иерархический подход к функциональной организации, а также унификация архитектур и стандартизация используемых аппаратных и программных средств.

Физическими блоками структуры сети являются ВМ (либо вычислительные системы, локальные сети), образующие узел сети, терминалы, каналы связи, коммуникационные средства. Они отличаются большим разнообразием по своим характеристикам, функциональным возможностям, конструктивному исполнению.

Представление логической организации сети позволяет абстрагироваться от многих деталей, характеризующих физическое исполнение подсистем и блоков. В логических структурах с точки зрения передачи данных представлены на информационном уровне структурные блоки сети, их архитектура, распределение функций, интерфейсы. На уровне логического представления сети можно по единой технологии управлять каналами связи и устройствами.

Элементами логических структур сети являются узлы, представляющие (на логическом уровне) ВМ, терминалы, устройства коммутации; соединительные элементы, представляющие каналы связи; процессы, представляющие выполнение программ в отдельных узлах.

Форму объединения узлов в сеть характеризует топология сети. Используют различные топологии сетей: «звезда», «кольцо», «дерево» и др.

По содержанию процесса обработки в логических структурах можно выделить три типа обработки: переадресацию данных, обработку сообщений, информационную обработку.

Переадресация подразумевает выполнение функций, обеспечивающих продвижение сообщения по сети без какой-либо обработки данных. К таким функциям относится управление маршрутизацией, передачей данных между соседними узлами и т.п.

Обработка сообщения обеспечивает накопление данных, управление темпом операций, преобразование кода, сжатие данных, шифрование и другие виды дополнительной обработки, не меняющие информационное содержание сообщений.

Информационная обработка обеспечивает преобразование информации, семантика которой определена соглашениями, положенными в основу организации сети, в информацию, семантика которой понятна абоненту.

Разные типы обработки связывают с разными уровнями иерархии протоколов. Понятие иерархии протоколов по существу аналогично понятию модульности в программном обеспечении. Распределяя обработку между несколькими модулями, можно обеспечить гибкость при расширении сети так, чтобы модификация имеющихся и добавление новых функций касались только небольшого числа связанных модулей. Свойство модульности состоит в том, что, приписывая программам пользователя высокий ранг, а каналам передачи данных - низкий, обработку протоколов распределяют между несколькими уровнями иерархии и интерфейсы стандартизуют для каждого уровня. Стандартизуют как протокольные, так и обслуживающие интерфейсы. *Протокольные интерфейсы* обеспечивают передачу данных между модулями одинакового иерархического уровня двух узлов, между которыми идет обмен. *Обслуживающие интерфейсы* обеспечивают связь между соседними иерархическими уровнями внутри одного узла.

Важным вопросом при разработке архитектуры сети является число уровней иерархии протоколов. Чем больше глубина иерархии (больше уровней), тем больше гибкость системы, но тем больше «накладных расходов» при обмене сообщениями. Международной организацией по стандартизации (International Organization for Standardization – ISO) принята и рекомендована 7-уровнёвая эталонная модель взаимодействия открытых систем (ЭМВОС), называемая также моделью – Open Systems Interconnection (OSI). Эта модель является базовой для разработчиков сетей. Реальная архитектура может иметь меньше уровней в целях повышения экономичности реализации.



### *6. Семиуровневая эталонная модель взаимодействия открытых систем.*

В ТВС каждая ВМ рассматривается как *открытая система*: реализованное в ней ПО взаимодействия по сети удовлетворяет некоторому набору универсальных соглашений (протоколов), точное выполнение которых и делает возможным взаимодействие с любой другой ВМ. ЭМВОС - это система стандартных протоколов, которые служат единым руководящим документом для разработки различных архитектур ТВС. Стандарты ЭМВОС распределяют функции передачи данных в сети по уровням иерархии, регламентируют функции каждого уровня и взаимосвязи между уровнями. В эталонной модели представлены следующие стандарты:

Средства межуровневого обслуживания. Для каждого уровня регламентированы семантика и виды обслуживания, но форматы протоколов не устанавливаются, чтобы не ограничивать выбор средств и методов при проектировании систем.

Внутри уровневые протокольные средства. Регламентируются процедуры коммутации, семантика и классы сигналов коммутируемых каналов.

Методы программной и аппаратной реализации протоколов и других средств обслуживания в эталонной модели не регламентируются.

Концепция построения сети как открытой системы предполагает придание ей следующих свойств:

- большое количество элементов, состав которых может изменяться за период жизненного цикла системы;

- большое число функций, реализуемых и системе, состав и способ реализации которых может меняться;

- развитие системы как по составу элементов так и по составу и реализации функций не должно нарушать ее работоспособности.

В архитектуре открытых систем поддерживается «открытость» вширь (по составу элементов) и вглубь (по набору функций). Обеспечение этих свойств накладывает ограничения на структуру связей между элементами, на логику взаимодействия, на функциональную организацию элементов системы.

ЭМВОС не является сборником стандартов, а только дает рекомендации по их разработке. Руководствуясь ЭМВОС, международные и национальные организации по стандартизации разрабатывают стандарты. Следует отметить, что использование международных стандартов и рекомендаций снижает трудоемкость разработки и отладки сетей, но в определенной мере сковывает инициативу разработчиков и их естественное стремление к постоянному совершенствованию ранее принятых решений.

В ЭМВОС функции обработки и передачи данных разделены на семь функциональных групп. С учетом взаимодействия функциональных групп они упорядочены и образуют семь уровней иерархии:

- прикладной уровень (application) – 7;
- представительский уровень (presentation) – 6;
- сеансовый уровень (session) – 5;
- транспортный уровень (transport) – 4;
- сетевой уровень (network) – 3;
- канальный уровень (data-link) – 2;
- физический уровень (physical) – 1;

Нумерация уровней осуществляется снизу вверх. Первым является самый низкий физический уровень. Самым высоким (7) является прикладной уровень. Рассмотрим распределение функций взаимодействия открытых систем по уровням иерархии и краткую характеристику каждого уровня.



*Прикладной уровень* (7), будучи самым верхним, обеспечивает доступ в среду взаимодействия открытых систем прикладных процессов, т.е. он предназначен для организации доступа каждого пользователя к программам удаленных пользователей. На этом уровне предусмотрены такие функции как пересылка файлов, пересылка заданий, обращение к базам данных. На этом уровне функционируют технологии, являющиеся надстройкой над инфраструктурой передачи данных. К ним относятся: электронная почта, удаленный доступ к ресурсам, телеконференции и др.

*Представительский уровень (6)* транслирует различные форматы данных и коды в стандартную форму представления заданий и наборов данных, а также интерпретирует стандартные сообщения применительно к конкретным ВМ и терминалам. Форма выражения информации (синтаксис), обрабатываемой в нескольких узлах сети, может отличаться. Следовательно, для обеспечения возможности обмена информацией между такими объектами необходимо согласование синтаксисов, которое и является основной функцией представительного уровня. Поэтому на этом уровне устанавливают общий синтаксис, который могут понять оба объекта. Здесь осуществляются преобразование форматов, сжатие данных. На практике многие функции этого уровня задействованы на прикладном уровне. Поэтому во многих сетях протоколы уровня представления практически не используются. Это свидетельствует о том, что рекомендации ЭМВОС не являются жесткими.

*Сеансовый уровень* (5) обеспечивает средства, необходимые абонентам для организации, синхронизации и административного управления обменом данными между ними. Уровень обеспечивает по запросам процессов создание портов для приема и передачи сообщений, организацию соединений (логических каналов), а также выполнение универсальных функций управления для протоколов представительного уровня, например установление режимов передачи (односторонняя, полудуплексная или двухсторонняя). Кроме того, на сеансовом уровне реализуются функции контроля местоположения данных в нормальном режиме и функции восстановления для аномальных ситуаций.

Набор функций, требуемых на сеансовом уровне, зависит от вида передачи на прикладном уровне, т.е. от типа диалога, режима пересылки, длины сообщений. Многие функции этого уровня на практике реализуют протоколы транспортного уровня.

*Транспортный уровень (4)* обеспечивает управление трафиком (передачей данных) в системе, управление качеством передачи, расчленение данных на фрагменты (сегментирование данных) и синтез их на приемной стороне, пересылку данных между абонентами. С точки зрения прикладных процессов важно обеспечить требуемую скорость передач. На транспортном уровне устанавливают: время задержки при передаче, длину блока данных в транспортном протоколе и ряд других параметров. Кроме того, на этом уровне обеспечивается уплотнение каналов. Это позволяет одновременно установить несколько транспортных соединений на одном сетевом соединении для более эффективного использования каналов связи. На этом уровне оптимизируют использование услуг, предоставляемых на сетевом уровне, с целью обеспечить максимальную пропускную способность при минимальных затратах. Протоколы транспортного уровня широко развиты и интенсивно используются на практике.

*Сетевой уровень (3)* осуществляет управление логическим каналом передачи данных в сети (адресация, выбор маршрута, коммутация). На сетевом уровне реализуют такие функции, как установка сетевого соединения, выбор маршрута передачи данных (трассировка), ретрансляция, управление потоком путем регулирования объемов пересылаемых данных (структуризация данных, разбивка на пакеты), извещение транспортного уровня о потере данных. Каждый пользователь сети обязательно использует протоколы сетевого уровня, имеет свой уникальный адрес.

*Канальный уровень (2)* осуществляет установление, поддержание и разъединение логических каналов, повышение надежности передач с использованием методов помехоустойчивого кодирования. Основными функциями канального уровня являются установка и разъединение соединения на канальном уровне, выявление ошибок при передаче данных и, если возможно, восстановление передачи. На канальном уровне предусмотрена так называемая функция дробления - установка на одном соединении канального уровня нескольких соединений физического уровня. При этом резервируют несколько каналов связи. Функция дробления предназначена для повышения производительности, распределения нагрузки и обеспечения устойчивой работы при отказах отдельных каналов связи. Протоколы этого уровня весьма многочисленны и существенно отличаются друг от друга своими функциональными возможностями.



*Физический уровень* (7) обеспечивает передачу двоичных сигналов через передающую среду (образуемую, например, каналом связи, модулятором и демодулятором). Передача последовательности двоичных сигналов осуществляется по соединению физического уровня, которое устанавливается между объектами канального уровня. На этом уровне реализуют такие функции управления каналом связи, как подключение и отключение, формирование и прием сигналов. Установка и разъединение соединений канального уровня учитывают все особенности реальной системы. Например, передача одного сообщения нескольким объектам (точка - многоточие) на физическом уровне моделируется как соединение с одной исходящей точкой и несколькими концевыми точками.

Многоуровневая организация обеспечивает независимость управления на $i$-м уровне от порядка функционирования ($i$-1)-го и ($i$+1)-го уровней, что упрощает управление взаимодействием. В частности, прикладной процесс создается только для обработки данных без учета структуры сети, типа каналов связи, способа выбора маршрутов и т. п.

**Зачетные вопросы, рассматриваемые в теме**

1. Принципы построения телекоммуникационных вычислительных систем: понятие, организация, параметры, классификация, архитектура.

2. Семиуровневая эталонная модель взаимодействия открытых систем (OSI).

## Примечания

При разработке данной темы первоисточником явились изданиях [6-8], содержание которых было дополнено актуальной информацией полученной из открытых источников.

При изучении настоящей темы так же рекомендуется обратиться к соответствующим презентациям по дисциплине, где представлены в качестве сбавочных данных некоторые сведения по имеющимся в настоящее время достижениям в области компьютерных, сетевых и информационных технологий.



# ТЕМА № 5. КОММУТАЦИЯ И МАРШРУТИЗАЦИЯ ПРИ ПЕРЕДАЧЕ ДАННЫХ ПО СЕТИ

*Вопросы, рассматриваемые в данной теме*:

1. Коммутация при передаче данных по сети.
2. Решение задач маршрутизации.
3. Оборудование, используемое при передаче данных по сети.

## Теоретические сведения

В рамках данной темы рассматривается вторая часть главы 2 «Сетевые технологии».

### *1. Коммутация при передаче данных по сети*

Для передачи данных между абонентами сети требуется установление соединений, проходящих через узлы и каналы связи. Важнейшие характеристики передачи данных зависят от пропускной способности линий связи, производительности узлов, способов организации каналов связи и способов передачи данных по каналам связи.

Возможны три способа коммутации при передаче данных: коммутация каналов, сообщений и пакетов.

Коммутация каналов обеспечивает создание физического канала, обеспечивающего прямую передачу данных между абонентами сети. Рассмотрим фрагмент сети, содержащий абонентов $a_i$ и $a_j$ а также расположенные между ними узлы связи A, B, C и D. Пусть абонент $a_i$ имеет непосредственную связь с узлом A, а абонент $a_j$ - с узлом D. Рассмотрим организацию обмена между абонентами:

- абонент $a_i$, инициирует установление связи с абонентом $a_j$;
- узел связи A, реагируя на адрес абонента $a_j$ коммутирует соединение линии абонента $a_i$ с линией, соединяющей узел A и узел B;

- процедура соединения повторяется с узлами B, C и D, в результате между абонентами коммутируется канал;

- узел D и абонент $a_j$ посылают сигнал обратной связи;

- получив сигнал обратной связи, абонент $a_i$ начинает передачу данных.

Коммутация сообщений производится передачей сообщения по маршруту, определяемому узлами сети с использованием информации, содержащейся в заголовке сообщения. Сообщение, создаваемое отправителем принимается узлом A и хранится в памяти узла. Узел A обрабатывает заголовок сообщения и определяет маршрут передачи сообщения, ведущий к узлу B. Узел B принимает сообщение, помещает его в память, обрабатывает заголовок и передает сообщение следующему узлу. Процесс приема, обработки заголовка и передачи сообщения повторяется последовательно всеми узлами на маршруте от абонента $a_i$ до абонента $a_j$.

Коммутация пакетов происходит аналогично коммутации сообщений. Отличием данного способа передачи является то, что учитывается ограничение на максимальную длину пакета. С учетом этого ограничения длинное сообщение разделяется на пакеты с ограниченной длиной, и далее пакеты передаются с использованием рассмотренного способа коммутации сообщений.

Коммутация пакетов - основной способ передачи данных в ТВС. При коммутации каналов требуется, чтобы все участки скоммутированного канала имели одинаковую пропускную способность, что накладывает жесткие ограничения на систему передачи данных. При коммутации пакетов и сообщений пропускная способность отдельных линий может отличаться. Представление данных пакетами ограниченной фиксированной длины позволяет эффективно использовать принципы временного разделения каналов для одновременной передачи нескольких потоков данных.

В сетях с коммутацией пакетов используются два способа передачи данных: дейтаграммный и виртуальный канал.

Дейтаграммой называют пакет, заголовок которого содержит адрес получателя и необходимые служебные маршрутные признаки. Совокупность процедур управления передачей таких пакетов называют дейтаграммной службой. Дейтаграммную сеть называют также бесконтактной (нет непосредственного контакта между отправителем и получателем). Особенность организации ее работы заключается в отсутствии фазы установления соединения и фазы разъединения. В ней нет сквозного механизма подтверждений, управления потоком данных, исправления ошибок. Эти услуги обеспечиваются на уровне отдельных звеньев сети. Работу бесконтактной сети можно сравнить с отправкой письма по почте. Обычно письмо доходит до адресата, но автор не знает этого. Получатель для уведомления автора должен послать ему свое (другое) письмо. Это соответствует уже протоколу высокого уровня.

Виртуальный канал – это способ, при котором сначала устанавливается маршрут передачи всего сообщения с помощью специального служебного пакета - запроса на соединение. Для этого пакета и выбирается маршрут, который при согласии получателя на соединение закрепляется для всего последующего трафика и получает номер определенного виртуального канала (соединения) для использования его другими пакетами того же сообщения, направляемыми в тот же адрес. Пакеты, передаваемые по одному виртуальному соединению, не являются независимыми и поэтому могут иметь сокращенный заголовок с номером пакета в составе сообщения. Сеть с установлением виртуальных каналов называют *контактной* в том смысле, что при передаче сообщений предполагается предварительно установление контакта между отправителем и получателем. Контактную сеть можно сравнить с телефонной системой. Человек, который звонит, знает, когда устанавливается соединение.

Сравнивая рассмотренные два способа передачи данных можно отметить следующее:

- виртуальное соединение обеспечивает упорядоченность пакетов и более простое управление потоком данных вдоль маршрута;

- процедура установления виртуального соединения сложнее, чем при передаче дейтаграмм;

- при виртуальном соединении нет возможности учитывать изменение ситуации в сети во время передачи сообщения (при дейтаграммном способе такая возможность существует).

Приведенное сравнение показывает, что для всех случаев нет единой рекомендации выбора лучшего способа передачи данных. При передаче больших массивов данных предпочтительнее виртуальные соединения. При передаче коротких сообщений, что характерно для обмена между ВМ и терминалом, выгоднее использовать дейтаграммы.

### *2. Решение задач маршрутизации*

Адрес, указанный в заголовке пакета, необходим для выбора маршрута. Выбор проводится на основе определенного алгоритма.

Алгоритм маршрутизации - это правило назначения выходной линии связи данного узла телекоммуникационной системы для передачи пакета, основанное на информации в заголовке пакета (адреса отправителя и получателя) и на информации о загрузке этого узла. Существуют различные алгоритмы маршрутизации. Их разделяют на *фиксированные* и *адаптивные*.

В фиксированных алгоритмах выбор направления передачи не зависит от загрузки узлов и каналов.

В адаптивных учитывается текущая информация о состоянии и загрузке элементов сети.

Коммутация пакетов относится к логическим типам коммутации. При логической коммутации в узлах связи часто используют специальные ВМ, осуществляющие прием, хранение, анализ, выбор маршрута и отправку сообщений адресату.

В узлах коммутации выполняются также функции внутрисетевого и межсетевого интерфейса, функции моста - соединителя нескольких сегментов сети.



### *3. Оборудование, используемое при передаче данных по сети*

*Сетевой адаптер* (сетевая карта) – это плата расширения, включаемая в разъем шины PCI компьютера. Сетевые адаптеры различаются выполняемыми функциями и ценой. Их можно разделить на две группы: адаптеры для клиентских компьютеров; адаптеры для серверов.

Адаптеры для клиентских компьютеров проще и дешевле. Они реализуют функции физического и канального уровней. При этом значительная часть функций по приему и передаче сообщений выполняется соответствующими системными программами, в значительной мере загружающими центральный процессор компьютера.

Адаптеры для серверов реализуют функции четырех уровней: физического, канального, сетевого и транспортного. Они имеют собственные процессоры, что позволяет разгружать центральный процессор. Цена таких адаптеров существенно выше, чем клиентских.

*Коммутатор* это устройство, осуществляющее поочередное подключение нескольких входных каналов связи на один выходной без изменения скорости передачи.

*Концентратор* (хаб - hub) это устройство, осуществляющее переключение потока данных из одного либо нескольких каналов на другой (другие). Обычно концентраторы переключают потоки данных от нескольких низкоскоростных каналов на меньшее число более скоростных с использованием функций физического уровня. Они позволяют объединять несколько локальных сетей в одну. Используются для формирования различной топологии сети.

*Повторитель* (repeater) это усилитель, восстанавливающий амплитуду и форму сигнала при передачах на большие расстояния. Он регенерирует пакеты данных, обеспечивает электрическую независимость сопрягаемых сетей и защиту от действия помех.

*Мост* (bridge) это устройство, предназначенное для соединения идентичных сетей. Работа моста соответствует протоколам сетевого уровня. Соединяемые мостом сети могут иметь некоторые отличия на физическом и канальном

уровнях, но должны иметь одинаковые протоколы на сетевом и выше уровнях. Мост выполняет фильтрацию информационных пакетов в соответствии с адресами получателей.

*Шлюз* (gateway) это устройство, позволяющее объединять различные сети. Шлюз выполняет протокольные преобразования всех уровней модели взаимодействия открытых систем.

*Маршрутизатор* (router) это устройство, выполняющее соединение на транспортном уровне. Он обеспечивает соединение логически не связанных сетей, имеющих одинаковые протоколы на сеансовом уровне и выше, создает нужный логический канал и передает сообщение по назначению. Маршрутизаторы могут соединять сети с разными методами доступа, перераспределять нагрузки в линиях связи, выбирая наилучшие маршруты, выполнять буферизацию данных.

*Мультиплексор* это многоканальное устройство согласования. В частности он может выполнять функцию объединения нескольких низкоскоростных каналов связи на один более скоростной методом кодового, частотного либо временного уплотнения.

### Зачетные вопросы, рассматриваемые в теме

Коммуникация и маршрутизация при передаче данных.

### Примечания

При разработке данной темы первоисточником явилось издание [8], содержание которых было дополнено актуальной информацией полученной из открытых источников.

При изучении настоящей темы так же рекомендуется обратиться к соответствующим презентациям по дисциплине, где представлены в качестве сбавочных данных некоторые сведения по имеющимся в настоящее время достижениям в области компьютерных, сетевых и информационных технологий.



## ТЕМА № 6. РАЗЛИЧНЫЕ СЕТИ И ИХ ИСПОЛЬЗОВАНИЕ

*Вопросы, рассматриваемые в данной теме*:

1. Локальные вычислительные сети.
2. Локальная вычислительная сеть Ethernet.
3. Корпоративные сети и сеть Интернет.

### Теоретические сведения

В рамках данной темы рассматривается третья часть главы 2 «Сетевые технологии».

### *1. Локальные вычислительные сети*

ТВС, в состав которой входят ВМ, терминалы и другие устройства, расположенные относительно близко (например, в пределах одного здания) называют *локальной вычислительной сетью* (ЛВС). ЛВС используют для: управления и другого информационного обслуживания в технических системах, оборудование которых распределено в пространстве. Это автоматизированные системы управления технологическим оборудованием, системы управления и контроля оборудованием кораблей, самолетов и др. Здесь осуществляется передачи больших объемов информации на небольшие расстояния.

Основная отличительная особенность ЛВС – наличие единого для всех абонентов сети высокоскоростного канала передачи данных. При этом ЛВС обеспечивает более дешевый способ комплексирования оборудования, чем глобальная сеть. Наиболее важными характеристиками ЛВС являются:

- тип передающей физической среды;
- тип передачи данных (синхронный или асинхронный);
- метод доступа к среде (каналу связи), топология;
- методы передачи данных;
- типы компьютеров в узлах сети; число узлов;

- тип управления в сети (одноранговая или двухранговая - с выделенным сервером),

- типы протоколов, регламентирующих форматы и процедуры обмена информацией.

Рассмотрим типы каналов связи и их характеристики. В ЛВС используют несколько типов каналов связи (передающих сред): витую пару, коаксиальный кабель, волоконно-оптический кабель. Реже используют инфракрасный и микроволновый каналы. От среды зависит ряд характеристик ЛВС (таблица 6.1). Заметим, что данные параметры приведены для сравнения и были усреднены по передающим частотам.

*Таблица 6.1.* Усредненные характеристики сред передачи данных

| Характеристика | Тип канала связи | | |
|---|---|---|---|
| | витая пара | коаксиальный кабель | волоконно-оптический кабель |
| Относительная скорость передачи | До 10 | До 140 | До 500 |
| Относительная дальность передачи | 0,01-0,1 | До 2,5 | До 200 |
| Число узлов в сети | 10-100 | До 100 | 2 точки кольца |
| Сложность соединения | Низкая | Средняя | Высокая |
| Возможность ответвления | Плохая | Средняя | Плохая |
| Помехозащищенность | Средняя | Высокая | Очень высокая |
| Относительная стоимость, включая подсоединение | 1 | 10 | 50 |

Наиболее распространенным является построение каналов связи с использованием витой пары. В тех случаях, когда требуется обеспечить высокую помехозащищенность (например, в производственных условиях с высоким уровнем электромагнитных помех), либо требуется наиболее высокая скорость передачи сигналов, используют волоконно-оптические линии связи. Волоконно-оптические каналы сопрягаются с приемопередатчиками, фото- и светодиодами через специальные оптические разъемы.



В последнее время все большую популярность приобретают радиоканалы – WiFi и сети построенные на оптико-волоконных кабелях (в качестве магистральных и узловых соединений). Витая пара сохраняет лидерство по популярности и в зависимости от типа кабеля и протокола передачи скорость передачи по ней может достигать больших значений – см. презентации к лекциям.

На выбор метода передачи данных существенно влияют два основных требования: уплотнение канала и обеспечение надежной передачи данных. Небольшая протяженность линий связи в ЛВС позволяет использовать в них немодулированные сигналы, что, в свою очередь, позволяет удешевить аппаратуру для передачи и приема сигналов. При организации обмена важной задачей является синхронизация процессов передающей и принимающей стороны. Приемник и передатчик сигналов в сети представляют собой синхронные конечные автоматы. Для организации взаимодействия необходима их синхронизация. Это можно сделать разными способами. В ЛВС различают два типа передачи данных: асинхронный и синхронный. Соответственно используются два соглашения о форматировании сообщений.

При асинхронном формате сообщения каждый байт данных имеет дополнительные сигналы «старт» и «стоп». Назначение этих сигналов состоит в том, чтобы известить приемник о приходе данных, а также дать приемнику время для выполнения некоторых функций, связанных с синхронизацией, до поступления следующего байта. «Старт» и «стоп» - уникальные специальные сигналы, которые распознает приемник. При известной скорости передачи сигналов, восприняв сигнал «старт», в приемнике нетрудно организовать синхронизацию приема отдельных битов данных. Так как синхронизация устанавливается для каждого байта, допустима невысокая точность совпадения частот синхронизирующих импульсов у передающей и приемной стороны, так как неточность не ведет к существенному расхождению по фазе, а корректируется в каждом бай-

те. При этом требования к аппаратуре невысоки, что приводит к снижению ее стоимости.

Более сложной является синхронная передача. При синхронной передаче нет необходимости вставлять старт-стоповые сигналы в каждый байт. Начальные сигналы сообщения называют синхробайтом либо флагом. Основная функция этих сигналов состоит в том, чтобы оповещать приемник о приходе сообщения. При длинных сообщениях требуется средство против накапливания разницы фаз сигналов синхронизации приемника и передатчика. Для решения этой проблемы используют два метода: добавляют в канал связи отдельную линию синхронизации, либо применяют специальные коды. При передаче синхроимпульсов они используются для синхронизации автоматов передатчика и приемника.

Когда узлы разделены большими расстояниями (в сравнении с блоками и устройствами внутри ВМ), экономически более выгодно встроить временную настройку в сам сигнал, вместо того чтобы использовать отдельный сигнал синхронизации. Коды, имеющие средства синхронизации, называют синхронизирующими (а также само синхронизирующими). При использовании таких кодов принимающее устройство может периодически проверять себя, чтобы убедиться в том, что оно опрашивает линию точно в тот самый момент, когда некоторый бит поступает в приемник.

Идея построения синхронизирующих кодов заключается в том, чтобы организовать регулярные и частые изменения уровней сигнала в линии. Переходы осуществляют разделение «0» и «1» в приемнике, и логические схемы отслеживают изменения состояния для того, чтобы выделять «1» и «0» из потока битов для целей настройки. Апробирование обычно проводится приемником с более высокой скоростью, чем скорость передачи данных, чтобы более точно определить элементы данных. Наиболее распространенные цифровые коды: NRZ, RZ, Манчестерский и AMI (рис. 11.10).

*Код NRZ* – это униполярный код без возвращения к нулю. Время делится на дискретные интервалы *x*, соответствующие длительности передачи одного



бита. Уровень сигнала принимает два значения: H (высокий) и L (низкий). Низкий уровень соответствует нулевому напряжению. Смена уровней на границе интервалов происходит тогда, когда смежные биты имеют различное значение. Этот код очень эффективен по полосе частот в канале, но возможности самосинхронизации ограничены (при длинных последовательностях «0» или «1» перепады уровня отсутствуют). Чем больше перепадов уровня сигнала в единицу времени, тем больше требуется полоса частот в канале. Поэтому скорость передачи сигналов в канале связи измеряют в бодах. Бод соответствует одному изменению уровня сигнала в секунду. Код NRZ относительно прост, имеет низкую стоимость и поэтому широко используется.

*Код RZ* (*с* возвращением к нулю) предусматривает, что в представлении каждого бита сигнал изменяется, по меньшей мере, один раз. Код обладает очень хорошими свойствами синхронизации. Основной его недостаток - необходимость двух изменений уровня при передаче каждого бита. Поэтому код RZ требует большей полосы частот в канале, чем код NRZ.

*Манчестерский код* используется во многих системах передачи данных. Он обеспечивает изменение уровня сигнала при передаче каждого бита, что обеспечивает возможность самосинхронизации. Однако, он требует большей полосы частот в канале и ведет к удорожанию аппаратных средств в сравнении с кодом NRZ. Этот код используется в ЛВС.

*Код AMI* является примером биполярного кодирования. В нем для представления «1» используются уровни разной полярности. В отношении синхронизации с этим кодом возникают проблемы при передаче длинных последовательностей «0».

Рассмотрим методы доступа к каналам связи. В ЛВС для передачи информации используют моноканал (mono channel). Моноканал – это канал связи, одновременно используемый несколькими абонентами. Для организации доступа наиболее часто используют временное разделение канала. При этом разработано несколько методов.

При централизованном доступе управляет доступом сервер. Децентрализованные методы доступа реализуются на основе протоколов, обязательных для всех узлов сети. Существуют детерминированные и случайные методы децентрализованного доступа. Детерминированные методы обеспечивают более полную загрузку моноканала. Протокол гарантирует каждому компьютеру в сети определенное время доступа к моноканалу. Обращение компьютера к сети может быть только в отведенные ему протоколом интервалы времени. При случайном методе доступа к моноканалу обращения компьютера могут происходить в любой момент времени, но нет гарантий, что по каждому такому обращению будет эффективно выполнена требуемая передача данных.

Рассмотрим топологии ЛВС. В ЛВС используют различные топологии: «шина», «кольцо» («петля»), «звезда», «дерево».

Шинная топология характеризуется использованием разомкнутого сегмента кабеля, к которому подключаются абонентские станции. Передаваемая станцией информация распространяется в обе стороны. Применение шин снижает стоимость проводки, унифицирует подключение различных модулей, обеспечивает возможность широковещательного обращения. Используют два метода уплотнения передачи сигналов по шине: временное и частотное разделение. При временном разделении каждая станция использует определенный временной интервал для передачи данных, что регламентируется используемым методом доступа к среде. При частотном разделении несколько станций могут одновременно передавать информацию по различным частотным подканалам.

При кольцевой топологии сигналы передаются по кольцу, в большинстве случаев в одном направлении. Каждая станция непосредственно подсоединяется к двум соседним, но «прослушивает» передачу любой станции. Все станции могут иметь равные права доступа к среде, но иногда одна станция выделяется в качестве главной. Она выполняет функцию монитора в сети: инициация и тестирование кольца. Такую организацию (с монитором) называют «петлей».

При топологии «звезда» центральная станция выполняет функции коммутатора в сети, соединяющего различных пользователей. Узел в центре звезды



может выполнять функции, отличные от обычной обработки данных и коммутации каналов. Он может согласовывать скорости передачи и приема у станций, участвующих в обмене. Передатчик и приемник в сети могут работать, используя различные связные протоколы. Существенно, что в такой сети большая часть средств, требующихся для управления сетью, может находиться в одном месте и разделяться между всеми устройствами системы. Для доступа к каналам не требуется специальной логики, так как каждый канал связан с одним устройством. Для разных каналов одной сети, связывающих абонентские станции с центральным узлом, можно использовать различную передающую среду.

Топология «дерево» представляет собой несколько шин, подключенных к магистральной шине через повторители. Иерархическая организация такой сети позволяет упростить ПО для управления.

По типам используемых ВМ в узлах сети делят на однородные и неоднородные (гетерогенные). В однородных ЛВС во всех узлах используются однотипные ВМ с одинаковыми ОС и однотипными средствами для работы в составе сети. Многие процедуры распределенной обработки данных в однородных сетях проще, чем в неоднородных.

ЛВС могут объединять от единиц до нескольких десятков ВМ. Существуют два подхода к организации управления процессами в сети: с централизованным и децентрализованным управлением. В сетях с централизованным управлением среди узлов один выделен для управления и называется сервером. В ЛВС с децентрализованным управлением функции сервера распределены между узлами. Такие ЛВС называют одноранговыми сетями (peer-to-peer).

В одноранговой сети каждая ВМ может выполнять функции как клиента так и сервера. Она может выполнять запросы от других ВМ и направлять свои запросы для обслуживания в сеть. Отсутствие сервера не позволяет администратору централизованно управлять ресурсами. Достоинствами такой организации является низкая стоимость. Недостатки: возможность подключения небольшого числа узлов (не более 10), сложность управления, сложность обновления ПО.

В сетях с централизованным управлением (двухранговых или серверных) один узел (сервер) управляет взаимодействием всех остальных узлов сети и выполняет ряд сервисных функций. Сети, в которых сервер выполняет только функции организации, хранения и выдачи клиентам информации по их запросам, называют системами *«файл-сервер»,* или сетями с выделенным сервером. В таких системах сервер называют *файл-сервером,* как и саму систему. Системы, в которых сервер в дополнение к функциям файл-сервера выполняет еще и функции обработки данных, называют системами *«клиент-сервер»*. В таких системах сервер называют *сервером приложений.* Двухранговые сети позволяют объединять большое число ВМ, имеют более эффективное управление ресурсами, но имеют более высокую стоимость.

### *2. Локальная вычислительная сеть Ethernet*

*Рис. 6.1.* Схема организации ЛВС Ethernet

Одной из распространенных реализаций ЛВС является сеть Ethernet.

Эта сеть была разработана в США в 70-х гг. XX в. Сеть использует топологии: «шина», «дерево», «звезда». В качестве передающей среды используется коаксиальный кабель, витая пара либо волоконно-оптическая линия связи.

В простейшем случае сеть состоит из одного сегмента с топологией «шина». Длина кабеля не должна превышать 500 м.



К кабелю можно подключать до 100 приемопередатчиков от узлов сети. Сеть с топологией «дерево» организуют из нескольких сегментов, соединяемых через повторители. Максимальное число сегментов - 15. Максимальная скорость передачи - 10 Мбит/с. Сеть Ethernet имеет 3-уровневую организацию (рис. 6.1).

Верхний (прикладной) уровень объединяет функции обработки данных и ввода-вывода сообщений. Канальный и физический уровни обеспечивают доступ к моноканалу и передачу по нему сообщений. Функции канального и физического уровней реализуются аппаратно контроллером Ethernet и приемопередатчиком. Контроллер выполняет функции интерфейса с шиной ВМ (обычно с шиной PCI), формирования кадров (разделения сообщения на блоки ограниченной длины), управления связью, кодирования и декодирования. Приемопередатчик устанавливается непосредственно на кабеле и служит для формирования сигналов, приема сигналов и выявления столкновений.

Рассмотрим конструктивную реализацию средств подключения ВМ к ЛВС Ethernet. Контроллер реализован в виде платы расширения с разъемом для подключения к шине ВМ. Плата имеет второй разъем для соединения специальным кабелем с модулем приемопередатчика. Приемопередатчик устанавливается непосредственно на коаксиальном кабеле ЛВС.

Передаваемые по каналу данные представлены Манчестерским кодом. Для исключения потерь данных их передача предваряется последовательностью из 64 бит: 10101010 ... 10101011.

Последовательность «1010 ... 10» - это последовательность импульсов (меандр) с частотой 5 МГц. Код «11», завершающий последовательность, отмечает момент начала кадра. Приемник включается в работу при обнаружении последовательности «101010» ... и после приема «11» начинает принимать кадр. После исчезновения сигналов в канале кадр считается принятым и начинается его обработка.

В ЛВС Ethernet используется метод множественного доступа к среде с контролем несущей частоты и разрешением конфликтов, обозначаемый как ме-

тод CSMA/CD (Carrier Sense Multiple Access with Collision Detection). При этом методе контроллер проверяет наличие сигналов в канале, и после исчезновения несущей частоты начинает передачу кадра в канал с фиксированной задержкой (9,8 мкс). Задержка создаст интервал между кадрами, необходимый для восстановления аппаратурой контроллеров исходного состояния, в котором они могут начинать прием очередного кадра.

В том случае, если при передаче приемопередатчик выявляет столкновение, контроллер прекращает передачу кадра и передает в канал помеху в виде последовательности из 32-48 бит, тем самым «подкрепляя» столкновение. Системе разрешается сделать еще 15 попыток передачи кадра. Если все 16 попыток неудачны, формируется сигнал об ошибке.

Рассмотрим формат кадра ЛВС Ethernet. *Адреса приемников*, кодируемые в начальной части кадра (6 первых байт) могут быть физическими (один узел), групповыми (группа) или широковещательными (ко всем узалам). Тип адреса кодируется в 1-м бите адреса приемника: «0» - физический, «1» - групповой или широковещательный. При широковещательном адресе во всех битах 48 разрядного поля адреса приемников записаны единицы. В поле *адреса источника* (следующие 6 байт) записывается его физический адрес. *Поле типа кадра* (следующие 2 байта) идентифицирует команды и ответы. Для повышения достоверности передачи используются циклические коды. Для *данных* выделяют 46-1500 байт. В поле «контрольная сумма» (последние 4 байта) записывается остаток от деления передаваемого кода на определенный полином.

Согласованный набор протоколов и реализующих их аппаратных и программных средств, достаточных для построения сети, называют сетевой технологией. Самой распространенной сетевой технологией является технология IEEE 802.3/Ethernet. Количество сетей, выполненных по этой технологии, превысило 5 млн. Технология создана в 70-х гг. XX в. В первоначальном варианте в качестве линии связи использовался коаксиальный кабель. Позже были разработаны модификации этой технологии, рассчитанные на другие среды: 10 Base-2 - использует тонкий коаксиальный кабель (четверть дюйма) и обеспечивает



сегменты длиной до 185 м с числом подключаемых узлов до 30; 10 Base-5 - использует толстый (полдюйма) коаксиальный кабель, обеспечивает сегменты длиной до 500 м, с числом подключаемых к сегменту узлов до 100; 10 Base-T - использует витую пару, обеспечивает сегменты длиной до 100м, с числом подключаемых к сегменту узлов до 1024; 10 Base-F - использует волоконно-оптический кабель, обеспечивает сегменты длиной до 2000 м, с числом подключаемых к сегменту узлов до 1024. Скорость передачи при этих технологиях равна 10 Мбит/с.

В развитие технологии Ethernet созданы варианты технологий Fast Ethernet со скоростью передачи 100 Мбит/с (100Base-TX, 100 Base-T4, 100-Base-FX) и Gigabit Ethernet со скоростью 1000 Мбит/с (1000 Base-LX и 1000 Base-SX с использованием волоконно-оптического кабеля, а также 1000 Base-CX и 1000 Base-T с использованием витой пары).

### *3. Корпоративные сети и сеть Интернет*

Интернет это глобальная сеть, которая объединяет множество различных сетей в единое информационное пространство. Интернет может обеспечить информационный обмен между любыми компьютерами, которые входят в состав сетей, подключенных к ней. Основные узлы сети Интернет это ЛВС.

В 1983 г. было принято решение об использовании протоколов TCP/IP на всех узлах сети ARPANET. Установление TCP/IP в качестве стандарта положило начало развитию сети Интернет. «TCP/ IP» обозначает стек протоколов, связанных с протоколами TCP (Transmission Control Protocol) и IP (Internet Protocol).

Протокол IP (рис. 6.2) организует разбиение сообщений на пакеты, выбирает маршрут для передаваемого пакета и обрабатывает получаемые. Протокол TCP - протокол транспортного уровня. Он управляет потоком данных, обрабатывает ошибки, обеспечивает сборку сообщения из пакетов. Детальное описание протоколов, входящих в стек протоколов TCP/IP можно найти в справоч-

ной литературе. Примерная схема соответствия стека протоколов сети Интернет и эталонной модели ЭМВОС представлена в табл. 6.2

*Таблица 6.2.* Соответствие протоколов ЭМВОС

| Уровень | | Протоколы TCP/IP | |
|---|---|---|---|
| ЭМВОС | Интернет | | |
| Прикладной (7) | IV | Telnet, FTP, SMTP, HTPP и др. | TFTP, DNS |
| Представитель- | | | |
| Сеансовый (5) | | | |
| Транспортный (4) | III | TCP | UDP |
| Сетевой (3) | II | IP | |
| Канальный (2) | I | (Вне стандарта TCP/IP) Ethernet, Fast Ethernet и др. | |
| Физический (1) | | | |

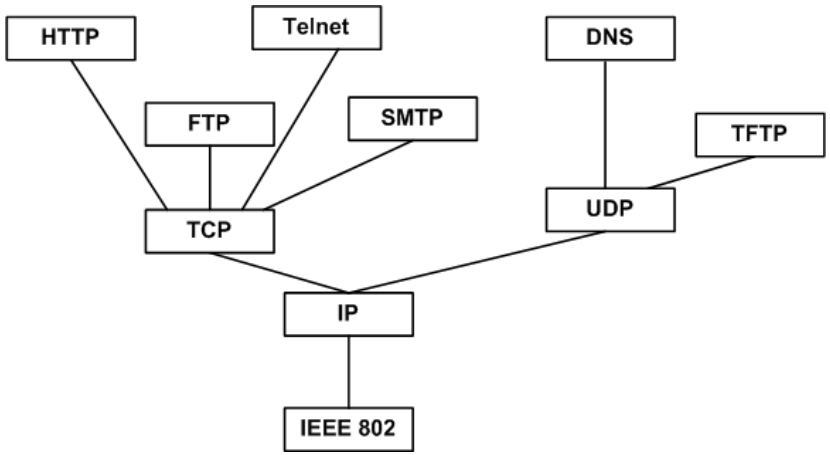



*Рис. 6.2.* Стек протоколов сети Интернет

В иерархии стека протоколов сети Интернет выделены четыре уровня.

Самый нижний уровень (I) соответствует физическому и канальному уровням эталонной модели. Этот уровень в протоколах TCP/IP не регламентируется. Здесь возможно использование всех популярных стандартов физического и канального уровней. Так для локальных сетей - это Ethernet, Fast Ethernet и др.

Следующий уровень (II) - это уровень межсетевого взаимодействия, функции которого предназначены для передачи пакетов. В терминах эталонной модели этот уровень соответствует сетевому. На этом уровне используется про-



токол IP, который проектировался как протокол передачи пакетов между сетями. Поэтому он хорошо работает в сетях со сложной топологией, рационально используя подсистемы и пропускную способность линий связи. Протокол IP является дейтаграммным протоколом, он не гарантирует доставку пакетов до узла назначения, но старается эту доставку осуществить.

Следующий уровень (III) называют основным. Он соответствует транспортному уровню эталонной модели. На этом уровне функционируют протокол управления передачей TCP (Transmission Control Protocol) и протокол дейтаграмм пользователя UDP (User Datagram Protocol). Протокол TCP обеспечивает надежную передачу сообщений между удаленными прикладными процессами за счет образования виртуальных соединений. Протокол UDP обеспечивает передачу пакетов дейтаграммным способом, как и протокол IP, и выполняет только функции связующего звена между сетевым протоколом и многочисленными прикладными процессами.

Верхний уровень (IV) называется прикладным. За долгие годы использования (с 1983 г.) стек TCP/IP накопил большое число протоколов прикладного уровня. К ним относится протокол копирования файлов FTP, протокол электронной почты SMTP, протокол гипертекстового доступа к удаленной информации HTTP и др. Рассмотрим подробнее некоторые из них.

*Протокол Telnet* позволяет пользователю одного компьютера устанавливать связь по сети Интернет с удаленной ВС (обслуживающей машиной). Когда связь установлена, пользователь может работать так, будто его клавиатура, мышь и монитор подключены непосредственно к удаленному компьютеру. Протокол Telnet работает на базе протокола TCP. На прикладном уровне над протоколом Telnet находится либо программа поддержки реального терминала, либо прикладной процесс в обслуживающей машине. Работа с протоколом Telnet походит на набор телефонного номера. Например, пользователь набирает на клавиатуре telnet delta и получает на экране приглашение на вход в машину delta.

*Протокол FTP* пользуется транспортными услугами протокола TCP. Пользователь протокола FTP может вызывать несколько команд, которые позволят ему посмотреть каталог удаленной машины, скопировать один или несколько файлов.

*Протокол SMTP* (Simple Mail Transfer Protocol) поддерживает передачу сообщений электронной почты между произвольными узлами сети Интернет. Он содержит механизмы промежуточного хранения почты и повышения надежности доставки. Протокол обеспечивает группирование сообщений в адрес одного получателя и размножение нескольких копий одного сообщения для передачи в разные адреса. Протокол допускает использование различных транспортных служб.

*Протокол HTTP* (Hyper Text Transfer Protocol) передачи гипертекста используется в одной из самых популярных служб сети Интернет службе «Всемирная паутина» - Word Wide Web (WWW). При помощи этого протокола и клиентской программы - браузера (программы чтения гипертекста) пользователь имеет возможность получать информацию с узлов (сайтов - sites) всемирной паутины WWW.

Существуют несколько способов подсоединения абонента к сети Интернет. Во всех случаях необходимо обеспечить соединение компьютера абонента с сетью, имеющей связь с Интернетом, иметь на компьютере специальное ПО. Услуги, связанные с доступом к Интернету, предоставляют фирмы, называемые провайдерами – Internet Service Provider (ISP). Провайдер имеет ВС, постоянно соединенную с Интернетом и содержащую серверы доступа, через которые осуществляется подключение абонентов. Рассмотрим два распространенных способа подключения абонента к сети.

Первый способ предполагает, что все компьютеры организации (университета) объединены в ЛВС. Сервер ЛВС через устройство сопряжения и выделенную линию связи соединен с сервером доступа провайдера. От типа выделенной линии связи зависит скорость передачи. В качестве выделенной линии могут использоваться: цифровая волоконно-оптическая линия связи со скоростью передачи 600 Мбит/с и более; кабельная линия связи со скоростью передачи от 64 Кбит/с; телефонный канал со скоростью 48 – 56 Кбит/с. Подключение по выделенной линии обеспечивает наиболее комфортные условия для пользователя, но достаточно дорого.



Второй вариант соединения по коммутируемой модемной линии с использованием телефонной сети. Этот вариант предусматривает подключение к серверу доступа провайдера по телефонной линии путем «дозвона» до сервера, регистрации с вводом имени пользователя и пароля. После этого компьютер абонента соединен с сетью Интернет в течение времени поддержания телефонной связи. В настоящее время такой способ почти не используется.

Среди ЛВС выделяют *корпоративные сети.* Корпоративные сети - это сети масштаба предприятия. Они имеют значительную территорию охвата и обычно используют коммуникационные возможности Интернета. Корпоративные сети активно развиваются. Их называют также сетями *интранет* (интра–сеть). Интранет - это такая ЛВС, которая имеет расширенные функциональные возможности благодаря использованию технологий Интернета, имеет доступ в Интернет и при этом имеет защиту от доступа к своим ресурсам со стороны внешних пользователей. Функции защиты выполняют специальные средства - брандмауэры. Они как минимум проверяют полномочия внешнего абонента, знание им пароля. Этим обеспечивается защита от несанкционированного доступа к сети. Информация в Интернет и все ее услуги доступны пользователям сети Интранет.

Корпоративные сети являются частью *корпоративных информационных систем* (КИС). Эти системы пришли на смену автоматизированным системам управления. КИС - это интегрированные системы управления организацией с распределенными в пространстве подразделениями. К важным особенностям КИС относятся: управление, основанное на углубленном анализе данных; использование современных средств информационного обслуживания лиц, принимающих решения; электронный документооборот и делопроизводство.

ТВС - основа создания и развития информационной инфраструктуры современного общества. Поэтому интегрированные сетевые технологии находятся в состоянии быстрого развития и оказывают существенное влияние на различные сферы жизни и деятельности людей.

### Зачетные вопросы, рассматриваемые в теме

1. Локальные вычислительные сети и сеть Ethernet.
2. Корпоративные сети и сеть Интернет.

### Примечания

При разработке данной темы первоисточником явилось издание [8], содержание которых было дополнено актуальной информацией полученной из открытых источников.

При изучении настоящей темы так же рекомендуется обратиться к соответствующим презентациям по дисциплине, где представлены в качестве сбавочных данных некоторые сведения по имеющимся в настоящее время достижениям в области компьютерных, сетевых и информационных технологий.



## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *ГОСТ Р 52653-2006.* Информационно-коммуникационные технологии в образовании. Термины и определения. – Введ. 27.12.2006. – М.: Федеральное агентство по техническому регулированию и метрологии.

2. *ГОСТ 15971-90.* Системы обработки информации. Термины и определения. – Введ. 01.01.1992. – М.: Государственный комитет СССР по управлению качеством продукции и стандартам.

3. *ГОСТ 19781-90.* Обеспечение систем обработки информации программное. Термины и определения. – Введ. 01.01.1992. – М.: Государственный комитет СССР по управлению качеством продукции и стандартам.

4. *ГОСТ 27459-87.* Системы обработки информации. Машинная графика. Термины и определения. – Введ. 01.07.1988. – М.: Государственный комитет СССР по стандартам.

5. *Бройдо, В.Л.* Архитектура ЭВМ и систем [Текст]: учеб.: рек. Мин. обр. РФ /В. Л. Бройдо, О. П. Ильина. – 2-е изд. – СПб.: Питер, 2009. – 720 с.

6. *Пятибратов, А.П.* Вычислительные системы, сети и телекоммуникации [Текст]: Учебник: Рек. Мин. обр. РФ / А.П. Пятибратов, Л.П. Гудыно, А.А. Кириченко; Под ред. А.П. Пятибратова. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2003. – 510 с.:

7. *Теличенко, Д.А.* Цифровые узлы и элементы организации вычислительных систем [Текст]: лаб. практикум /Д.А. Теличенко, М. В. Романова ; АмГУ, Эн.ф. – Благовещенск: Изд-во Амур. гос. ун-та, 2004. – 104 с.

8. *Мелехин В.Ф.* Вычислительные машины, системы и сети: учебник для студ. высш. учеб. заведений /В.Ф. Мелехин, Е.Г. Павловский. – 2-е изд. стер. – М. : Издательский центр «Академия», 2007. – 560 с.

## *СОДЕРЖАНИЕ*

ПРЕДИСЛОВИЕ ..... 3
ВВЕДЕНИЕ ..... 4
УСЛОВЫЕ ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И ОПРЕДЕЛЕНИЯ..... 5
Тема № 1. ИСТОРИЯ РАЗВИТИЯ, АРХИТЕКТУРА И БАЗОВЫЕ ПОНЯТИЯ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ ..... 18
Теоретические сведения..... 18
1. Введение. Основные понятия о процессе автоматизированной обработке данных ..... 18
2. Принцип действия вычислительных машин: логические основы, построение и работа простейших цифровых устройств..... 27
3. Понятие архитектуры. Классификация вычислительной техники 82
4. Функциональная и структурная организация ВМ ..... 92
5. Аппаратные особенности вычислительных машин различных поколений..... 99
Зачетные вопросы, рассматриваемые в теме ..... 108
Примечания ..... 109
Тема № 2. ОСОБЕННОСТИ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И КОМПЬЮТЕРНЫХ СИСТЕМ ..... 110
Теоретические сведения..... 110
1. Общие вопросы организации процессора, памяти и обмена данными ..... 110
2. Персональные компьютеры: особенности архитектуры и применения ..... 148
Зачетные вопросы, рассматриваемые в теме ..... 181
Примечания ..... 181
Тема № 3. СОВРЕМЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ..... 183
Теоретические сведения..... 183



1. Однопроцессорные и многопроцессорные системы. Таксономия М. Флина. Однопроцессорные конфигурации..... 183
2. Централизованные и распределенные системы обработки данных. Вычислительные системы параллельной обработки. Многопроцессорные и многоядерные конфигурации..... 192
3. Вычислительные системы – состояние, производительность, направления развития..... 217
Зачетные вопросы, рассматриваемые в теме ..... 232
Примечания ..... 233
Тема № 4. ПРИНЦИПЫ ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ..... 234
Теоретические сведения..... 234
1. Телекоммуникационные вычислительные сети: основные понятия..... 234
2. Организация и работа простейшей сети..... 236
3. Параметры производительности сети. ..... 238
4. Классификация вычислительных сетей..... 240
5. Архитектурные принципы построения сетей. ..... 241
6. Семиуровневая эталонная модель взаимодействия открытых систем..... 243
Зачетные вопросы, рассматриваемые в теме ..... 247
Примечания ..... 248
Тема № 5. КОММУТАЦИЯ И МАРШРУТИЗАЦИЯ ПРИ ПЕРЕДАЧЕ ДАННЫХ ПО СЕТИ ..... 249
Теоретические сведения..... 249
1. Коммутация при передаче данных по сети ..... 249
2. Решение задач маршрутизации..... 252
3. Оборудование, используемое при передаче данных по сети ..... 253
Зачетные вопросы, рассматриваемые в теме ..... 254
Примечания ..... 254

Тема № 6. РАЗЛИЧНЫЕ СЕТИ И ИХ ИСПОЛЬЗОВАНИЕ ..... 255
Теоретические сведения..... 255
1. Локальные вычислительные сети ..... 255
2. Локальная вычислительная сеть Ethernet..... 262
3. Корпоративные сети и сеть Интернет ..... 265
Зачетные вопросы, рассматриваемые в теме ..... 270
Примечания ..... 270
БИБЛИОГРАФИЧЕСКИЙ СПИСОК ..... 271
СОДЕРЖАНИЕ ..... 272



**Денис Алексеевич Теличенко**

*доцент кафедры АППиЭ АмГУ, канд. техн. наук*

**«Компьютерные, сетевые и информационные технологии».**

*Учебное пособие.*

---

Изд-во АмГУ. Подписано к печати 14.10.13. Формат 60x84/16. Усл. печ. л. 17.21.
Тираж 100. Заказ 157.
Отпечатано в отделе оперативной типографии АмГУ.