

**МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО  
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ  
УЗБЕКИСТАН**

**ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ  
ИНСТИТУТ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ГРАЖДАНСКОЙ АВИАЦИИ**

**Кафедра «Управление воздушным движением»**

**М.К.Арипджанов, Э.Улжаев**

**Конспект лекций  
по курсу «Наземные информационно-вычислительные сети»  
Часть 1 «Распределенные автоматизированные системы  
управления (АСУ)»**

**По направлению: 5А 522008 – «Системы и устройства локации, навигации  
и управления»**

**Ташкент 2003**

М.К.Арипджанов, Э.Улжаев (Таш ГТУ им. А.Р.Беруний каф."АТ") «Наземные ИВС» Часть 1 «Распределенные АСУ»  
Ташкент, Таш ГАИ, 2003 г.

Конспект лекций обсужден и одобрен на заседании кафедры «УВД»  
« \_\_\_\_ » \_\_\_\_\_ 2003г. протокол № \_\_\_\_\_ и рекомендован к размножению.

Одобен и рекомендован к внутривузовскому изданию на заседании  
методсовета ФГА. Протокол № \_\_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 2003г.

Председатель:

доц. А. Абдурахманов

Рецензент:

доц. Т.З.Закиров

**Лекция №1**

# **Архитектура распределенных автоматизированных систем управления технологическими процессами (АСУ). Функционально-программная структура**

## **План.**

- 1. Архитектура распределения АСУ.**
- 2. Функционально программная структура.**
- 3. Прикладные функции АСУ.**

### **1..Архитектура распределения АСУ.**

Распределенная АСУ представляет собой структурно сложную систему. При построении таких систем в последние годы применяется так называемый архитектурный подход, возникший первоначально при разработке вычислительных сетей.

Под архитектурой вычислительной сети понимают функциональные, логические и физические принципы организации сети. При создании этих сетей, состоящих из большого числа компонентов зачастую различного происхождения, уже на весьма ранних стадиях проявилась необходимость в унифицированных архитектурных решениях. Одни из таких унифицированных решений распространяются на все классы вычислительных сетей, другие применяют только в локальных вычислительных сетях; наконец, отдельные архитектурные решения принимают как унифицированные лишь в распределенных АСУ.

Функционально распределенные АСУ организуют по большей части как взаимодействующие между собой по определенным принципам слои, или уровни. Физическая структура определяет соединение компонентов сети передачи данных распределенной АСУ. Логическая структура устанавливает правила взаимодействия между функциональными уровнями и объектами внутри одного уровня и оформляется в виде междууровневых интерфейсов и внутриуровневых протоколов.

### **2. Функционально-программная структура**

Практика разработки вычислительных сетей привела их создателей к введению некоторой стандартной декомпозиции функций - разделению их на архитектурные слои, или уровни.

Уровни функциональной структуры условно располагаются один над другим, причем только верхний выполняет прикладные функции, т. е. те, ради которых создана система, остальные уровни выполняют внутренние (служебные) функции системы, с помощью которых реализуются прикладные функции.

Архитектурные уровни функциональной структуры разделяются интерфейсами, через которые происходит обмен кадрами информации по правилам, устанавливаемым интерфейсами.

В целях упрощения анализа, синтеза, эксплуатации и развития уровни функциональной структуры вычислительной сети выделяются по определенным принципам, оправдывающим принятое расслоение структуры.

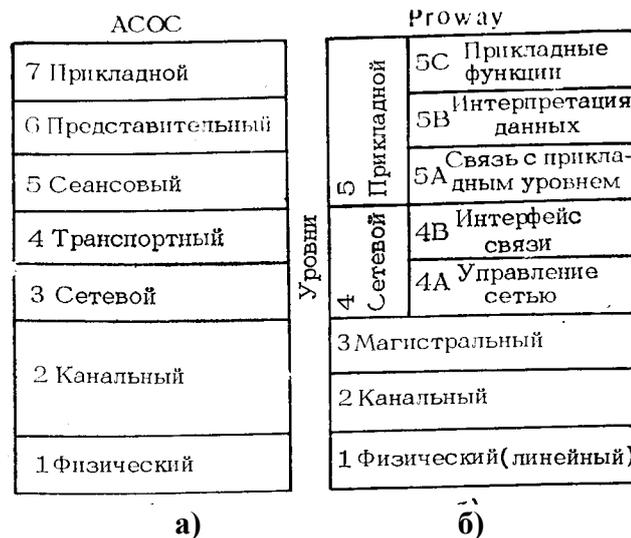
Один из принципов заключается в иерархичности расслоения. Каждый уровень пользуется в осуществлении своих функций средствами (наряду с собственными) только соседнего с ним нижнего уровня; к средствам нижних уровней, отделенных от него другими уровнями, данный уровень может обращаться только через соседний с ним нижний уровень. К средствам верхних уровней данный уровень обращаться не может.

Другой принцип требует максимальной независимости решений, принимаемых на каждом уровне, относительно решений на остальных уровнях архитектуры и в первую очередь, очевидно, в соседних (сверху и снизу) уровнях. Независимость решений на каждом уровне достигается таким расслоением, которое по возможности сводит к минимуму их взаимодействие.

Еще один принцип расслоения на уровни требует прозрачности средств каждого данного уровня относительно содержания информации, поступающей с верхнего уровня. При каждой передаче кадра информации с верхнего уровня на нижний они сопровождаются «упаковочной» информацией, описывающей те особенности кадра, которые представляют интерес для нижнего уровня, такие, как размер кадра или адрес его назначения. Эта «упаковочная» информация, которую удобно назвать конвертом, в общем случае делится на две части: первая, предваряющая кадр, называется заголовком, вторая, его замыкающая, - концевиком. Каждый уровень «читает» только надпись на предназначенном ему конверте, сохраняя прозрачность относительно самих данных, в состав которых могут входить и конверты более высоких уровней.

Выделение архитектурных уровней, сообщающее широкую независимость решениям внутри каждого уровня, значительно облегчает создание и развитие вычислительных сетей. Исторически каждый разработчик сети выделял архитектурные уровни по-своему, однако их принципы оказались в основном сходными, и на этой основе Международной организацией по стандартизации (МОС, по-английски ISO) была разработана архитектура связи вычислительной сети, получившая название а р х и т е к т у р ы соединения открытых систем (АСОС, по-английски OSIA).

Архитектура, соединения открытых систем предусматривает расслоение функциональной, а следовательно, и программной структуры вычислительной сети на семь уровней, представленных на рис.1,а.



**Рис.1. Архитектурные уровни вычислительной сети:**  
**а - архитектура соединения открытых систем; б - архитектура Proway**

Нижний уровень - ф и з и ч е с к и й - охватывает управление процессами в каналах передачи данных: генерирование импульсов определенной формы в определенные моменты времени. Второй снизу уровень - к а н а л ь н ы й - управляет логикой канала передачи данных: здесь устанавливается и прекращается взаимодействие абонентов, определяются форматы передаваемых кадров информации, размещение данных в кадре и описание кадра, проверяется достоверность принятых данных. К третьему снизу уровню, называемому с е т е в ы м, относятся функции, с помощью которых из отдельных каналов передачи данных между соседними узлами организуется сквозной канал между узлами сети; это - виртуальный канал, который вовсе не обязательно должен существовать физически.

Четвертый, т р а н с п о р т н ы й уровень АСОС связывает сквозным виртуальным каналом терминалы: вся сеть для соответствующей пары терминалов представляется как некий «транспортный ящик», внутренний механизм действия которого терминалам безразличен.

В то время как четыре нижних уровня относятся к функциям сети передачи данных, верхние три уровня - надсетевые - охватывают функции, которые обеспечивают совместную работу нескольких вычислительных процессов, независимо от того, требуется ли для этого пользоваться сетью передачи данных. Иными словами, программное обеспечение надсетевых уровней нужно в любой АСУ.

К пятому уровню (нижнему из трех надсетевых) относятся функции управления с е а н с а м и обмена данными между взаимодействующими процессами (задачами). Шестой уровень служит для представления кадров, которыми обмениваются процессы, процессу-получателю: здесь вырабатывается описание кадров для процесса-получателя и выполняется их преобразование в подходящую для него форму. Наконец, функции самого верхнего, седьмого уровня - п р и к л а д н о г о - выполняют содержательную обработку данных.

Соблюдение стандартизированных АСОС архитектурных уровней при построении вычислительных сетей означает разделение программ, реализующих функции каждого уровня. На практике это требование не всегда удобно или целесообразно выполнять, особенно в локальных сетях и АСУ, где сетевой или транспортный уровень может оказаться вырожденным. Что же касается трех верхних, надсетевых уровней, то, пожалуй, ни в одной из реализованных АСУ такое разделение программ не осуществлено.

Для АСУ в настоящее время унифицируется в международном масштабе магистраль передачи данных Proway, рекомендации по которой подготовлены подкомитетом 65А Международной электротехнической комиссии. Рекомендации относятся к нижним уровням функциональной структуры и не касаются трех верхних уровней АСОС. Однако расслоение функциональной структуры Proway (рис.1,б) не совпадает полностью с расслоением на нижних уровнях АСИС.

Магистраль как физическая структура, получающая преобладающее распространение в распределенных АСУ, представляет собой особый вид сети: ее особенность заключается в том, что в каждый данный момент в ней может существовать только один информационный канал. Поэтому проблемы управления сетью в магистрали сводятся к управлению доступом станций к магистрали.

Задачи, относящиеся в АСОС к сетевому уровню, возникают при образовании сети из нескольких магистралей, и это обстоятельство отражается введением в Proway сетевого уровня, лежащего над магистральным. К сетевому же уровню относятся в стандарте Proway те функции, которые в АСОС возлагаются на транспортный уровень, а именно: сопряжения прикладных процессов с сетью - образования пакетов и получения их из сети.

Различие в расслоении означает не просто разный методологический подход к анализу и описанию систем, но и разное исполнение аппаратуры и программ, так как рекомендации Proway, как и АСОС, требуют независимости реализационных решений на каждом уровне.

В то время как в архитектурном аспекте функциональная структура распределенной АСУ определяется применяемыми программными средствами, конкретное назначение АСУ задает в первую очередь структуру ее прикладного уровня.

Структура прикладного уровня АСУ может быть представлена ориентированным графом, в котором вершины соответствуют источникам и получателям информации, а дуги - процессам передачи информации. Назовем этот оргграф *функциональным графом*. Каждому внешнему источнику и получателю информации соответствует *терминальная* вершина. В терминальные вершины источников не входит ни одна дуга и ни одна дуга не исходит из терминальных вершин получателей, между тем как с каждой внутренней вершиной функционального графа, представляющей прикладную задачу, обязательно связана по крайней мере одна входящая в нее и одна исходящая из нее дуга. В дальнейшем из-за многозначности и общности

понятия задачи, а также из-за близости понятий задачи и функции будем называть прикладные задачи п р и к л а д н ы м и операциями.

Понятие прикладной операции весьма неопределенно, поскольку степень дробления действий, выполняемых АСУ, произвольна и зависит от цели, которую ставит перед собой проектант или исследователь. Принцип выделения операций может заключаться в недопустимости объединения в одной операции действий, выполняемых с разной периодичностью, но если речь идет о нерегулярных действиях, то этим принципом трудно воспользоваться. Во всяком случае, целесообразно выделять в качестве прикладной операции некоторую вычислительную программу, выполняемую каждый раз над одними и теми же именами данных из одних и тех же источников и направляющую результаты с одними и теми же именами одним и тем же потребителям. Выделенные таким образом прикладные операции можно отождествить с прикладными вычислительными процессами - понятием, широко применяемым в современной вычислительной технике.

Здесь уместно связать с функциональным графом понятие прикладной функции, или функции АСУ, которое будем использовать впоследствии при анализе эффективности распределенной АСУ. Функция АСУ требует для своей реализации выполнения одной или нескольких прикладных операций по данным, получаемым АСУ извне, и вывода результата на ТП или человеку-оператору. Если при выполнении некоторой операции вычисляется несколько выходных переменных, имеющих самостоятельный прикладной смысл, то каждой такой переменной ставится в соответствие отдельная функция и выполнение одной операции обеспечивает реализацию нескольких функций АСУ. Если же прикладной смысл имеет только набор выходных переменных, то одна функция АСУ связывается с такой векторной переменной.

В АСУ все источники и приемники информации могут быть отнесены к одному из следующих классов: объекту управления; человеку-оператору; ЭВМ вышестоящего уровня; архиву (последний - только приемник). Это деление можно положить в основу классификации прикладных функций АСУ. Выделим следующие классы функций:

- автоматическое регулирование и управление (объект - объект);
- измерение и контроль (объект - оператор);
- ручное управление (оператор - объект);
- автоматическая регистрация процессов и событий на объекте и действий оператора (объект - архив и оператор - архив).

Отсутствие однозначного соответствия между выделенными классами прикладных функций и классами источников и приемников информации объясняется прежде всего тем, что в прикладной функции АСУ источники информации могут принадлежать более чем к одному классу, тогда как в классификации прикладных функций указан только один класс - источника, который можно считать основным. Так, автоматическое регулирование (класс объект - объект) обычно выполняется по заданию, вводимому оператором или вышестоящей ЭВМ. Измерение или регистрация может производиться автоматически (в функции времени или по внутренним тактам системы либо

при совершении события, связанного с состоянием объекта) или же по вызову оператора.

ЭВМ верхнего уровня обычно не воздействует непосредственно на технологический процесс, а лишь вводит задание в операцию по прикладной функции класса объект - объект. Сообщения оператору АСУ из ЭВМ верхнего уровня применяют весьма редко, поэтому здесь не выделен соответствующий класс прикладных функций. Архив, сохраняемый на случай необходимости дальнейшего анализа, может накапливаться как на носителях, так и в долговременной памяти ЭВМ АСУ или вышестоящего уровня, например на магнитных дисках. Рассмотрим в качестве иллюстративного примера чрезвычайно упрощенную распределенную АСУ, представленную на рис.2. Автоматическая система управления служит для управления двумя взаимосвязанными установками.

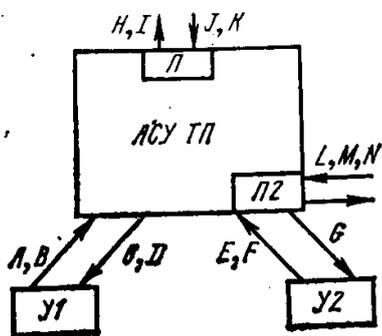


Рис.2. Упрощенный пример АСУ:  
П, П2 – операторские пункты;  
У1, У2 - Технологические установки

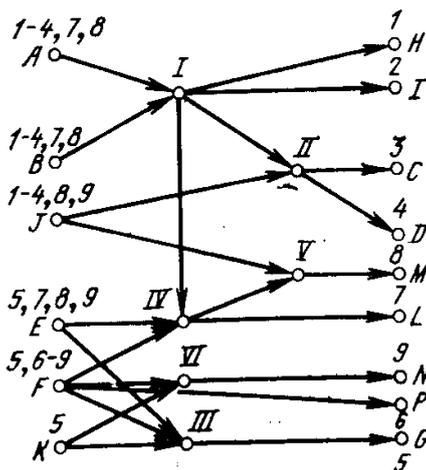


Рис.3. Функциональный граф АСУ, представленной на рис.2

На первой установке У1 измеряются переменные А и В на второй У2 - Е и F. Управляющие воздействия: на У1 - С Р, У2 - Установка У2 снабжена местным пультом управления П2, с которого оператор по показателям Н и I ВЫВОДИМЫМ на дисплей, может дистанционно вводить управляющее воздействие G. Кроме того имеется центральный пункт управления П, общий для обеих установок. Оператор центрального пункта вводит уставки регулирования J и K с воздействиями C, D и G и контролирует ход процессом логических переменных Н, I, выводимым на дисплей и измерений техно в установках по результатам обработки печатающее устройство.

### 3. Прикладные функции АСУ

АСУ выполняет следующие прикладные функции:

I - измерение переменной А с коррекцией по переменной В; результат измерения Н представляется на экране дисплея, установленного в центральном пункте П;

2 - то же измерение с представлением результата функции  $I$  на печатающем устройстве центрального пункта  $П$ ;

3 - регулирование переменной с управляющим воздействием  $C$  по значению переменной  $A$ , скорректированной по  $B$ ; уставка  $J$  вводится оператором (человеком) с клавиатуры дисплея на пункте  $П$ ;

4 - регулирование переменной с управляющим воздействием  $D$  по тем же исходным данным, что и переменной с управляющим воздействием  $C$ ;

5 - регулирование переменной с воздействием  $G$  по измерению переменных  $E$  и  $F$ , а также уставке  $K$ , вводимой оператором с клавиатуры дисплея на пункте  $П$ .

6 - представление результата измерения  $P$  переменной  $F$  на дисплее местного пункта  $П2$ ;

7 - вычисление показателя  $L$  по измерениям переменных  $A$  (с коррекцией по  $B$ ),  $E$  и  $F$  и представление результата на дисплее центрального пункта  $П$ ;

8 - вычисление показателя  $M$ , зависящего от уставки  $J$ , вводимой оператором, и показателя вычисляемого по функции 7, с представлением результата на дисплее местного пункта  $П2$ ;

9- вывод на экран дисплея на местном пункте  $П2$  показателя  $N$ , определяемого по измерению переменной и уставки  $K$ , вводимой с клавиатуры на пункте  $П$ .

Для реализации перечисленных прикладных функций должны выполняться следующие операции вычисления:

I - коррекции переменной  $A$  и  $B$ ;

II- управляющих воздействий  $C$  и  $D$ ;

III - управляющего воздействия  $G$ ;

IV - показателя  $L$ ;

V - показателя  $M$ ;

VI - показателя  $N$ .

Функциональный граф АСУ приведен на рис.3, где арабскими цифрами обозначены прикладные функции у соответствующих терминальных вершин, помеченных буквами технологических переменных, а римскими - операции.

Перейдем к уровню представления. На этом уровне в рассматриваемой АСУ нужно выполнить ряд операций преобразования формы представлений данных. Наряду с представлением данных при передаче их от одной прикладной операции к другой для АСУ ТП характерно преобразование данных при вводе их в АСУ из процесса, при ручном вводе, при выводе оператору и при выводе на процесс.

При вводе переменных в датчиках производится первичное преобразование данных. Если выходом датчика служит аналоговый сигнал, то последний преобразуется в цифровую форму, а затем, чаще всего программным путем, масштабируется и линеаризуется для приведения к стандартной шкале. Далее определяется адрес назначения вводимых данных.

Представление данных при ручном вводе заключается преимущественно в переходе от функционального шифра, под которым вводятся данные, к адресу в памяти, по которому эти данные должны быть направлены.

При выводе управляющего воздействия на процесс определяется физический адрес устройства вывода, а если требуется вывод в аналоговой форме, то производится цифро-аналоговое преобразование.

Наконец, при выводе данных оператору наряду с определением физического адреса устройства вывода производится преобразование выводимых данных в форму, удобную для чтения и приспособленную к используемому устройству вывода - дисплею, печатающему устройству, графопостроителю, цифровому индикатору и т. д.

Поскольку представление данных в общем случае требуется при каждом обмене между вычислительными процессами, а также между вычислительными процессами, с одной стороны, и внешними источниками и приемниками информации, с другой, каждой дуге в функциональном графе соответствует по крайней мере одна операция представления, содержание которой зависит от прикладных операций, соответствующих вершинам дуги.

Переходя к уровню сеансов, заметим, что каждая дуга функционального графа задает требование об организации сеансов обмена, т. е. операции передачи данных. Эта передача в зависимости от физической структуры АСУ может происходить внутри одного устройства, между несколькими устройствами внутри одного узла, между соседними или удаленными узлами сети. Следовательно, программа управления сеансом существенно определяется физической структурой АСУ ТП.

В операциях передачи данных внутри устройств и внутри узла не используются каналы связи сети. Кроме того, что особенно характерно для АСУ непрерывными процессами, по значительному количеству линий терминальной сети АСУ передаются аналоговые сигналы, передача которых идет непрерывно по линии, выделенной для сигнала данной переменной. Обе эти операции передачи данных обходятся без участия нижних уровней функциональной структуры сети передачи данных.

Операция представления данных в каждом звене информационной связи (дуге функционального графа) может выполняться либо до, либо после передачи, а в ряде случаев оказывается целесообразным разбить операцию представления на две, выполняя часть программы представления до передачи, а часть - после нее. Решение о порядке выполнения операций представления и передачи данных следует принимать уже после определения физической структуры АСУ и размещения прикладных операций в узлах.

В заключение отметим, что в конкретных распределенных АСУ или комплексах средств, выпускаемых для построения распределенных АСУ, уровни функциональной структуры могут существенно отличаться от охарактеризованных здесь стандартных уровней или даже вовсе отсутствовать. Так, например, в [7] описана архитектура локальной сети АСУ производством со следующими уровнями: физический канал, обработка пакетов, управление подключением, логический канал, системный процесс, процесс пользователя. Однако развитие техники распределенных АСУ демонстрирует неуклонное стремление к стандартному выделению уровней функциональной структуры, а

следовательно, стандартной структуризации операционных систем, а в ряде случаев и используемой в АСУ аппаратуры.

### Контрольные вопросы

1. По структуре какую систему представляет собой распределенная АСУ? [Л.7, 10-11].
2. Расскажите о уровнях функциональной структуры вычислительной сети? [Л.7, 10-11].
3. Чем отличается архитектура Proway от архитектуры соединения открытых систем? [Л.7, 11-12].
4. Можно ли представить структуру прикладного уровня АСУ ориентированным графом? [Л.7, 12-13].
5. Выделите классы функций АСУ? [Л.7, 13-14].
6. Расскажите назначение ЭВМ верхнего уровня. [Л.7, 15-16].
7. Какие прикладные функции выполняет АСУ? [Л.7, 16-17].
8. Начертите функциональный граф АСУ для какого-либо примера. [Л.7, 17-18].
9. Приведите упрощенный пример для АСУ. [Л.7, 17-18].
10. Чем отличаются уровни функциональной структуры конкретных распределенных АСУ от стандартных уровней? [Л.7, 19-20].

### Лекция №2

## Интерфейсы и протоколы

### План

1. Физическая структура.
2. Интерфейсы.
3. Протоколы

### 1. Физическая структура.

Физическая структура распределенной АСУ образуется соединением терминальной аппаратуры, связывающей АСУ с технологическим объектом и с людьми, управляющими процессом, и аппаратуры передачи и обработки данных.

Информация извне, от технологического процесса или человека-оператора, воспринимается аппаратурой, установленной непосредственно на объекте или на пункте управления. Точно так же непосредственно на объекте или на пункте управления устанавливается аппаратура, выводящая управляющее воздействие или информацию для контроля. Каждое место установки аппаратуры, связывающей АСУ с внешними для нее объектами, назовем *терминальным узлом*. Остальные узлы физической структуры будем называть *внутренними*.

Узлы физической структуры соединяются между собой в е т в я м и, образуя сеть. Физически ветви реализуются с помощью к а н а л о в с в я з и.

Узлы, содержащие процессоры и соединяющие их ветви, образуют базовую сеть, а терминальные узлы и ветви, соединяющие их с базовой сетью, терминальную сеть. В последнюю могут входить также мультиплексные узлы, где размещается аппаратура, выполняющая поочередное подключение терминальных узлов к базовой сети.

На рис.4 показаны базовая и терминальная сети одной из возможных структур АСУ, приведенной на рис.2.

Терминальная сеть присутствует в АСУ любой физической структуры, базовая же сеть характерна для распределенных АСУ. В АСУ с централизованной физической структурой базовая сеть вырождается в единственный узел.

По ветвям базовой сети данные всегда передаются в цифровой форме, тогда как сигналы в ветвях терминальной сети имеют цифровую либо аналоговую форму в зависимости от принципа действия терминальной аппаратуры комплексных узлах. Линию связи, по которой данные передаются в цифровой форме, называют также линией передачи данных.

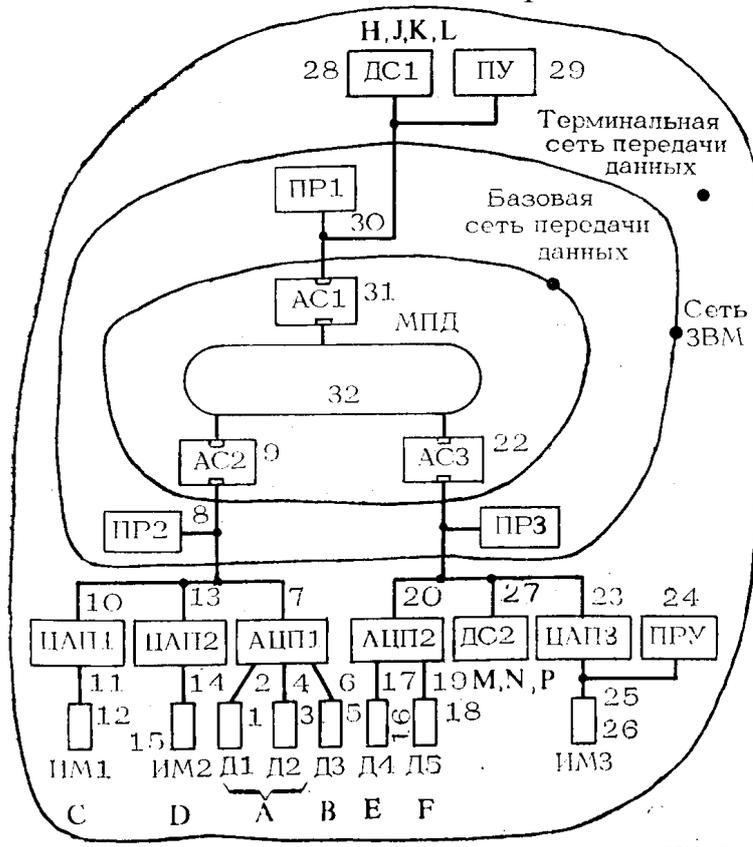
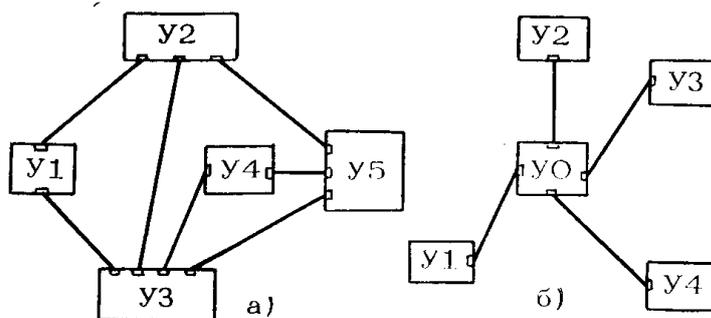


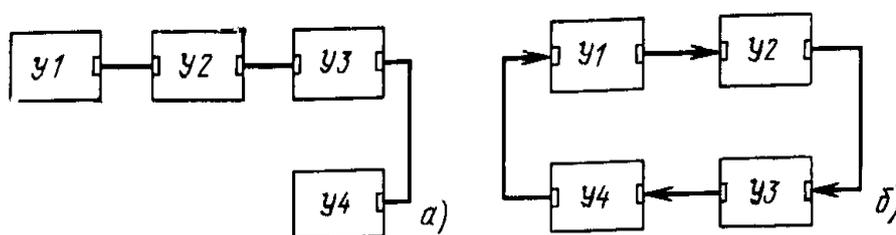
Рис.4. Сети ЭВМ и передачи данных распределенной АСУ:

АС1 – АС3 - адаптеры связи; АЦП1, АЦП2 – аналого-цифровые преобразователи; Д1 – Д5 – датчики; ДС1, ДС2 – дисплеи; ИМ1 – ИМ3 – исполнительные механизмы; МПД – магистраль передачи данных; ПР1 – ПР3 – процессоры, ПУ – печатающее устройство; ЦАП1 – ЦАП3 – цифро-аналоговые преобразователи



**Рис.5. Радиальная структура сети передачи данных:**  
**а - общий вид; б - звезда; У0 - У5 - узлы**

Если линия передачи данных базовой сети используется для связи только тех двух устройств, которые она непосредственно соединяет, то образуемый ею физический канал связи будем называть каналом парной связи. Пример базовой сети с парными связями показан на рис.5,а.



**Рис.6. Цепная структура сети передачи данных.**  
**а – разомкнутая структура; б – кольцо; У1 - У4 – узлы**

То функциональное устройство узла, через которое данные проходят из узла в сеть или из сети в узел, называется порт о м. Для парной связи должен выделяться порт, поэтому для того чтобы можно было построить сеть с парными связями, необходимо располагать для применения в базовой сети аппаратурой, снабженной несколькими портами.

Одним весьма распространенным видом структуры с парной связью является радиальная структура, или звезда (рис.5,6). В этой структуре один узел, обозначенный У0 - центральный, а остальные  $n$  - периферийные. Периферийным узлам достаточно иметь по одному порту, в то время как центральный должен иметь  $n$  портов.

Иногда специально выделяют структуру, в которой каждый узел соединен каналом парной связи с каждым из остальных узлов сети, - так называемую структуру с полными перекрестными связями.

Частным случаем структуры с парными связями можно считать цепную, или каскадную, структуру, которую иногда выделяют в особый класс. В этой структуре, представленной на рис.6,а, каждый узел, кроме двух

оконечных, снабжен двумя портами. Если данные передаются не между соседними узлами, например от  $U1$  к  $U4$ , то все узлы, лежащие на пути между передающим и приемным, являются т р а н з и т н ы м и: в них возможно промежуточное восстановление передаваемых сигналов.

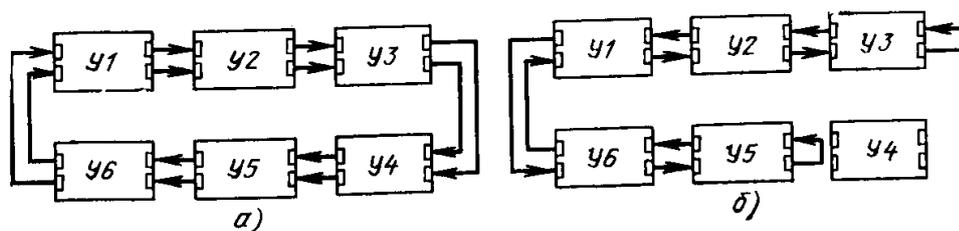


Рис.7. Двойное кольцо:

*а* - исходная структура; *б* - реконфигурация при отказе узла  $U4$ ;  $U1-U6$  - узлы

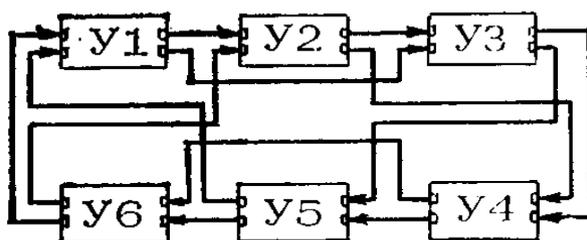


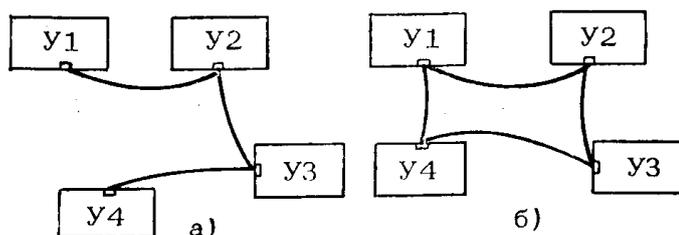
Рис.8. Кольцо с двойным плетением

Весьма часто цепную структуру замыкают в кольцо (рис.6,б). При этом также возможна передача данных в обе стороны, хотя чаще всего она ведется всегда в одном направлении, как это показано стрелками на рис.6,б.

Для повышения надежности сети применяют двойное кольцо (рис.7,а). Каждая пара соседних узлов кольца соединена двумя ветвями - рабочей и резервной. В случае нарушения передачи данных по какой-либо рабочей ветви в рабочее кольцо вводится соответствующая резервная ветвь. Другой способ восстановления двойного кольца, названный с а м о и з л е ч е н и - ем, пригоден в случае отказа аппаратуры узла. Он заключается в том, что данные, дойдя до последнего сохранившего работоспособность узла, направляются обратно по ветвям резервного кольца (рис.7,б). В США выдан патент на кольцо передачи данных специальной структуры, повышающей надежность сети (рис.8). Структура, названная двойным кольцом с плетением, состоит из двух колец: в одном из них ветви последовательно проходят через все узлы, а второе кольцо как бы сплетено из двух ветвей, из которых первая проходит через все нечетные узлы, а вторая - через все четные. Если аппаратура одного из узлов. выходит из строя (например, исчезнет электропитание в данном узле), то узел можно обойти по ветви, не проходящей через отказавший узел.

В отличие от структуры с парными связями, м а г и с т р а л ь н а я структура характеризуется тем, что в ней все узлы соединяются одной непрерывной линией передачи данных, последовательно обходящей все узлы сети (рис.9,а) и соединяющейся с каждым узлом через один его порт. В целях

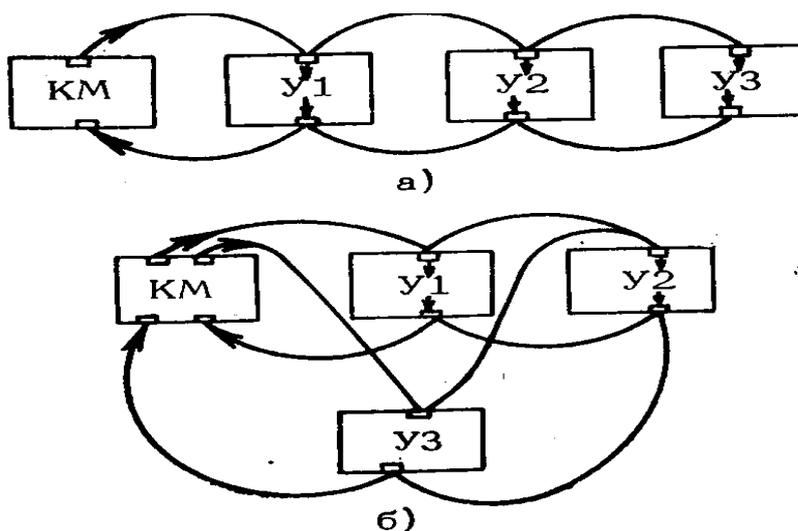
повышения надежности магистраль часто замыкают в кольцо, которое в отличие от кольца цепной структуры будем называть петлей. (рис.9,б). Передача по магистрали ведется в любом направлении.



**Рис.9. Магистральная структура сети передачи данных:**  
*a* - разомкнутая структура; *б* - петля; *У1* - *У4* - узлы

В настоящее время магистральные структуры следует считать перспективно преобладающими в АСУ, что нашло свое выражение в рекомендациях МЭК по магистрали Ргоway.

Для повышения быстродействия иногда применяют расщепление магистрали. Каждый узел имеет по одному входному и по одному выходному порту (рис.10,а). Магистраль расщеплена на две шины; к одной из них, принимающей данные, присоединены все выходные порты узлов, к другой, выдающей данные, - все входные порты. Данные из приемной в передающую шину поступают через контроллер магистрали *КМ*, называемый иногда модификатором шины. Здесь также применяют петлевые структуры (рис.10,б).



**Рис.10. Расщепленная магистраль:**  
*a* – разомкнутая структура; *б* – петля; *КМ* – контроллер магистрали; *У1* – *У3* - узлы

Оригинальна физическая структура сети передачи данных: Строится матрица из магистралей, и каждый узел соединяется с двумя магистралями - горизонтальной и вертикальной (рис.11). Аппаратура узла связана с магистралями интерфейсным блоком, который может соединять между собой подходящие к узлу магистрали, так что цепь передачи данных между парой

узлов может состоять из двух или даже большего числа горизонтальных и вертикальных магистралей.

В последние годы проявляется тенденция к усложнению физической структуры сетей АСУ. Так, например, в предложена физическая структура переменной конфигурации. В зависимости от управления, организуемого сетевым протоколом, отдельные отрезки сети могут представлять собой либо магистраль, либо цепную структуру.

Принцип присоединения аппаратуры станции к каналам связи показан на рис. 1.12. При замкнутом ключе  $K$  отрезок линии связи между станциями  $i - 1$  и  $i+1$  является магистралью, а при разомкнутом - цепью.

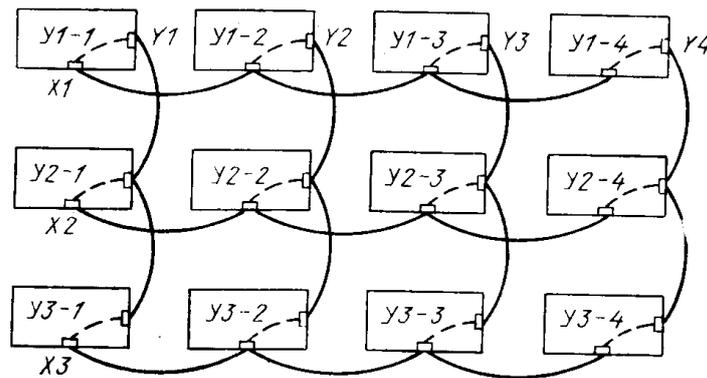


Рис.11. Матрица магистралей:  
У1- У3-4 – узлы

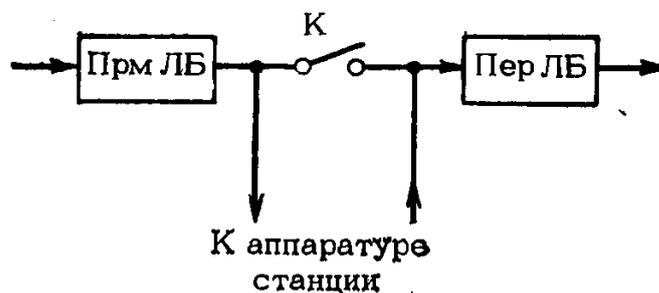


Рис.12. Присоединение станции к сети передачи данных по цепной или магистральной структуре:  
К – ключ; ПерЛБ – передающий линейный блок; ПрмЛБ – приемный линейный блок

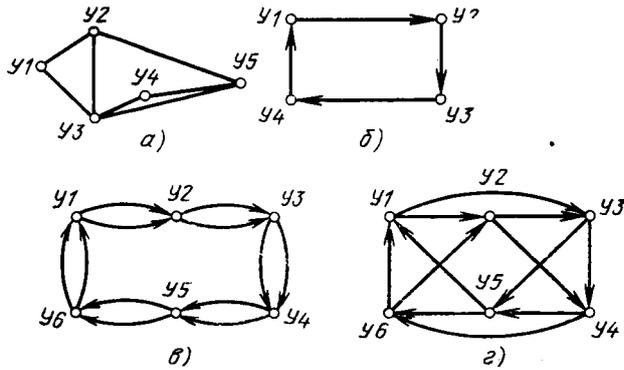


Рис.13. Структурные графы сетей радиальной и цепной структур

Физическую структуру любой системы, состоящей из отдельных устройств, можно описать ориентированным графом, в котором каждому устройству будет соответствовать вершина, а средствам передачи сигналов или воздействий от одного устройства к другому - дуга. Обозначим этот граф как  $G_A = (X, Y)$ , где  $X$  - множество вершин, а  $Y$  - множество дуг, и назовем его **а п п а р т у р н ы м** графом.

Для того чтобы отразить структуру распределенной АСУ, введем другой граф, который назовем **с т р у к т у р н ы м**. В этом графе  $G_s = (V, W)$ , где  $V$  - множество вершин, а  $W$  - дуг, вершина соответствует узлу сети распределенной АСУ, а дуга - ветви между узлами. Очевидно, что граф  $G_s$  может быть получен из  $G_A$  стягиванием в одну вершину всех вершин  $G_A$ , которые отвечают устройствам, находящимся в одном узле. Структурные графы, описывающие структуры на рис.5,а, 6,б, 7,а, 8, приведены на рис.13,а - г соответственно. Ветви направленной связи отображены в структурных графах рис.13 дугами (со стрелками), ветви двусторонней связи - ребрами (без стрелок).

Для отображения структурным графом топологии магистральной структуры в него приходится вводить специальные вершины, соответствующие не узлам сети, а местам при соединения узлов к магистрали. Полученные таким образом структурные графы для сетей на рис9,а, б, 10,б, 11 приведены на рис.14,а - г соответственно.

Для построения распределенных АСУ производители выпускают комплексы аппаратурных средств, оснащаемые широким набором программ, так что вернее говорить о программно-аппаратурных комплексах для построения распределенных АСУ. Некоторые из них описываются в 9 и 10 лекциях. Здесь же рассмотрим физическую структуру распределенных АСУ, реализуемых из современных программно-аппаратурных комплексов, с учетом распределения прикладных функций между узлами.

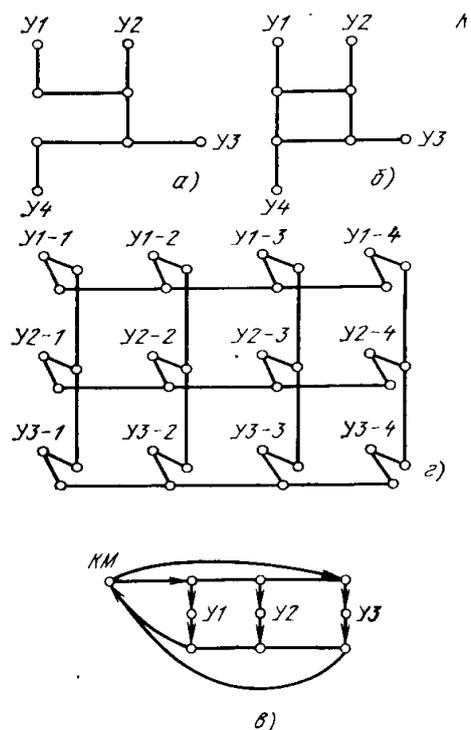


Рис.14. Структурные графы сетей магистральной структуры

Основной единицей аппаратуры современного комплекса является с т а н ц и я - оформленный в единой конструкции набор устройств, присоединяемый к сети АСУ, как правило, с одним сетевым адресом; таким образом, это широко принятое понятие аппаратурной станции согласуется с понятием логической станции, введенным выше. Станции предназначаются для выполнения операций, реализующих прикладную функцию определенного класса.

Типичная физическая структура распределенной АСУ показана на рис.15. Место выполнения операций регулирования и управления, т. е. для функций класса «объект - объект», целесообразно приблизить к процессу; для реализации этих операций служат локальные с т а н ц и и . Эти станции соединяются, с одной стороны, с сетью передачи данных АСУ, а с другой - непосредственно с датчиками и исполнительными механизмами процесса. В локальных технологических станциях выполняется и ряд операций, участвующих в прикладных функциях классов «объект - оператор», «оператор - объект» и «объект - архив» первичная обработка данных о процессе и преобразование управляющего воздействия в форму, воспринимаемую исполнительными органами.

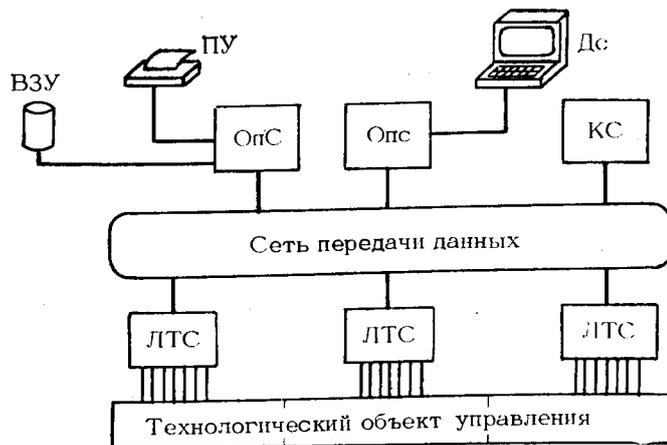
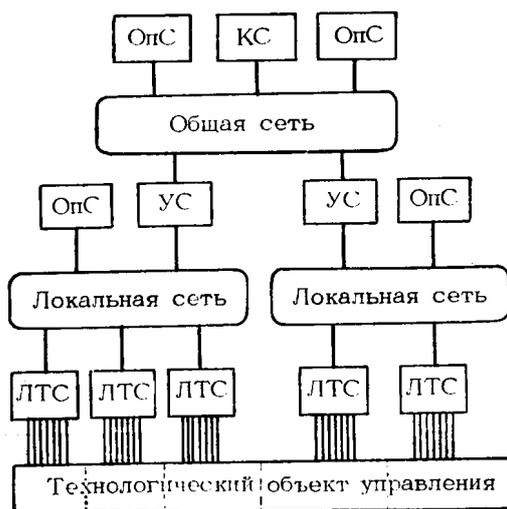


Рис.15.Типичная физическая структура распределенной АСУ:

ВЗУ – внешнее запоминающее устройство; Дс – дисплей; КС – координирующая станция; ЛС – локальная станция; ОпС – операторская станция; ПУ – печатающее устройство

Прикладные функции класса «объект - объект», относящиеся не к какой-либо части, а к процессу в целом, выполняются на станциях, которые размещаются не у объекта; а на операторском пункте-к о о р д и - н и р у ю щ и х с т а н ц и я х КС, эти станции не имеют непосредственной связи с датчиками и исполнительными механизмами, установленными на объекте, а выходят на объект только через сеть и локальные станции.

Операции непосредственного сопряжения с оператором для выполнения прикладных функций «объект - оператор» и «оператор - объект, а именно визуального (а в ближайшем будущем и звукового) вывода и восприятия и преобразования управляющих действий оператора возлагаются на операторские станции ОпС, которые размещаются в операторском пункте. Эти станции выходят на процесс только через сеть передачи данных АСУ и локальные станции. Если архив ведется на том же операторском пункте, то хранящие его внешние запоминающие устройства ВЗУ и печатающие устройства ПУ либо присоединяются к операторской станции, либо подключаются к сети передачи данных АСУ как самостоятельные станции. Дисплей Дс на электронно-лучевых трубках как основное средство визуального вывода с клавиатурой, являющейся основным средством восприятия команд оператора, могут либо конструктивно входить в состав операторской станции, либо присоединяться к ней в виде отдельных устройств.



**Рис.16. Физическая структура двухуровневой сети передачи данных распределенной АСУ:**  
 УС – устройство связи. Остальные устройства см. на рис.15

Средства общения с оператором могут входить и в состав локальных станций; поскольку работа оператора на локальных станциях - явление эпизодическое, эти средства весьма ограничены (несколько индикаторов и небольшая- клавиатура) и иногда присоединяются к станции в виде портативного (ручного) пульта.

Физическая структура распределенной АСУ, представленная на рис.15, характерна для большинства программно-аппаратурных комплексов; различия между ними наблюдаются в физической структуре и (или) протоколах сети передачи данных, в способах программирования или заранее заготовленных программах, в числе технологических переменных, которые обслуживает каждая станция.

Наряду со структурой на рис.15 иногда применяют и другие физические структуры. Некоторые комплексы позволяют строить сети передачи данных двух родов - локальные и общую (рис.16). Первая сеть рассчитана на расстояния примерно 100 м, вторая - на несколько километров. Локальные сети соединяются с общей сетью устройством связи УС, благодаря которому каждая из соединяемых сетей обращается с другой как со своей станцией.

Еще одна физическая структура распределенной АСУ показана на рис.17. Здесь наряду с общей сетью передачи данных, к которой присоединяются все локальные станции, имеются еще локальные сети, соединяющие несколько локальных станций с локальной операторской станцией.

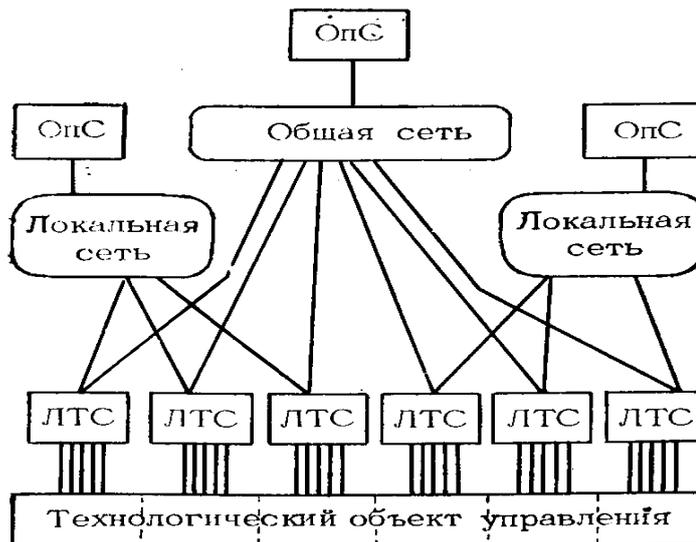


Рис.17. Физическая структура распределенной АСУ с локальными сетями присоединения операторских станций:  
 Обозначения см. на рис.15

## 2. Интерфейсы

Интерфейсом обычно называют воображаемую границу, разделяющую две части системы между которыми осуществляется взаимодействие. В слоистой функциональной структуре под и н т е р ф е й с о м понимают правила, по которым происходит связь между двумя соседними функциональными уровнями.

Задание междууровневых интерфейсов заключается в установлении сопряжения между процессами на двух уровнях. Так, например, интерфейс между физическим и логическим уровнями канала (физическим и канальным уровнями в терминологии АСОС) задает правила перехода от сигналов в линии связи к набору двоичных разрядов в информационном канале и обратного перехода от двоичных разрядов информационного канала к сигналам в линии. Стандартизация делает эти правила едиными внутри одной системы или в целом множестве систем.

Поскольку непосредственная связь между объектами имеет место лишь на физическом уровне, только здесь и применимо понятие интерфейса как границы между однородными частями системы. Процесс связи между объектами внутри одного уровня развивается во времени, ввиду чего необходимо стандартизовать последовательность операций в этом процессе. Набор правил взаимодействия между двумя или несколькими объектами внутри одного уровня системы передачи данных называют п р о т о к о л о м. В протоколах задаются форматы кадров, которыми обмениваются между собой объекты, и последовательность этого обмена во времени.

Стандартные интерфейсы и протоколы разрабатывались в процессе создания вычислительных сетей, начиная с середины 60-х годов, еще до того,

как обозначилось распределение сетевых средств на уровни. Поэтому практически применяемые стандарты на интерфейсы и протоколы, особенно давно утвердившиеся, не ограничиваются одним каким-либо уровнем или сопряжением между соседними уровнями, а нередко захватывают несколько уровней. Первые стандартные интерфейсы относились к местной связи между устройствами в вычислительных системах и распространялись на характеристики уровней физического и информационного каналов, а также частично на сетевой и транспортный уровни. Они предусматривали параллельный обмен данными и сигналами контроля и управления по нескольким десяткам проводов. Стандарты устанавливали тип линии связи, способ подключения к ней, назначение отдельных проводов, уровни сигнала, коды, способы синхронизации, порядок получения доступа к общей линии связи, способы выбора, адреса, задания начала и окончания передачи.

Интерфейсы, шин, соединяющих отдельные платы и узлы внутри устройств ЭВМ,- так называемые в н у т р е н н и е интерфейсы - подробно рассмотрены в литературе. Здесь упомянем лишь в качестве примеров «общую шину», применяемую в различных модификациях в микро-ЭВМ «Электроника-60» и мини-ЭВМ СМ-3 и СМ-4, 100-проводную шину S100, используемую в таких распространенных микропроцессорах американских фирм, как 8080 фирмы INTEL, 280 фирмы Zilog, 6800 фирмы Motorola и 6502 фирмы Mostech, а также более экономные по затратам проводов 50-проводную шину S50 и Multibus. Более широкой стандартизации требуют интерфейсы связи между устройствами в локальных вычислительных системах. Стандартизация этого интерфейса, обычно называемая в н е ш - н и м, позволяет объединять в одной системе устройства, выпускаемые разными фирмами.

В настоящее время связь устройств реализуется по нескольким наиболее распространенным стандартным интерфейсам; часть из этих стандартов имеет международный характер. Их преимущественная область действия - уровень физического канала и сопряжения физического и информационного каналов.

Одним из наиболее ранних и весьма распространенным стандартом на внешний интерфейс является RS232-C, введенный в 1969 г. Ассоциацией электронной промышленности (EIA) в США. Он предусматривает последовательный обмен между устройствами на расстоянии, не превышающем 15 м, со скоростью до 20 кбит/с. Последующие стандарты той же ассоциации RS422 и RS423 увеличивают дальность связи до 1200 м, а скорость передачи - до 10 Мбит/с.

Интерфейс V.24, рекомендованный Международным консультативным комитетом по телеграфии и телефонии (МККТТ, по-английски ССИТТ), относится к передаче данных по общественным сетям с применением модемов и распространяется на каналы связи между терминальными устройствами для данных и аппаратурой передачи данных.

К наиболее распространенным международным стандартам этого рода относятся RS422/423, рекомендованный МККТТ после его введения EIA, и интерфейс по стандарту TC66 (02) Международной электротехнической

комиссии (МЭК), впервые введенный Институтом по электротехнике и электронике (IEEE) в США под названием IEEE-488-75.

Радиальный интерфейс 2К, предназначенный для связи между ЭВМ и ее периферийными устройствами, нашел широкое применение в распространенных ЭВМ типов М6000, М7000, СМ1, СМ2. Шина 2К содержит 50 проводов, из которых 26 служат для передачи сигналов от ЭВМ к периферийным устройствам, а 23 - для передачи в обратном направлении.

Интерфейс между физическим и канальным уровнями для синхронной передачи данных по дискретным каналам связи (передача видеоимпульсами) общественных сетей стандартизован в рекомендациях X.21 МККТТ. Модификация этих рекомендаций X.21 бис относится к передаче по телефонным каналам с модемами и совместима с интерфейсом V.24 МККТТ.

В бывшем СССР стандартизован интерфейс СЗ, рассчитанный на быструю параллельную передачу данных по радиальным шинам, содержащим до 40 проводов. В дальнейшем этот стандарт был распространен на страны СЭВ под названием ИРПР (интерфейс радиальный параллельный).

### 3. Протоколы

С развитием систем передачи данных постепенно стандартизовались протоколы канального уровня. Сначала эти стандарты вводились внутри фирм или национальных организаций; таким путем возникли протоколы ADCCP (Национальный институт стандартов США ANSI), DDCMP (фирма Digital Equipment Corp., США), SDLC (фирма IBM). В 1973 г. Международная организация стандартов (МОС, по-английски ISO) одобрила протокол канального уровня HDLC, весьма сходный с SDLC. Впоследствии протокол HDLC был утвержден как стандарт МККТТ для общественных вычислительных сетей. Многие фирмы стремятся следовать этому протоколу и в своих разработках локальных сетей, и он получает все более широкое распространение. Требования чрезвычайно высокой достоверности передачи данных в АСУ (рекомендация стандарта Proway - вероятность ошибочного бита в данных  $3 \cdot 10^{-15}$ ) находят отражение в стандартизации способа контроля достоверности принятого кадра на уровне канального протокола.

В АСУ преобладает магистральная структура сетей передачи данных, в связи с чем развиваются стандартные интерфейсы и протоколы, относящиеся к магистралям. Они охватывают обычно три нижних уровня архитектуры соединения открытых систем - от физического до сетевого.

Первым стандартом для магистральной сети следует считать интерфейс САМАС-вертикаль, рекомендованный ESONE (Европейским комитетом стандартов по ядерной электронике) для связи между каркасами (или, как их называют в системе САМАС, крейтами) и предусматривающий применение 14 радиальных и 49 магистральных проводов.

К магистральным относятся многочисленные интерфейсы вычислительных систем, в частности, ЭВМ СМ-3, СМ-4, ЕС ЭВМ, М4030, «Электроника-60».

Интерфейс ввода-вывода ЕС ЭВМ построен на 34 проводах, а интерфейс "Общая шина", применяемый в СМ-3, СМ-4,- на 56 проводах.

Во всех интерфейсах магистрали устанавливаются характеристики сопряжения между физическим и канальным и между канальным и сетевым уровнями, а также протоколы канального и сетевого уровней. Канальный протокол, как правило, основывается на стандарте HDLC.

Для стран бывшей СЭВ был подготовлен стандарт ИЛПС, где доступ к магистрали осуществлялся по протоколу с передачей маркера под управлением центральной станции или без централизованного управления.

Если в первых магистралях АСУ применялись простые протоколы с циклическим представлением доступа к магистрали в фиксированные интервалы времени, то затем требования быстрой реакции системы в реальном времени заставили перейти к более сложным магистральным протоколам, в которых доступ предоставляется станции по запросу в зависимости от присвоенных постоянно или в данный момент приоритетных прав. Упомянутый стандарт Proway вводит протокол магистрали, по которому станциям присваиваются постоянные ранги (имеется шесть различных рангов). Станции старшего ранга могут присваивать станциям младшего ранга статус, позволяющий им распоряжаться магистралью. В последние годы в США получает широкое распространение в локальных вычислительных сетях, в том числе и в распределенных АСУ, протокол полностью децентрализованного управления, по которому все станции равноправны, а конфликты разрешаются повторением неудавшихся передач.

В последние годы локальные вычислительные сети, предназначенные для автоматизации учреждений, стали получать столь широкое распространение, что далеко обогнали другие области применения, в том числе и АСУ. В связи с подобным развитием ведущие фирмы США, действующие в этой области, Хегох, DEC и Intel, объединившись, приступили к подготовке стандарта IEEE 802 на магистраль, в котором предусматриваются два децентрализованных протокола доступа: с контролем несущей и разрешением конфликтов, когда каждая станция может начать передачу, коль скоро магистраль свободна, и с циклической передачей маркера, означающего право пользования магистралью.

Основные идеи, содержащиеся в проекте IEEE 802, были использованы в стандартах, принятых ЕСМА (Европейской ассоциацией производителей вычислительной техники) в конце 1982 г. К этим стандартам уже присоединился ряд фирм США, ведущих в области вычислительной техники, в том числе и фирмы - инициаторы разработки IEEE 802. Архитектура локальной сети по стандартам ЕСМА соответствует АСОС (рис.1,а).

В начале 1984 г. была предложена новая версия магистрали Proway, названная Proway C. В этой версии архитектурное расслоение совпадает с проектом IEEE 802, основным положением которого предложенная новая версия следует. Протоколы сетевого и транспортного уровня для сетей АСУ еще далеки от стандартизации. Они предусматривают возможность связи нескольких магистралей и организацию передачи данных в кольцевых физических структурах. Транспортный протокол в большинстве

распределенных АСУ совмещается с сетевым. Структурирование протоколов надсетевых уровней т.е. трех верхних уровней АСОС, в АСУ практически нигде не реализуется. Операционные системы АСУ независимо от их физической структуры если и структурируются, то по принципам отнюдь не совпадающим с требованиями АСОС. Заметим, что слоевая структура надсетевых уровней Proway; в общем близкая к АСОС, еще не оформлена в виде каких либо требований.

### Контрольные вопросы

1. Как образуется физическая структура распределенной АСУ? [Л.7, 20-21].
2. Начертите сети ЭВМ и передачи данных распределенной АСУ. [Л.7, 22-23].
3. Приведите пример радиальной структуры базовой сети с парными связями передачи данных. [Л.7, 23-24].
4. Расскажите о способе восстановления двойного кольца «самоизлечением». [Л.7, 24-25].
5. Какие структуры в настоящее время следует считать перспективно преобладающими в АСУ? [Л.7, 25-26].
6. Для какой цели применяют расщепление магистрали? [Л.7, 26-27].
7. Как можно описать физическую структуру любой системы? [Л.7, 27-28].
8. Нарисуйте типичную физическую структуру распределенной АСУ. [Л.7, 29-30].
9. Что такое интерфейс? [Л.7, 30-32].
10. Протоколы канального уровня. [Л.7, 33-35].

При этом требуется соблюдать определенные временные ограничения. Мероприятия по их реализации называют с и х р о н и з а ц и е й. Они сводятся к тому, что процессы поочередно переводятся в одно из трех состояний: выполнения, готовности (ожидания выделения времени процессора) и блокировки (ожидания разрешающего события, например завершения операции ввода-вывода). В тех случаях, когда состав программ меняется, необходимы дополнительные операции образования новых процессов (объектов, управляемых ОС) и уничтожения (удаления из ОС) существующих процессов. Эта работа составляет функцию управления процессами. Она дает пользователю возможность описать алгоритм функционирования АСУ в виде совокупности любого числа взаимосвязанных программ, выполняемых в разное время и при разных условиях. Машина  $M_1$  выполняет также функцию ввода-вывода, однако в отличие от машины  $M_0$ , которая реализует только одну операцию, машина  $M_1$  возвращает управление вызывающему процессу только после выполнения всех операций, требуемых в обрабатываемом запросе. В функции машины  $M_1$  входит обработка исключительных состояний в тех случаях, когда они выходят за рамки возможностей машины  $M_0$ .

Третий уровень иерархии - псевдомашину M2 операционной системы - составляют драйверы. Драйвером принято называть часть ОС, предназначенную для управления аппаратным средством и реализующую интерфейс с указанным средством, единый для различных функциональных блоков. В общем случае драйвер выполняет следующие функции: форматирование передаваемых данных в соответствии с выбранным протоколом; управление аппаратным средством (включение, выключение, поиск сектора на диске, протяжку бумажной ленты и т. п.); синхронизацию процессов, участвующих в передаче данных, между собой и с вызывающей программой; обработку исключительных состояний и ошибок передачи.

Драйвер выполняет свои функции, пользуясь услугами машин M0, M1. Новой функцией на этом уровне является лишь форматирование данных. В зависимости от вида аппаратуры различают драйверы стандартных устройств (например, ввода-вывода), специальных устройств (например, связи с объектом управления), межмашинного обмена и т. п.

Дальнейшее развитие функции обработки данных ведет к системе управления памятью, за которой логически следуют системы управления файлами и базой данных. Эти системы можно рассматривать как следующие псевдомшины (см. рис.45). Однако реализация этих псевдомшин в распределенной системе требует дальнейшего развития функции управления процессами. При динамическом распределении заданий здесь возникает качественно новая проблема динамической обработки прерываний, не известная в однопроцессорных системах.

Другая качественно новая проблема, возникающая в распределенной системе, связана с обработкой исключительных состояний (далее просто исключений). Исключение есть отклонение от нормального хода процесса. Хотя оно и возникает вследствие ошибок аппаратуры или программ, само исключение является ожидаемым состоянием и поэтому распространенный термин «обработка ошибок» здесь заменен.

При обнаружении исключения ОС должна запустить соответствующий процесс обработки.

В функцию псевдомшины M4 входит динамическая обработка прерываний и исключений. На этом уровне решаются проблемы планирования заданий, так что машина M4 в целом реализует функцию динамического распределения заданий в распределенной системе. Наличие этого функционального слоя позволяет реализовать распределенные банки данных реального времени (псевдомшина M<sub>0</sub>). Файловая система (машина M5) реализуется обычно в однопроцессорном варианте, так как внешняя память используется в распределенных АСУ лишь в операторских станциях. В ОС обычно выделяют универсальное ядро, охватывающее функции, необходимые практически при любом применении, и высшие уровни, к которым относят функции, повышающие комфорт пользователя, но не являющиеся строго необходимыми.

Основное назначение ядра - поддержка одновременно (параллельно) исполняемых процессов. Оно обеспечивает синхронизацию процессов,

управление процессами и управление устройствами ввода-вывода и передачи данных, а также обработку исключительных состояний. Ядро ОС распределенной конфигурации содержит, таким образом, псевдомашину M0 - M2. К высшим уровням ОС относят функции планирования процессов, управления памятью и файлами, управления базой данных и т. п. Эти функции реализуются машинами M3 - M7. В распределенных системах с жестко связанными процессорами принципиально допустима как полная децентрализация, так и полная централизация ОС. В первом случае все станции располагают одинаковыми ОС, реализующими всю сумму функций, указанных на рис.45. Во втором случае в составе системы выделяют единый компонент, централизованно управляющий всеми ресурсами. В многомашинных конфигурациях, где отсутствует общая память, централизация функций ОС принципиально невозможна.

Статическое распределение достигается дублированием компонентов ОС в станциях. Программы, реализующие функции псевдомашин M4 - M7, обычно закрепляют за операторскими станциями, а программы ядра ОС дублируют на каждой локальной технологической станции. Для примера на рис.46 показаны функции ОС в распределенной конфигурации на средствах программно-аппаратурного комплекса RDC. Операционная система комплекса содержит центральную часть, часть, обслуживающую пульт оператора, и локальные части (далее каждая часть считается отдельной ОС). За центральной ОС закреплены лишь безусловно необходимые функции начального запуска, а также подготовки (трансляции, компоновки и т. п.) и отладки программ, необязательные функции учета и т. п. Эти функции выполняются, естественно с использованием компонентов ядра (локальной ОС). Локальная ОС выполняет на уровне своего региона функции синхронизации, управления процессами, вводом-выводом, обработки исключений (машины M0 - M2). На сетевом и транспортном уровнях, т. е. на уровне системы в целом, она управляет прикладными процессами через общую магистраль, поддерживая передачу данных и программ по магистрали, а также обработку сообщений о статусе системы в целом (машина M3). Таким образом, локальные ОС несут основную нагрузку по организации функционирования АСУ в реальном времени.



## **Контрольные вопросы**

1. Что понимают под операционной системой? [Л.7, 127-128].
2. Проиллюстрируйте место операционной системы в АСУ . [Л.7, 128-130].
3. Роль процессоров и запоминающих устройств при построении операционных систем. [Л.7, 130-131].
4. Начертите схему расположения запоминающих устройств в распределенной АСУ. [Л.7, 132-134].
5. Расскажите о статическом и динамическом распределении ресурсов. [Л.7, 135-137].
6. Принципы описания функций операционной системы, положенные в основу архитектуры открытых систем. [Л.7, 137-138].
7. Что составляет логическую структуру АСУ ? [Л.7, 137-138].
8. Какой метод используется для представления типичных функций ОС? [Л.7, 138-139].
9. Какая работа входит в функцию управления процессами? [Л.7, 138-139].
10. Приведите пример для распределения функций операционной системы в АСУ . [Л.7, 139-140].

## **Лекция №9**

### **Проектирование распределенных АСУ. Модели распределенных АСУ**

#### **План**

1. Проектирование распределенных АСУ.
2. Математические модели распределенных АСУ.
3. Понятия структурного поля и поля реализации для операций цифровой обработки.

#### **1. Проектирование распределенных АСУ**

Выше была рассмотрена архитектура распределенных АСУ, и представляется естественным связать принципы проектирования распределенных АСУ с их архитектурой. Проектирование можно трактовать как выбор ресурсов АСУ и распределение их между пользователями. При архитектурном подходе, изложенном в 1 и 2 лекциях, ресурсы заключены в каждом слое архитектуры, и этими ресурсами пользуется для выполнения своих функций ближайший сверху слой. Наряду с функциональными ресурсами, размещающимися в слоях архитектуры, следует рассматривать также аппаратные ресурсы, закрепленные за одним или сразу несколькими слоями архитектуры. Таким образом, к выбираемым и распределяемым ресурсам относятся аппаратура обработки данных (процессоры), аппаратура связи, а также программы, реализующие обработку и передачу данных. Эти ресурсы используются для выполнения функций практически на всех уровнях, начиная с обработки данных на прикладном уровне и кончая реализацией канальных протоколов.

В то время как программы легко структурируются и, следовательно, без труда могут быть расслоены по архитектурным уровням, иначе обстоит дело с аппаратурой. Программная реализация широкого круга функций АСУ приводит к тому, что аппаратные ресурсы процессоров не могут быть закреплены за одним каким-либо слоем архитектуры, а могут перераспределяться также между слоями. Лишь отдельные устройства, реализующие микропрограммным или чисто аппаратным путем передачу данных или некоторые операции обработки, оказываются закрепленными за определенным слоем, главным образом, слоем управления физическим или информационным каналом.

Хотя архитектура АСУ предусматривает слоевую иерархию пользования ресурсами, на любом слое можно, исключив управляющие операции общесистемного назначения, выделить действия, которые служат решению какой-либо прикладной задачи, так что прикладные функции являются конечными пользователями услуг, оказываемых ресурсами АСУ. Поэтому можно, опуская промежуточных пользователей, говорить, когда это удобно, о распределении ресурсов между прикладными функциями АСУ.

Ресурсы могут распределяться по статическому или динамическому принципу. В первом случае уже при проектировании ресурс закрепляют за определенной прикладной функцией АСУ. Это не обязательно означает, что данный ресурс выделен специально для данной прикладной функции, хотя и такое положение может иметь место.

Динамическое распределение ресурсов между прикладными функциями осуществляется в процессе функционирования АСУ в зависимости от текущего состояния системы, прежде всего от загрузки ресурсов и их готовности, обусловленной надежностью и быстродействием аппаратуры. Тогда задачей проектирования является установление протоколов, распределяющих ресурсы. Разумеется, протоколы в общем случае распределяют ресурсы между пользователями не произвольно, а в определенных границах, которые могут

частично устанавливаться при проектировании. Такие границы устанавливает, например, топология связи между узлами в сети АСУ. Изложенные соображения свидетельствуют о том, что четкого различия между статическим и динамическим распределением ресурсов не существует.

Чтобы количественно оценить влияние загрузки ресурсов АСУ и надежности аппаратуры, ниже выводятся соответствующие соотношения, которые носят, впрочем, приближенный характер и применимы при определенных предпосылках. В п.3 рассматриваются методы оптимального проектирования вычислительных сетей, которые могут быть распространены на проектирование распределенных АСУ.

## 2. Математические модели распределенных АСУ

Описание предлагаемой модели распределенной АСУ облегчается, если расщепить прикладной уровень архитектуры АСУ по два, выделив внизу подслою способов исполнения прикладных функций.

Пусть АСУ состоит из  $s$  устройств и реализует  $|u|$  функций. В некоторых случаях одна и та же функция может выполняться различными наборами операций, и тогда будем говорить о режимах выполнения функций. Пусть всего имеется  $q$  режимов выполнения всех  $|u|$  функций, эффективность каждого режима определяется величиной  $E_i$ ,  $i=1, \dots, q$  (выражаемой, например, в суммах в единицу времени). Величина  $E_i$  называется частной, или условной, эффективностью АСУ.

Выделим среди режимов такие, которые характеризуются различными значениями частной эффективности  $E_i$  и будем говорить, что подобные режимы обеспечивают неэквивалентном резервировании с функции. В связи с режимами, имеющими одинаковую частную эффективность, будем говорить об эквивалентном резервировании и рассмотрим их в целях упрощения подхода отдельно, поскольку они не затрагивают структуры функционального графа.

Неэквивалентное резервирование может найти отражение в функциональном графе АСУ. Введем для каждого режима реализации функции свои пометки терминальных вершин и, следовательно, свои пути между терминальными вершинами одной функции. В общем случае терминальные вершины у разных режимов реализации одной функции могут частично или полностью совпадать, но пути между ними обязательно должны различаться для каждого способа. Таким образом, каждому режиму  $f$  в функциональном графе  $G_f$  учитывающем неэквивалентное резервирование, соответствует свой подграф  $G_{fi}$ .

Вернемся к примеру распределенной АСУ на рис. 2 и 4. Пусть в ней предусматривается неэквивалентное резервирование функций по режимам, представленное в табл. 1. Первым режимом реализации каждой функции является основной режим, совпадающий с содержанием функции, описанным в 1-ой лекции, поэтому его содержание в табл. 1 не повторяется.

Таблица 1.

Функция	Режимы	Содержание режима
1	1'	Измерение $A$ не корректируется по $B$
2	2'	Измерение $A$ не корректируется по $B$
3	3'	Измерение $A$ не корректируется по $B$
4	4'	Измерение $A$ не корректируется по $B$
5	5'	Ручное управление с местного пункта $П2$
7	7'	Вывод $L$ на печать (вместо вывода на дисплей)
	7''	Измерение $A$ не корректируется по $B$
	7'''	Вывод $L$ на печать вместо дисплея и измерение $A$ не корректируется по $B$
8	8'	Измерение $A$ не корректируется по $B$

Функциональный граф  $G'_f$ , учитывающий неэквивалентное резервирование в АСУ, показан на рис.47, где обозначения режимов при терминальных вершинах соответствуют табл.1. Новая по сравнению с рис.3 вершина  $Q$  относится к новой операции вывода показателя  $I$  на печать.

Каждая функция выполняется в процессе работы АСУ в каждый данный момент времени в одном каком-либо режиме. Для нескольких режимов исполнения одной функции тем или иным образом задается порядок их предпочтения, или приоритета. Режим с низшим приоритетом применяется только тогда, когда режим с более высоким приоритетом имеет нулевую готовность, т. е. не может быть реализован из-за отсутствия какого-либо ресурса, требующегося для его исполнения.

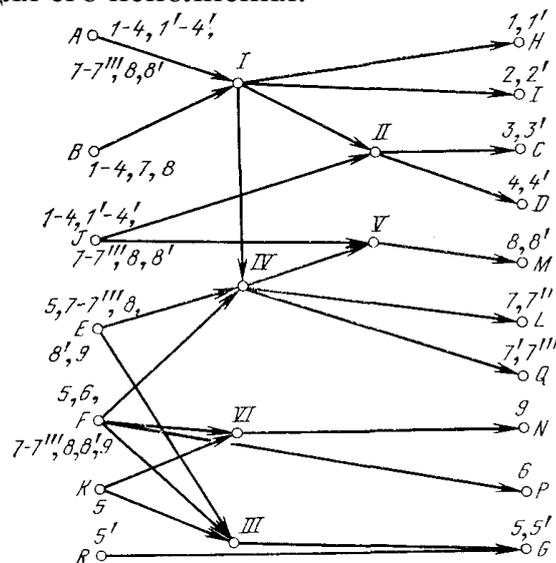


Рис. 47. функциональный граф АСУ рис. 3 с учётом неэквивалентного резервирования

Протокол, в соответствии с которым подслои прикладных функций пользуется услугами подслоя режимов их исполнения, можно описать квадратной булевой матрицей  $\Psi$  размерностью  $q \times q$ . Элемент  $\Psi_{ij}$ , этой матрицы

равен 1, если режим  $i$  резервирует режим  $j$  (т. е. режим  $j$  имеет приоритет выше, чем  $i$ ), и нулю в противном случае, в частности при  $i=j$ , а также тогда, когда режимы  $i$  и  $j$  относятся к разным функциям.

При построении матрицы приоритетов  $\Psi$  целесообразно ее нормализовать, или привести. Будем считать матрицу  $\Psi$  приводимой, если ее строки (а следовательно, и столбцы) можно перенумеровать в порядке убывания приоритетов при группировке их по функциям, так что режимы исполнения одной функции имеют номера строк (столбцов), идущие подряд. Положим, что матрица  $\Psi$  рассматриваемых нами АСУ приводима, что практически всегда имеет место. Приведенная матрица  $\Psi$  оказывается блочной, в ней каждой функции  $l$  соответствует блок, строки и столбцы которого относятся к режимам выполнения одной и той же функции  $l$ . Кроме того, каждый блок представляет собой треугольную подматрицу с единицами ниже главной диагонали и под ней.

Для АСУ рассматриваемого нами примера приведенная матрица приоритетов строится исходя из того, что режимы исполнения каждой функции уже упорядочены по убыванию их приоритета. Приведенная матрица  $\Psi$  размерностью  $18 \times 18$  имеет следующий вид:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Матрица состоит из девяти блоков, соответствующих девяти функциям, со строками и столбцами: (1,2); (3,4); (5,6); (7,8); (9,10); (11);(12, 13, 14, 15); (16, 17); (18).

Матрица приоритетов  $\Psi$  может фиксироваться на стадии проектирования, но может также и строиться протоколом прикладного уровня в зависимости от текущего состояния ресурсов, которыми пользуется прикладной слой.

Рассмотрим, как пользуются ресурсами режимы исполнения прикладных функций. Имея в виду невозможность расслоения аппаратных ресурсов,

непосредственно опишем применение отдельных ресурсов при реализации каждого режима.

Участие ресурсов в режиме выполнения функции можно описать булевой матрицей  $\Phi$  размерности  $q \times s$ , элемент которой  $\varphi_{ik}$  равен 1, если  $k$ -ый ресурс участвует в  $i$ -м режиме, и 0 в противном случае. Если ограничиться рассмотрением только аппаратных ресурсов, считая ресурсом каждое устройство, то для АСУ, построенной согласно схеме рис. 1.4, с неэквивалентным резервированием согласно табл.1 матрица  $\Phi$  будет иметь размерность  $18 \times 34$ .

Номера устройств, выступающих как ресурсы в матрице  $\Phi$ , указаны на рис.4.

Учтем то обстоятельство, что ресурс не всегда готов к применению: либо из-за отказа, либо из-за занятости его другой прикладной функцией. Введем булев вектор  $h$ , содержащий  $s$  компонентов и характеризующий текущую готовность ресурса к работе:  $k$ -й компонент равен 1, если  $k$ -й ресурс готов в данный момент к работе, и 0, если он занят или отказал. Тогда готовность режимов, характеризуемая булевым вектором  $f$  (компонент  $f_i$  равен 1, если режим работоспособен), находится как

$$f = \Phi h \# . \quad (13)$$

где  $\Phi$  - матрица с элементами, являющимися отрицаниями элементов  $\Phi$  (нулю в  $\Phi$  соответствует единица в  $\Phi$ , а единице в  $\Phi$  – нуль в  $\Phi$ ). Значок  $\#$  показывает, что умножение булевой матрицы  $\Phi$  на булев вектор  $h$  выполняется таким образом, что

$$f_i = \left( \bigcap_{k=1}^s (\overline{\varphi_{ik}} \vee h_k) \right) \quad (5.2)$$

т. е. компонент вектора  $f$  образуется как конъюнкция дизъюнкций элементов строки матрицы  $\Phi$  и компонентов вектора  $h$ .

Фактически выполняемый АСУ набор режимов для текущего момента времени определяется с учетом готовности ресурсов как

$$g = f \wedge \hat{f} , \quad (5.3)$$

где  $\hat{f}$  - булев вектор отсутствия готовности режимов, имеющих приоритет, т. е.  $f_i=1$ , если все режимы  $f_j$  с приоритетом старше, чем у режима  $f_i$  не могут исполняться, и  $f_i=0$  в противном случае. Компоненты вектора  $\hat{f}$  находятся как

$$\hat{f}_i = \bigvee_{j=1}^q (\psi_{ij} \wedge \overline{\hat{f}_j}) . \quad (16)$$

Они равны единице тогда, когда для каждого режима  $j$  соблюдается одно из двух условий: он либо не имеет приоритета перед режимом  $i$ , либо если имеет такой приоритет, то не работоспособен. Таким образом, компонент  $g_i$  вектора  $g$  согласно (15) равен единице тогда и только тогда, когда одновременно удовлетворяются два условия: режим  $f_i$  может исполняться по состоянию готовности ресурсов и все режимы  $f_j$ , имеющие приоритет перед  $f_i$ , по состоянию готовности ресурсов не могут исполняться.

Текущее значение эффективности  $E(t)$  функционирования АСУ зависит не только от ее состояния в данный момент  $t$ , но и от ее прошлого. Однако здесь в целях упрощения анализа допустим, что такая зависимость отсутствует. Подчеркнем, что подобное допущение означает отказ от учета таких последствий отказов АСУ, как остановки и аварии. При введенном допущении средние значения частной эффективности каждой функции АСУ за некоторый отрезок времени оказываются пропорциональными времени ее выполнения, или ее готовности.

Текущее значение эффективности  $E$  функционирования АСУ зависит от набора исполняемых режимов:

$$E = E(g) \quad (17)$$

С учетом (13) и (15) текущая эффективность АСУ в зависимости от готовности ресурсов определяется как

$$E = E(f \wedge \Psi f^*) = E(\bar{\Phi} h \# \wedge \Psi(\bar{\Phi} h \#)^*), \quad (18)$$

где значок  $*$  указывает на то, что умножение булевой матрицы  $\Psi$  на булев вектор  $f$  выполняется так, что компонент вектора произведения образуется путем дизъюнкции конъюнкций элементов строки матрицы  $\Psi$  и компонентов вектора  $f$ , т. е.

$$(\Psi f^*)_i = \bigcup_{j=1}^n (\psi_{ij} \wedge f_j). \quad (19)$$

Соотношение (18) может служить основой для построения модели, описывающей зависимость текущей эффективности АСУ от ее надежности.

Зависимость (17) в силу того, что аргумент есть булева переменная, не поддается аналитическому описанию и может быть задана только таблично, причем таблица должна иметь до двух аргументов.

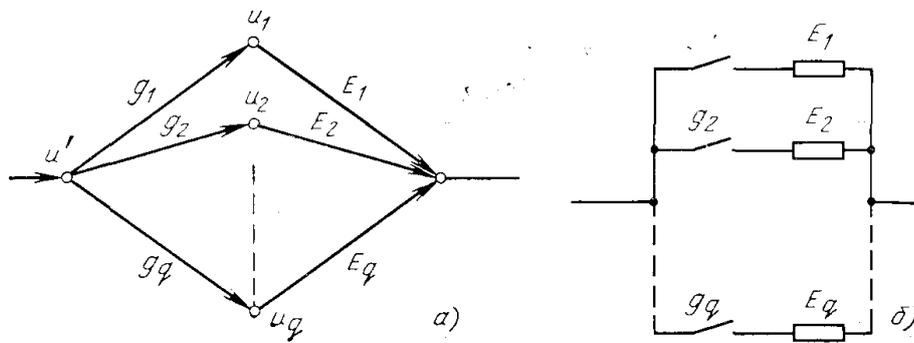


Рис.48. Модель эффективности АСУ:  
 в — потоковая сеть; б — электрическая схема

Для сколько-нибудь сложной АСУ (например, при  $(q > 5)$ ) составление такой таблицы становится практически неосуществимым. Поэтому для исследования влияния надежности на эффективность АСУ приходится прибегать к упрощающим допущениям. Одним из таких допущений может быть аддитивность зависимости  $E(g)$ , приводимой тогда к виду

$$E = \sum E_i g_i = E g, \quad (20)$$

где  $g_i$  - значение  $i$ -го компонента вектора  $g$ , равного 1 или 0, а  $E$  - вектор эффективности режимов выполнения функций АСУ с компонентами  $E_i$ . В этом случае эффективность в зависимости от готовности ресурсов выражается линейным алгебраическим соотношением

$$E = E g = E (\bar{\Phi} h \# \wedge \Psi (\Phi h \#)^* ). \quad (21)$$

Модель, описываемая соотношением (21), может быть наглядно представлена потоковой сетью специального вида, показанной на рис.48,а. Эта сеть состоит из  $q$  дуг  $(u', u_i)$ ,  $i=1, \dots, q$ , с пропускной способностью  $\beta_i$ , равной нулю при  $g_i=0$  и бесконечности при  $g_i=1$ , и  $q$  дуг  $(u_i, u'')$ ,  $i=1, \dots, q$ , с пропускной способностью  $E_i$ . Значения  $g$  образуются как логические функции вектора  $h$ , поэтому может оказаться удобным при графическом представлении заменить дуги  $(u', u_i)$  логической схемой (рис.48,б), реализующей преобразование  $g(h)$  над входным сигналом по соотношению

$$g = \bar{\Phi} h \wedge \Psi (\Phi h)^* \quad (22)$$

Приведенное выше матричное описание не всегда практически удобно, поскольку несмотря на предложенное выше сокращение размерности матриц  $\Psi$  и  $\Phi$  они хотя и получаются слабо заполненными, но все еще имеют большое

число строк и столбцов. Поэтому может оказаться полезным, особенно при машинном моделировании распределенных АСУ, описание структуры и резервирования с помощью списков, тем более что таким образом можно учитывать также эквивалентное резервирование, пока исключенное из рассмотрения.

Матрица ресурсов  $\Phi$  в списочном представлении заменяется списками

$$A_i = \{a_k\}, \quad (23)$$

в которые для режима  $i$  вводятся идентификаторы используемых ресурсов - номера тех столбцов, которые содержат единицу в  $i$ -й строке. Списки ресурсов по режимам в примере АСУ на рис.4:

$$A_1 = (1, 2, 5, 6, 7, 8, 9, 28, 30, 31, 32);$$

$$A_2 = (1, 2, 7, 8, 9, 28, 30, 31, 32);$$

$$A_3 = (1, 2, 5, 6, 7, 8, 9, 29, 30, 31, 32)$$

и т. д. в соответствии с матрицей  $\Phi$ .

Вместо матрицы приоритетов  $\Psi$  можно ввести списки режимов исполнения функций в предпочтительном порядке их применения

$$F_l = \{i\}_l, \quad (24)$$

где  $l$  - номер функции, а  $i$  - номер режима, резервирующего основной (первый) режим исполнения  $l$ -й функции. В нашем примере

$$F_1=(1,2); F_2=(3,4); F_3=(5,6); F_4=(7,8); F_5=(9,10);$$

$$F_6=(11); F_7=(12, 13, 14, 15); F_8=(16,17); F_9=(18).$$

Наконец, списками удобно описывать эквивалентное резервирование, не отражаемое матрицами  $\Phi$  и  $\Psi$ .

Список эквивалентных резервирующих ресурсов надо составлять для каждого ресурса с учетом функции, в которой он должен быть резервирован, и даже с учетом того, какие из ресурсов АСУ уже не готовы к использованию к этому моменту времени. Однако столь сложные дисциплины резервирования обычно трудно реализовать, и чаще всего резерв вводится независимо от того, какая функция выполняется АСУ в данный момент.

Необходимо сделать одно уточнение: если один и тот же ресурс выполняет принципиально различные операции, то он может резервироваться по каждой такой операции различными ресурсами. Так, печатающее устройство с клавиатурой выполняет операции ввода и вывода информации: по вводу его

можно резервировать другой клавиатурой, а по выводу - устройством цифровой индикации, в частности дисплеем. В связи с принципиально различными операциями, реализуемыми некоторыми устройствами, целесообразно такие устройства в списках  $A_i$  разбивать по числу операций. В примере на рис.4 к устройствам этого рода относятся дисплеи 27 и 28 и печатающее устройство 29: каждое устройство выполняет операции цифрового ввода и вывода. Сохранив за операциями ввода прежние номера 27, 28 и 29 введем новые номера 35 - 37 для операций вывода, выполняемых этими устройствами, тогда списки  $A_i$ , перепишутся в виде

$$A_1=(1,2,5,6,7,8,9,30,31,32,35);$$

$$A_2=(1,2, 7,8, 9,30,31,32, 36);$$

$$A_2=(1, 2, 5, 6, 7, 8, 9, 30, 31, 32, 37);$$

$$A_3=(1,2, 7, 8, 9, 30, 31, 32, 37);$$

$$A_4=(16, 17,20,21,22,35);$$

$$A_{11}=(1,2,5,6,7,8,9, 16, 17,18, 19,20,21,22,30,31,32,36);$$

$$A_{12}=(1,2,5,6,7,8,9, 16, 17, 18, 19,20,21,22,30,31,32,37);$$

$$A_{13}=(1,2,7,8,9, 16, 17,18, 19,20,21,22,30,31,32,36);$$

$$A_{14}=(1,2,7,8,9, 16, 17, 18, 19,20,21,22,30,31,32,37);$$

$$A_{15}=(1,2,5,6,7,8,9, 16, 17, 18, 19,20,21,22,28,30,31,32,33,35);$$

$$A_{16}=(16, 17, 20, 21, 22, 28, 30, 31, 33, 35).$$

Списки  $A_5$ —  $A_{10}$  сохраняются без изменения.

Обозначим списки ресурсов, последовательно резервирующих ресурс  $k$ , как

$$B_k=[a_j]_k. \quad (25)$$

Тогда

$$B_1=(3,4); B_2=(3,4); B_3= B_4= \dots = B_8=(0);$$

$$B_9=((8,30,31,32), (8,21,22,34)); B_{10}= B_{11}= \dots = B_{21}=(0);$$

$$B_{22}=(21,30,31,33), (21,8,9,34)); B_{23}=(24);$$

$$B_{24}= B_{25}= B_{26}= B_{27}= (0); B_{28}=(29) , B_{29}=(28);$$

$$B_{30}=(0); B_{31}=(21,22,30,33), (8,9,30,32));$$

$$B_{32}=(33,34); B_{33}=(32,34); B_{34}=(32, 33); B_{35}=\{0\};$$

$$B_{36}=(37); B_{37}=(36).$$

Изложенный выше анализ эффективности АСУ в функции надежности дал нам текущее значение эффективности АСУ. Однако нас интересует значение эффективности  $E$ , среднее за длительный период времени. В случае стационарности и эргодичности векторного случайного процесса  $h(t)$  это значение можно найти вероятностным усреднением с учетом многомерной плотности распределения  $\omega_n(h)$  как

$$\bar{E} = \int_{-\infty}^{\infty} E(h)W_n(h)dh \quad (26)$$

Найдем среднее значение  $E$  в предположении, что отказы всех ресурсов АСУ статистически независимы, т. е.

$$P = (h_k = 1 : k = 1, \dots, s) = \prod_{k=1}^s p(h_k = 1) \quad (27)$$

В этом случае  $E$  удобнее найти не по (26), а непосредственно по относительной готовности ресурсов, т. е. по средней доле времени, в течение которого ресурс доступен:

$$P_k = P(h_k = 1). \quad (28)$$

Пусть для выполнения некоторой функции  $l$  в  $i$ -м режиме требуется набор ресурсов  $A_i$  характеризуемый единицами вектора  $\varphi_i$  - строки матрицы  $\Phi$ . Тогда среднее относительное время готовности, или коэффициент готовности,  $i$ -го способа составляет

$$\hat{P}_i = \prod_{ak \in A_i} P_k \quad (29)$$

Среднее относительное время  $P_i$ , работы в  $i$ -м режиме равно произведению  $P_i$  на среднее относительное время неработоспособности режимов, резервируемых  $i$ -м.

Пусть  $i$ -й режим относится к функции  $l$ , которой в матрице приоритетов соответствует блок  $\Psi_l$  размерности  $n_i$ ; со строками и столбцами от  $i_l = j_i$ ; до  $i_l + n_i - 1 = j_l + n_l - 1$ . Среднее относительное время неработоспособности режима  $i_l$  исполнения функции  $l$  со старшим приоритетом равно  $1 - P_{i_l}$ . Среднее относительное время действия следующего по приоритету режима, а именно  $i_l + 1$ , составит

$$\tilde{P}_{i_l+1} = (1 - \hat{P}_{i_l}) \hat{P}_{i_l+1} \quad (30)$$

По индукции найдем для произвольного режима  $i$  выполнения функции  $l(i_l < i < i_l + n_l - 1)$

$$\tilde{P}_i = (1 - \sum_{j=i_l}^{i-1} \tilde{P}_j) \hat{P}_i \quad (31)$$

Полученное для  $P_i$ , выражение обобщает аналогичные формулы, приводимые для систем с восстановлением и относящиеся ко взаимному резервированию однотипных элементов, на случай резервирования элементов, различающихся по их участию в выполнении отдельных функций.

Вычислив значения  $\tilde{P}_i$  найдем значение частной эффективности в виде

$$\bar{E}_i = E_i \tilde{P}_i. \quad (31)$$

Выше отмечалось, что в списках  $A_k$  не учитывается эквивалентное резервирование. Значения готовности  $P_k$ , входящие в (29), с учетом эквивалентного резервирования нужно находить по готовности  $P_j$  ресурсов, образующих списки  $B_k$  по (25). При статистической независимости выходов из состояния готовности ресурсов  $a_k$  и  $(a_j)_k$  пользуемся формулой

$$P_k = 1 - (1 - P_k^*) \prod_{a_j \in B_k} (1 - P_j^*), \quad (33)$$

где  $P_k^*$ ,  $P_j^*$  - готовность ресурсов, взятых самими по себе, без учета эквивалентного резервирования.

Рассмотрим вычисление средних относительных времен работы  $\tilde{P}_i$ , для примера АСУ рис.2, например 7-й функции ( $l=7$ ,  $i_l=12$ ,  $12 < i < 15$ ) без учета возможной занятости устройств. Положим, что коэффициент готовности  $P_k^*$  составляет 0,99 для всех электронных устройств ( $k=7,8,9,10,13,20,21,22,23,30,31$ ) и линий связи ( $k=2,4,11,6,14,17,19,25,32,33,34$ ); 0,98 для датчиков ( $k=1, 3, 5, 16, 18$ ) и устройств ручного ввода ( $k=24, 27, 28, 29$ ); 0,97 для исполнительных механизмов ( $k=12,15,26$ ); 0,90 для дисплеев ( $k=35,36$ ); 0,80 для печатающих устройств ( $k=37$ ). Тогда получим следующие коэффициенты готовности с учетом резервирования  $P_k$  по (32):

Теперь можно найти коэффициенты готовности режимов. Так, для режимов 12—15 получим согласно (29)

$$P_{12} = 0,842; P_{13} = 0,773; P_{14} = 0,868; \\ P_{15} = 0,797.$$

Отсюда найдем средние относительные времена работы в режимах:

$$P_{12} = 0,842; P_{13} = 0,122; P_{14} = 0,031; \\ P_{15} = 0,004.$$

Очевидно, среднее относительное время, в течение которого 7-я функция вообще не будет выполняться из-за отказов устройств, составит

$$1 - \sum_{i=12}^{15} \tilde{P}_i = 0,001$$

Формула (32) справедлива для частной эффективности такой прикладной функции АСУ, у которой эта эффективность пропорциональна доле времени, в

течение которого функция выполняется, а потери, следовательно, пропорциональны даже времени, в течение которого функция не выполняется. В действительности, однако, нередко важный вклад в потери от отказов вносит не только время, в течение которого прикладная функция не выполняется, но и сам факт отказа функции, пусть даже кратковременного. Модель зависимости эффективности АСУ от надежности выполнения функций с учетом потерь от каждого отказа была предложена в литературе. Согласно этой методике средние потери от отказа  $l$ -й функции составляют

$$\Delta E_l = \gamma_l \omega_l T_l + \varepsilon_l \omega_l, \quad (34)$$

где  $T_l$ —среднее время восстановления после отказа  $i$ -й функции;  $\omega_l$  - интенсивность отказов  $l$ -й. функции;  $\gamma_l, \varepsilon_l$  - постоянные, зависящие от характеристик технологического объекта управления и подробнее рассматриваемые в дальнейшем.

В целях обобщения формулы (34). учтем возможность реализации прикладных функций в разных режимах. Тогда средние потери от отказа  $i$ -го режима выполнения функции определяются согласно (31) как

$$\Delta E_i = (\gamma_i T_i + \varepsilon_i) \omega_i \left( 1 - \sum_{j=i_1}^{i-1} \tilde{P}_j \right).$$

Характеристики надежности исполнения функций  $T_i$  и  $\omega_i$  могут быть найдены по соответствующим характеристикам надежности ресурсов  $T_k^*$  и  $\omega_k^*$ .

Среднее время восстановления в предположении, что моменты возникновения отказов статистически независимы, а средняя относительная продолжительность отказа мала

$$\omega_k^* T_k^* \ll 1, \quad (35)$$

составляет

$$T_i = \sum_{k=1}^s T_k^* \omega_k^* / \sum_{k=1}^s \omega_k^*. \quad (36)$$

### Интенсивность отказов в $i$ -м режиме

$$\omega_i = \sum_{k=1}^s \omega_k^*. \quad (37)$$

Заметим, что коэффициент готовности  $P_k^*$ , среднее время восстановления  $T_k^*$  и интенсивность отказов  $\omega_k^*$  связаны между собой соотношением

$$P_k^* = 1 - \omega_k^* T_k^*. \quad (38)$$

Коэффициент  $\gamma_i$  задает снижение эффективности технологического процесса в единицу времени в продолжение отказа  $i$ -го режима реализации  $i$ -й функции с переходом к  $(i+1)$ -му режиму ее реализации, если  $i$ -й режим резервируется, или просто к работе без выполнения  $i$ -й функции в случае отсутствия резервирования для  $i$ -го режима. Коэффициент  $\varepsilon_i$  задает разовые потери в эффективности процесса от одного отказа  $i$ -го режима реализации  $i$ -й функции с одним из охарактеризованных выше переходов.

Величина  $\gamma_i$  складывается из потерь, пропорциональных длительности отказа и связанных с тремя факторами:

дополнительными затратами сырья, топлива, заработной платы на выпущенную за время отказа продукцию;

увеличением брака;

потерями от простоя технологического объекта.

Величина  $\varepsilon_i$  складывается из потерь, пропорциональных числу отказов и обусловленных двумя факторами - браком продукции и затратами на ремонт.

Средняя эффективность АСУ в единицу времени с учетом разовых потерь от отказов составит

$$\bar{E} = \bar{E}_* - \sum^q \Delta \bar{E}_i, \quad (40)$$

где  $E_*$  - средняя эффективность действующей (т. е. выполняющей все предусмотренные функции в основных режимах) АСУ в единицу времени.

Как уже отмечалось, значение готовности  $P_k$  ресурса определяется двумя факторами: аппаратурной надежностью ресурса и его загрузкой, т. е. вероятностью того, что к моменту времени, когда  $k$ -и ресурс потребуется для исполнения прикладной функции в  $i$ -м режиме, этот ресурс будет занят исполнением другой функции. В то время как аппаратурная надежность не зависит от проектных решений по данной АСУ и ее можно знать заранее, на загрузку ресурса эти решения влияют. Составляющая потерь, обуславливаемая занятостью ресурса  $1 - P_k$ , зависит от дисциплины обслуживания ресурсом претендентов на него и может быть либо рассчитана методами теории очередей, либо оценена имитационным моделированием.

Установим взаимосвязи между матрицей  $\Phi$  и характеристиками ресурсов, вводимых в распределенную АСУ при ее проектировании. Начнем анализ с выяснения необходимого соответствия между функциональным графом  $G_F$ , описывающим прикладные функции АСУ, и ее структурным графом  $G_S$  (см. 2 лекцию).

Прежде всего следует указать на неизбежность взаимно однозначного соответствия вершин  $G_F$ , соседних с теми терминальными вершинами, которые связаны с процессом: для них свободы выбора не остается, они должны отображаться в вершины  $G_S$ , соседние с соответствующими терминальными,

поскольку место реализации терминальных операций на технологическом объекте управления зафиксировано заранее. Таким образом, свобода в построении структурного графа  $G_S$  распространяется только на его внутренние вершины, не соседствующие с терминальными, а если соседствующие, то только с терминальными, выходящими на человека-оператора: вопрос о месте размещения, операторских пунктов входит в предмет проектирования распределенной АСУ. Будем называть вершины структурного графа, размещение которых может быть выбрано, свободными.

Для свободных вершин взаимное соответствие не является однозначным ни при переходе от  $G_F$  к  $G_S$ , ни при обратном переходе. Одна свободная вершина  $G_F$  может проецироваться в несколько вершин, если соответствующая ей операция обработки дублируется в целях повышения надежности; вместе с тем  $n$  несколько свободных вершин  $G_F$  могут проецироваться в одну вершину  $G_S$ , если несколько операций реализуется в одном аппаратурном узле.

Обработка данных в целях решения прикладных задач АСУ может принципиально выполняться в любом узле  $G_S$ , оснащенный подходящей аппаратурой, если только имеется возможность привести в него необходимые исходные данные и вывести результаты соответствующим потребителям.

Введем понятие функционального поля операции. Это понятие связано с необходимостью соответствия класса операции классу аппаратуры, размещенной в узле; если вершина  $u$  графа  $G_F$  отображается в вершину  $v$  графа  $G_S$ , то в узле должна иметься аппаратура, способная реализовать операцию  $u$ . Множество вершин структурного графа, содержащих аппаратуру, функционально пригодную для реализации операции  $u$ , назовем функциональным полем операции  $u$  и обозначим как  $V_F(u)$ .

В узле устанавливается некоторый набор устройств  $A_v = \{a_j\}_v, j = 1, \dots, J_v$ . Каждое устройство должно быть выбрано из имеющегося ассортимента

$$a_i \in A \quad (41)$$

и может реализовать некоторый набор  $\Theta_j = \{\theta_i\}$  классов операций.

Перенумеруем классы устройств и классы операций. Классы устройств:

- 1 — датчик;
- 2 — устройство аналогового ручного ввода;
- 3 — аналого-цифровой преобразователь;
- 4 — исполнительный механизм;
- 5 — цифро-аналоговый преобразователь (с памятью);
- 6 — процессор (с оперативной памятью);
- 7 — устройство передачи данных (устройство интерфейса);
- 8 — печатающее устройство (с клавиатурой);
- 9 — дисплей (с клавиатурой);
- 10 — линия связи.

Классы операций:

- 1 — первичное преобразование переменной в аналоговый сигнал;
- 2 — ручной ввод цифровых сигналов;
- 3 — ручной ввод аналоговых сигналов;
- 4 — аналого-цифровое преобразование;
- 5 — цифровая обработка;
- 6 — цифро-аналоговое преобразование (с запоминанием);
- 7 — визуальный вывод на дисплей;
- 8 — вывод на печать;
- 9 — преобразование сигнала регулирования в сигнал управления исполнительным органом.

Тогда имеем  $\Theta_8 = (8)$ .

Требование о функциональном соответствии узла  $v$  классу операций  $\theta(u)$  операции  $u$  формально описывается соотношением

$$\theta(u) \in \bigcup_{aj \in Av} j, \quad \forall u \in U(v), \quad v \in V_0 \quad (42)$$

где  $U(v)$  - отображение, обратное  $V(u)$ .

Функциональное поле  $V_F(u)$  операции и состоит из всех вершин  $v$ , удовлетворяющих соотношению (42).

Наряду с требованием функционального соответствия вершин  $v$  структурного графа вершине  $u$  функционального графа должно быть сформулировано и требование, обусловленное структурой: структура должна обеспечивать возможность приводить в узел исходные данные для реализации в нем операции и выводить к соответствующим потребителям полученные результаты. В терминах графов это означает следующее. Пусть вершины  $u, u'$  и  $u''$  в  $G_F$  отображаются соответственно в вершины  $v, v'$  и  $v''$  в  $G_S$ . Тогда должно выполняться следующее условие: если вершина  $u$  достижима в  $G_F$  из терминальной вершины  $u'$ , а терминальная вершина  $u''$  достижима в  $G_F$  из  $u$ , то в  $G_S$  вершина  $v$  должна быть достижима из  $v'$ , а  $v''$  - из  $v$ .

Этому условию для данной вершины  $u$  в  $G_S$  может удовлетворять некоторое множество вершин  $V_S(u)$  в  $G_S$ . Назовем это множество вершин структурным полем операции  $u$ .

Чтобы операция  $u$  и могла быть реализована в узле, соответствующем вершине  $v$  структурного графа  $G_S$ , последняя должна принадлежать одновременно и функциональному, и структурному полю  $u$ -й операции:

$$v \in V_F(u) \cap V_S(u). \quad (43)$$

Пересечений функционального и структурного полей, которому должна принадлежать вершина  $v$ ,

$$V_R(u) = V_F(u) \cap V_S(u) \quad (44)$$

назовем полем реализации  $u$ -й операции.

Список вершин  $v$ , удовлетворяющих условию (43), с одной стороны, ограничивает возможности перераспределения операций между узлами распределенной АСУ, а с другой, - возможности эквивалентного резервирования устройств. Для терминальных операций (ввода и вывода данных) поле реализации  $V_n(u)$  вырождается и сводится к единственной вершине структурного графа, соседней с терминальной. Действительно, каждая терминальная вершина в структурном графе имеет единственную соседнюю вершину, а значит, через нее проходит любой путь, проходящий также через терминальную вершину. Очевидно, что исходя из требований функционального соответствия аппаратуры имеет смысл размещать датчики и устройства ручного ввода только в вершинах  $v \in V_o$ , соседних с входными. Таким образом, пересечение функционального и структурного полей для терминальных операций даст поле реализации, состоящее из одной только этой вершины.

### **3. Понятия структурного поля и поля реализации для операций цифровой обработки**

Особое значение имеют понятия структурного поля и поля реализации для операций цифровой обработки. В этом случае поле реализации всегда лежит внутри базовой сети. Если подграф структурного графа, описывающий базовую сеть, представляет собой бикомпоненту, так что все его вершины взаимно доступны, то возможность распределения операций цифровой обработки различных функций между процессорными узлами внутри базовой сети и эквивалентного резервирования ограничивается только характеристиками устройств, установленных в узлах. Заметим, что это условие в распределенных АСУ по большей части соблюдается; оно выполняется в кольцевых и магистральных структурах, а также в радиальной структуре. В общем случае, однако, распределяя операции между вершинами структурного графа  $G_S$ , следует учитывать принадлежность вершин  $v$  структурному полю операции.

Рассмотрим в качестве примера функциональный граф рис.49,а и структурный: граф рис.49,б, где к базовой сети относятся вершины 3 - 7. Пусть 1-я операция требует исходных данных  $A$  и  $B$  и выдаст результаты  $C$  и  $D$ , а 2-я требует только  $A$  и выдает результат  $E$ . Здесь структурное поле и поле реализации 1 - й операции состоит из вершин 4, 5 и 6, а структурное поле и поле реализации 2-й операции - из вершин 3 - 6.

Кроме условия отображения вершины операции и на ее поле реализации требования к допустимому отображению операций цифровой обработки должны предусматривать обеспечение вычислительной мощности, памяти и пропускной способности аппаратуры передачи данных в аппаратурных узлах, а также пропускной способности линий связи. Все эти характеристики, определяемые выбранным для узла  $v$  аппаратурным комплексом  $A_v$ , будем называть каталожными данными; они должны быть известны для того, чтобы можно было выбирать аппаратуру распределенной АСУ.

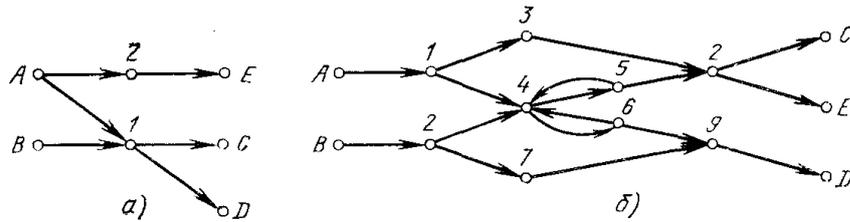


Рис.5.3. К понятию структурного поля и поля реализации  
 а – функциональный граф; б – структурный граф

К каталожным данным относят:  $A=\{a_j\}$ ,  $j = 1, \dots, J$ , - номенклатуру устройств, которые можно применять для построения АСУ;  $r(a_j)$ ,  $\theta_{o.d}$ ,  $\theta_j$  - производительность устройств  $a_j$ , измеряемую в некоторых стандартных действиях, выполняемых в единицу времени; параметр относится только к устройствам, выполняющим цифровую обработку ( $\theta_k = \theta_{o.d}$ );  $n(a_j)$ ,  $a_j \in A_{n.d}$ , - пропускную способность аппаратуры каналов передачи данных, оцениваемую в машинных словах или байтах, которые передаются и принимаются в единицу времени;  $A_{n.d}$  - класс аппаратуры передачи данных (интерфейса);  $n(c)$ ,  $c = 1, \dots, C$ , - пропускную способность линии связи  $c$ -й категории;  $b(a_j)$ ,  $a_j \in A$ , - цену аппаратуры;  $b_c$ ,  $c = 1, \dots, C$ , - цену единицы длины линии категории  $c$ .

Зная, какой набор аппаратуры  $A_v$  установлен в узле  $v$ , можно найти соответствующие параметры узлов  $r(v)$  и  $m(v)$ , а затем по  $n(a_j)$  и категории линии связи  $c(\omega)$ , проложенной в дуге  $\omega$ , рассчитать пропускную способность линии связи  $n(\omega)$ .

Далее тщательный анализ требует выяснения зависимости величины  $r, m$  и  $n$  от набора  $A_v = \{a_j\}_v$ , и эту зависимость следует учитывать во всех случаях, когда имеется возможность. Приближенно, однако, можно считать, что

$$r(v) = \sum_{a_j \in A_v} r(a_j) \quad (45)$$

$$m(v) = \sum_{a_j \in A_v} m(a_j). \quad (46)$$

Пропускная способность  $n(\omega)$  канала связи определяется и пропускной способностью линии связи, и характеристиками канальной аппаратуры. Поскольку линию связи и аппаратуру канала так или иначе согласуют по пропускной способности, введем в рассмотрение сразу пропускную способность канала  $n(\omega)$ .

Ниже, определяя загрузку узлов и линий связи, будем ориентироваться исключительно на средние значения, игнорируя очереди на обслуживание устройствами операций и функций. Столь упрощенный подход связан лишь с тем, что задача проектирования распределенной АСУ и без того достаточно сложна. Построенная АСУ может быть затем исследована в целях определения

характеристик запаздывания в выполнении функций, после чего возможно внесение поправок в принятые проектные решения. В принимаемой же здесь модели запаздывание в выполнении функций и его влияние на эффективность АСУ игнорируются. Исследование АСУ как системы массового обслуживания может производиться аналитически, но, пожалуй, более подходящим инструментом является имитационное моделирование.

Заметим также, что коррекция проектных решений, принятых на основе приводимых ниже зависимостей, может быть связана и с изменением соотношений между затратами памяти и элементарных операций процессора; это соотношение было принято нами как фиксированное на уровне функционального графа, однако оно может изменяться в довольно широких пределах в процессе программирования операций АСУ.

Загрузка узла  $v$  по обработке данных  $p(u)$ , измеряемая в тех же единицах, что и  $r(a_j)$ , находится в зависимости от отображения  $U(v)$ , обратного  $V(u)$ , как

$$\rho(v) = \sum_{u \in U(v)} \rho(u), \quad v \in V_0. \quad (47)$$

Аналогично загрузке оперативной памяти в узле  $v$  составляет

$$\mu(v) = \sum_{u \in U(v)} \mu(u), \quad v \in V_0 \quad (48)$$

При оценке загрузки каналов передачи данных будем исходить из структурного графа  $G_s$ , в котором выделены симплексные (односторонние) каналы, представляемые в структурном графе ориентированными дугами. Пусть операция  $u_1$  выполняется в узле  $v_1$  графа  $G_s$ , а операция  $u_2$  в узле  $v_2$ . Если загрузка дуги  $[U_1, U_2)$  передачей данных в функциональном графе  $G_F$  составляет  $\nu(u_1, u_2)$ , то загрузка дуги  $w = (v_1, v_2)$  в структурном графе составляет

$$(\omega) \quad \sum_{\substack{u_1 \in U \\ u_2 \in U(v_2)}} \nu(u_1, u_2). \quad (49)$$

Для отдельных весьма популярных структур, таких как кольцо и магистраль, (49) дает заниженную оценку загрузки, так как не учитывает разделения каналов между парами узлов  $(v_1, v_2)$ . В кольце для оценки загрузки каналов нужно учитывать пути прохождения пакетов между узлами. Пусть на узлах кольца задано отношение порядка  $v_\alpha < v_\beta < v_\gamma$ , если путь в структурном графе кольца из  $v_\alpha$  в  $v_\gamma$  проходит через вершину  $v_\beta$ . Тогда по каналу  $w$ , соединяющему соседние узлы  $v_1$  и  $v_2$  кольца ( $v_1 < v_2$ ) проходят все пакеты, передаваемые из узла  $v_\alpha$  в узел  $v_\beta$ , если последние удовлетворяют условиям

$$\begin{aligned} v_\alpha < v_1 < v_\beta; \\ v_\alpha < v_2 < v_\beta \end{aligned}$$

Следовательно, загрузка дуги  $w = (v_1, v_2)$  составляет

$$n(w) = \nu(v_1, v_2) = \sum_{\substack{v_\alpha \leq v_1 < v_\beta \\ v_\alpha < v_2 \leq v_\beta}} \sum_{\substack{u_1 \in U \\ u_2 \in U(v_\beta)}} \nu(u_1, u_2). \quad (50)$$

Магистраль разделяется между всеми парами узлов, так что ее загрузка

$$\nu_0 = \sum_{\substack{u_2 \in U, \\ (u_1) \neq V(u_2)}} \nu(u_1, u_2) \quad (51)$$

Сумма (51) не включает слагаемые  $\nu(u_1, u_2)$ , представляющие обмен между операциями, попадающими в один узел структурного графа, т. е. такими, для которых  $V(u_1) = V(u_2)$ .

В ряде случаев на передачу данных затрачивается время процессора. Положим для простоты рассуждений, что в узле  $v$  имеются процессоры только одного вида, например  $a_s$  и устройства передачи данных тоже только одного вида, например  $a_t$ . Обозначим число элементарных операций, затрачиваемых процессором на передачу единицы данных через устройство передачи  $a_t$ , через  $\varphi_{st}$ ; в том случае, когда передача данных идет без участия процессора,  $\varphi_{st} = 0$ .

С учетом затрат времени процессоров на передачу данных затраты производительности аппаратуры обработки данных в узле  $v$  составят

$$\rho'(v) = \rho(v) + \varphi_{st} \nu(v), \quad (a_s \in A_v) \wedge (\theta_{o,d} \in \Theta_s), \quad a_t \in (A_v \cap A_{п.д}). \quad (52)$$

Таким образом, условия, налагаемые на отображение  $V(u)$  согласованием параметров функционального и структурного графов, выражаются как

$$r(v) \geq \rho'(v), \quad v \in V_{00}; \quad (53)$$

$$m(v) \geq \mu(v), \quad v \in V_{00}; \quad (54)$$

$$n(w) \geq \nu(w), \quad w \in W_{00} \quad (55)$$

для сетей с парной связью узлов и

$$n(w) \geq \nu_0 \quad (56)$$

для магистралей.

Следует заметить, что поскольку чаще всего в сетях АСУ применяют каналы передачи данных с одинаковой пропускной способностью (лишь в

отдельных системах устанавливают две различные магистрали - ближнего и дальнего действия), соотношение (56) переписывается как

$$n_0 \geq v(\omega), \quad \omega \in W_{00}, \quad (57)$$

где  $n_0$  - пропускная способность применяемых каналов передачи данных.

Затраты на аппаратуру без труда находятся по каталожным данным, если выбраны наборы аппаратуры  $A_v = \{a_j\}_v$ . Стоимость аппаратуры в узле  $v$  составляет

$$\sum_{a_j \in A_v} \quad (58)$$

а общая стоимость аппаратуры в узлах

$$Q_A = \sum_{v \in V_0} \sum_{a_j \in A_v} \quad (59)$$

Затраты на линии передачи данных пропорциональны длине линий и цене приобретения и прокладки единицы длины. Длина линий определяется размещением узлов. Пусть координаты узла  $v$  описываются вектором  $y(v)$ , в общем случае трехмерным. Обозначим узлы, инцидентные линии  $\omega$ , через  $v_1$  и  $v_2$ , а длину линии  $\omega$  - через

$$l(\omega) = l(y(v_1(\omega)), y(v_2(\omega))). \quad (60)$$

Подробности определения длины линий связи в АСУ с учетом их категории изложены в [7].

Цену приобретения и прокладки единицы длины линии категории  $c$  обозначим как  $b_c(\omega)$ , показывая тем самым, что категория  $c$  зависит от линии  $\omega$ . Тогда общие затраты на провода для линии связи со

$$Q_W = \sum_{\omega \in W} b_c(\omega) l(v_1, v_2) \quad (61)$$

Задача прокладки магистрали минимальной длины, соединяющей набор узлов, при заданной матрице расстояний между узлами

$$L = \| \| l_{\alpha\beta} \| \| = \| \| l(v_\alpha, v_\beta) \| \| \quad (62)$$

оказывается близкой к известной задаче коммивояжера (см., например, [78]), а в случае петлевой магистрали в точности совпадает с ней. Общие затраты на технические средства АСУ составляют

$$Q = Q_A + Q_W, \quad (63)$$

т.е. равны сумме затрат на аппаратуру линии связи.

### Контрольные вопросы

1. Что такое проектирование АСУ? [Л.7, 174-176].
2. Постройте функциональный граф АСУ с учетом неэквивалентного резервирования. [Л.7, 176-178].
3. Как строится матрица приоритетов  $\psi$ ? [Л.7, 179-180].
4. Напишите формулу, определяющую текущее значение эффективности  $E(t)$  функционирования АСУ. [Л.7, 181-183].
5. Нарисуйте модель эффективности АСУ. [Л.7, 184-185].
6. Изложите анализ эффективности АСУ в функции надежности. [Л.7, 186-189].
7. По каким соотношениям определяется средняя эффективность АСУ в единицу времени с учетом разовых потерь от отказов? [Л.7, 190-192].
8. Какими факторами определяется значение готовности  $P_k$  ресурса? [Л.7, 193-194].
9. Напишите взаимосвязи между матрицей  $\Phi$  и характеристиками ресурсов. [Л.7, 194-195].

## Лекция №10

### Оптимальное проектирование распределенных АСУ

#### План

1. Задачи построения распределенной АСУ.
2. Методы оптимального проектирования (алгоритмы синтеза) вычислительной сети.
3. Оптимальное проектирование топологии сетей ЭВМ.

#### 1. Задачи построения распределенной АСУ

При построении распределенной АСУ можно исходить из разных уровней определенности ее структуры. В зависимости от выбранного уровня определенности приходится решать одну из трех задач формулируемых последовательно.

1) Структура АСУ не установлена - не известны ни размещение узлов базовой сети, ни параметры вершин и дуг соответствующего базовой сети подграфа структурного графа. Требуется оптимальным образом разместить узлы базовой сети, определить параметры структурного графа базовой сети и либо распределить операции между его узлами, либо установить протокол динамического распределения операций.

2) Структурный граф АСУ построен, т. е. известно размещение узлов, но параметры его базовой сети не заданы. Требуется определить оптимальные параметры базовой сети (выбрать аппаратуру обработки (передачи данных) и либо распределить операции между узлами, либо задать протокол динамического распределения операций.

3) Структурный граф АСУ построен и его параметры заданы; иными словами, аппаратура и её размещение выбраны. Требуется оптимально распределить операции обработки между узлами базовой сети либо установить протокол их динамического распределения.

Задачу 1 назовем общей задачей оптимального проектирования распределенной АСУ. Это - нелинейная комбинаторная задача. Уже при пяти-шести узлах базовой сети высокая размерность, обусловленная числом допустимых комбинаций переменных и необходимостью учета

многочисленных ограничений, не позволяет строго решить общую задачу. Поэтому обычно для решения подобных задач применяются декомпозиционные процедуры, разлагающие общую задачу на ряд связанных между собой частных подзадач.

Опыт практической разработки распределенных АСУ еще незначителен, однако стоящие здесь задачи во многом сходны с задачами, которые решаются при разработке сетей ЭВМ иного назначения, а в этой области за последние 10 - 15 лет уже накоплен значительный опыт как для сетей, распределенных по обширной территории, так и для локальных сетей. При проектировании сетей ЭВМ чаще всего выделяют следующие подзадачи:

А. Выбор оптимальной топологии вычислительной сети.

В. Выбор аппаратуры обработки данных.

С. Оптимальное распределение данных и программ по вычислительной сети.

Различная последовательность решения этих подзадач определяет схему той или иной эвристической декомпозиции общей задачи оптимального проектирования.

Несмотря на сложность общей задачи проектирования вычислительной сети в ряде работ делаются попытки решения задачи без декомпозиции ее на подзадачи. В некоторых работах две из трех подзадач А - С объединяются в одну. Естественно, что ввиду огромной размерности этих задач такие попытки осуществляются успешно, если благодаря вводимым при постановке ограничениям удастся выделить достаточно узкий класс структур сети, удовлетворяющих этим ограничениям.

Характеристики сети передачи и аппаратуры обработки данных, булевы переменные, характеризующие распределение данных и программ по сети, параметры самих данных и программ выступают в качестве переменных и ограничений единой задачи оптимального размещения вычислительных центров в отраслевых АСУ. Решение достигается путем значительного сокращения перебора благодаря введению достаточно жестких ограничений. Выбор топологии в этих постановках ограничен заданием так называемого графа возможных связей, для которого исходя из ограничений на показатели качества обслуживания рассчитываются матрицы стоимости каналов связи и матрицы стоимости передачи информации между всеми вершинами.

Оптимизация сети осуществляется при заданном распределении данных и программ по узлам обработки данных. Ввиду большей информационной связности процессов АСУ по сравнению с отраслевыми АСУ такое эвристическое априорное распределение в АСУ может быть оправдано только для программ обработки некоторых простых видов запросов. Например, если результат обработки в АСУ требуется практически в том месте, где собираются исходные данные, то естественно размещение программы обработки этих данных в процессоре, расположенном в непосредственной близости от места сбора данных. Это приводит к повышению надежности передачи данных. При

этом может достигаться и экономия на самой аппаратуре передачи данных. Впрочем, эвристическое распределение сложных прикладных программ, требующих сбора данных от территориально удаленных друг от друга источников информации, может привести к решению, сильно отличающемуся от оптимального.

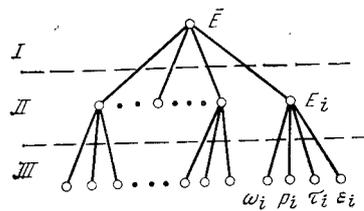
Подобная описанной выше постановка многомерной задачи целесообразна лишь тогда, когда процесс, природа объекта, а также аппаратура для построения АСУ, имеющаяся на момент проектирования, определяют достаточно жесткие ограничения при решении поставленной задачи.

Во всех охарактеризованных выше постановках в качестве критерия оптимального проектирования выбирается минимум суммы стоимости создания и функционирования вычислительной сети, а это может быть целесообразно при построении больших дорогостоящих сетей, состоящих из десятков машин, отстоящих на сотни километров друг от друга. В условиях построения относительно дешевой сети АСУ стоимость, как правило, можно считать ограничивающим фактором. В качестве критерия оптимального синтеза распределенной АСУ следует принять максимальную разность между средней экономией, отнесенной к единице времени,  $E$  и капитальными затратами на ее создание  $Q$ , отнесенными к нормативному сроку окупаемости  $T_0$ :

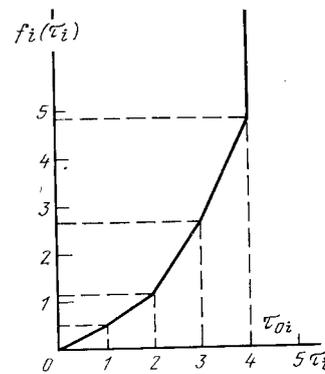
$$\text{Max} E_{0=\text{max}}(E-Q/T_0) \quad (64)$$

Основным же критерием функционирования АСУ должна быть максимальная эффективность выполнения ею предписанных ей функций. Эта эффективность характеризуется средней экономией, приносимой АСУ в единицу времени.

Обычно вследствие недостаточных сведений о зависимости средней экономии  $E$  от показателей, характеризующих функционирование АСУ, непосредственное исследование значения средней экономии  $E$  вряд ли возможно. Поэтому для перехода от средней экономии  $E$  к показателям АСУ можно предложить дерево показателей эффективности (рис.50). Переход к показателям, расположенным в дереве на более низком уровне, означает отказ от более общего показателя в пользу более частных, вносящих менее весомый вклад в общий критерий верхнего уровня, но зато легче вычисляемых при проектировании. Вопрос о том, какой (или какие) из показателей следует вводить в критерий, применяемый на данном уровне дерева показателей, а какие использовать в ограничениях, решается обычно эвристически, и это объясняет наблюдаемую пестроту в постановках проектных задач.



**Рис. 50. Дерево показателя эффективности**



**Рис. 51. Кусочно-линейная функция штрафа за запаздывание**

На уровне, следующем за средней экономией, располагаются так называемые частные эффективности  $E_i$  - средние значения эффективности выполнения функций АСУ. На третьем уровне дерева эффективностей располагаются факторы, оказывающие основное влияние на частные эффективности: прежде всего надежность выполнения функций, а также запаздывание в реализации функций, точность реализации функций. Пожалуй, наиболее практично в качестве критериев функционирования АСУ использовать максимум (минимум) некоторой функции от показателей третьего уровня дерева эффективностей, например минимум суммы взвешенных значений запаздывания в реализации функций. При этом веса выбираются или экспертами, или в соответствии с функциями штрафов  $f(\tau_i)$ , где  $\tau_i$  - запаздывание в реализации  $i$ -й функции, изменяющееся от нуля до момента времени  $\tau_{0i}$  когда окончание реализации данной функции равнозначно ее невыполнению (рис.51). В качестве другого оптимизируемого функционала можно предложить максимум минимального показателя надежности; согласно этому критерию повышается вероятность предотвращения невыполнения функции, надежность реализации которой имеет минимальное значение.

## 2. Методы оптимального проектирования (алгоритмы синтеза) вычислительной сети

Для решения общей задачи оптимального проектирования вычислительной сети разными авторами используются самые различные методы: динамическое программирование, метод ветвей и границ, эвристические подходы. Выбрать подходящий для проектирования АСУ принцип удастся, по-видимому, только в процессе решения конкретных задач.

Все методы решения общей задачи оптимального проектирования вычислительных сетей можно разделить на два класса: строгие и субоптимальные (эвристические). В силу отмеченной выше многомерности и нелинейности задачи комбинаторной оптимизации вычислительной сети строгие методы математического программирования не нашли широкого применения при проектировании вычислительных сетей.

Например, для классической схемы метода ветвей и границ объем вычислений растет экспоненциально; в лучшем случае достигается перебор порядка  $2C^{[n/2+1]}_{n+1}$ , где  $n$  - размерность задачи, а  $[\ ]$  - символ целой части числа.

Проектирование вычислительной сети обычно осуществляется в условиях неполной информации о возможном наращивании и изменении сети в будущем, об изменении состава задач, решаемых вычислительной сетью. В таких условиях вряд ли уместно применение строгих методов оптимизации и вполне оправдано использование различных эвристических процедур, позволяющих в приемлемые для проектирования сроки получить решения, несущественно отличающиеся от оптимальных.

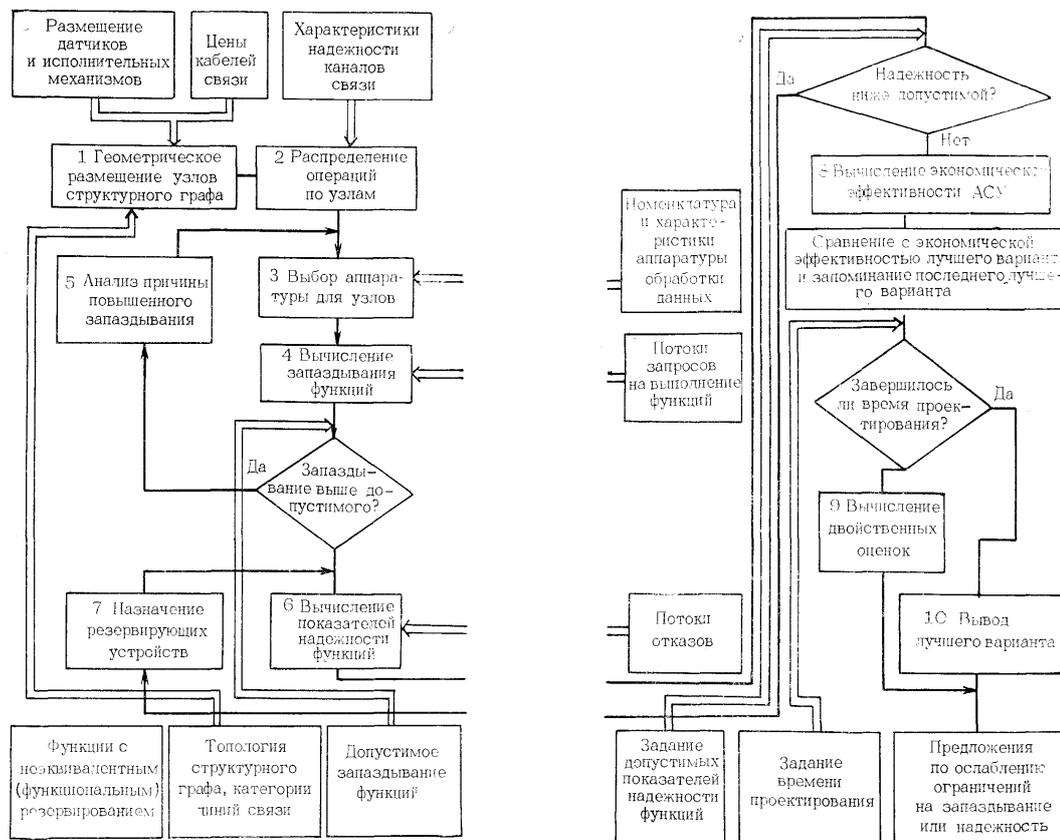


Рис.52. Схема алгоритма синтеза распределённой АСУ

Эвристические или субоптимальные алгоритмы решения общей задачи проектирования вычислительной сети можно разделить на два класса. К первому относятся алгоритмы локально-оптимального поиска. Для решения задачи такими алгоритмами экспертным путем выделяется ограниченная область пространства решений, а затем уже среди локальных экстремумов этой области производится поиск глобального экстремума. Эффективность таких алгоритмов во многом обусловлена тем, насколько удачно выбраны границы области пространства решений, что в свою очередь зависит от предполагаемой структуры проектируемой вычислительной сети и от характеристик имеющейся аппаратуры передачи и обработки данных; от размеров же выделенной области пространства решений зависит время решения задачи по алгоритму локально-оптимального поиска.

Субоптимальные алгоритмы второго класса используют эвристические декомпозиции общей задачи проектирования вычислительной сети.

Ниже описана схема синтеза распределенной АСУ, представленная на рис.52 и основанная на следующей эвристической последовательности решения общей задачи оптимального проектирования:

1) определение числа узлов и категорий линий связи между ними исходя из соображений, не отражаемых математической моделью, в частности с учетом повышения интенсивности поступления сигналов в аварийных условиях;

2) оптимальное размещение узлов без учета реализуемых в них операций, по минимальной стоимости линий связи;

3) оптимальное распределение операций между узлами обработки по критерию минимальной суммы взвешенных (по коэффициентам готовности) расстояний узла обработки от терминальных узлов, которые участвуют в операции, выполняемой в данном узле обработки;

4) подбор аппаратуры узлов обработки, отвечающей требованиям выполняемых в узлах операций, по минимальной стоимости аппаратуры.

Исходными данными для проектирования распределенной АСУ служат:

перечень функций и способов их выполнения (т. е. перечень способов функционального резервирования) с указанием их приоритетности;

эти данные желательно представить в виде функционального графа;

возможная пиковая интенсивность потока сигналов;

размещение датчиков и исполнительных механизмов на объекте управления;

частные эффективности выполнения функций АСУ, хотя бы в виде приближенных экспертных оценок, включая зависимость эффективности от запаздывания;

доступная номенклатура аппаратуры и ее основные технические характеристики - разрядность, производительность процессоров, оперативная память, пропускная способность каналов связи и терминалов;

цены на аппаратуру и линии связи;

характеристики надежности аппаратуры в виде коэффициентов готовности и интенсивности отказов;

число узлов базовой сети и категории каналов связи между ними;

эти данные желательно представить в виде структурного графа, в котором будут заданы параметры дуг, но не узлов.

Кроме того, проектант задается такими значениями запаздывания и надежности выполнения функций, которые представляются ему предельно допустимыми, а также определенным временем, отведенным на вычисления для проектирования.

Блок 1 укрупненной схемы алгоритма определяет координаты узлов структурного графа по минимальной суммарной стоимости линий связи.

Блок 2 определяет узел, в котором будет выполняться каждая операция по минимальной сумме взвешенных по показателям надежности расстояний от узла обработки до терминальных узлов, имеющих информационную связь с данной операцией.

Блок 3 подбирает аппаратуру для установки в каждом узле, способную выполнить операции, определенные в блоке 2.

Блок 4 выполняет анализ полученной аппаратурной структуры в целях определения запаздывания в выполнении каждой функции. Если по какой-либо функции запаздывание окажется больше, чем предполагавшееся допустимым, блок 5 производит анализ причины повышенного запаздывания и выдает в блок 3 дополнительные ограничения, с учетом которых заново подбирается аппаратура, с тем чтобы запаздывание по данной функции сократить до допустимого значения.

Блок 6 вычисляет показатели надежности для каждой функции (с учетом потока отказов), и если по какой-либо функции показатель окажется хуже допустимого, блок 7 назначает резервирующее устройство, улучшающее соответствующий показатель надежности.

Когда будут обеспечены заданные значения запаздывания и надежности, блок 8 вычисляет общий критерий эффективности АСУ. Полученное значение критерия сравнивается с лучшим значением критерия среди всех ранее проанализированных вариантов АСУ, и если полученное значение окажется лучшим (или если оно первое), вариант запоминается как лучший.

После этого проверяется, не истекло ли время, отведенное на синтез АСУ. Если время не истекло, блок 9 вычисляет двойственные оценки и выдает проектанту предложения об увеличении допустимого запаздывания или снижении допустимой надежности одной или нескольких функций. Проектант, приняв решение, вводит новые значения ограничений или изменяет топологию структурного графа, после чего строится новый вариант АСУ.

По истечении времени проектирования блок 10 выводит лучший из полученных вариантов со всеми его технико-экономическими характеристиками проектанту.

Задача распределения данных и программ по станциям обработки данных вычислительной сети заключающиеся в следующем: для каждой станции вычислительной сети задано множество программ и данных, подлежащих обработке в данной станции; надо распределить данные и программы по станциям вычислительной сети так, чтобы суммарная стоимость хранения и передачи данных и программ по сети была минимальной. При этом необходимо соблюдение следующих ограничений:

а) ожидаемое время доступа станций вычислительной сети к соответствующим данным и программам не должно превышать установленного граничного значения;

б) вероятность того, что определенные данные и программы доступны станциям вычислительной сети в произвольный момент времени, должна быть не меньше вероятности, установленной для каждого вида данных и программ;

в) емкость памяти в каждой станции ограничена некоторым значением.

Под временем доступа станции вычислительной сети к определенным данным или программам понимается отрезок времени от момента инициации запроса со стороны данной станции до момента начала приёма этой станцией требуемых данных или программ. Значение ожидаемого времени доступа данной станции к соответствующим данным или программам, требуемое для составления ограничения п. а, рассчитывается на основе вероятностных характеристик потоков запросов тех или иных данных и программ со стороны данной станции. Если поток запросов считать пуассоновским, то тогда ожидаемое время доступа  $a_{ijk}$   $i$  - й станции к  $j$  - му набору данных или программ, расположенному в  $k$ -й станции, определяется выражением

$$a_{ijk} = \tau_{ik} p_{ik} / 2(1 - p_{ik}), \quad i \neq k, \quad (65)$$

где  $p_{ik}$  - интенсивность потока сообщений от  $k$ -й к  $i$ -й станции;  $\tau_{ik}$  - среднее время передачи сообщения от  $k$ -й к  $i$ -й станции.

Если принять, что все станции обработки данных имеют один и тот же коэффициент готовности  $a_p$  и все каналы связи сети один и тот же коэффициент готовности  $a_c$ , то вероятность  $A_j$  того, что  $j$  - й набор данных или программ доступен определенным станциям сети в произвольный момент времени, определяется формулой

$$A_j = a_p [1 - (1 - a_c a_p)^j] \quad (66)$$

где  $r_j$  - количество копий  $j$  - го набора данных или программ, распределяемых по станциям вычислительной сети.

Таким образом, вероятность  $A_j$  значение которой требуется для записи ограничений п. б, равна произведению коэффициента готовности станции, запрашивающей  $j$  - й набор данных или программ, на коэффициент готовности  $r_j$ , копий  $j$  - го набора, хранящихся в других станциях  $1 - (1 - a_c a_p)^j$

При решении задачи распределения данных и программ строгими методами она записывается как нелинейная булева задача математического программирования. Оптимизируемый функционал представляет собой сумму стоимости хранения данных и программ и стоимости загрузки сети связи при данном распределении. В свою очередь стоимость загрузки сети связи является суммой стоимости передачи данных и программ между станциями сети и стоимости передачи сообщения для обновления данных и программ.

Переменные задачи  $x_{ij}$  принимают значения 0 или 1 в зависимости от того, будет ли размещен  $j$ -й набор данных или программ в  $i$ -станции,

Количество копий каждого набора данных или программ задаётся заранее как ограничение, но в наиболее общей постановке задачи эта величина также является оптимизируемой переменной. Если же принять, что в сети будет распределено только по одной копии данных или программ и при этом будут удовлетворены ограничения п.п. а, б, то задача становится линейной.

Среди известных моделей задачи наиболее подходящей для распределенной АСУ является описанная в [7]. Здесь вводятся два вида булевых переменных. Одни из них характеризуют распределение наборов данных по станциям сети, а другие - распределение программ. Предполагается, что программа всегда размещается в той же станции, где по этой программе решается соответствующая задача сети. Следовательно, при решении по такой модели задачи распределения данных и программ фактически определяются места решения задач в вычислительной сети. Такое ограничение на размещение программ, вполне оправданное для распределенной АСУ, означает, что если несколько задач вычислительной сети решаются в разных её станциях по одной и той же программе, то предпочтительнее хранить данную программу  $v$  во всех нуждающихся в ней станциях, а не в одной из них. Такой выбор объясняется как меньшей стоимостью хранения программы по сравнению с ее передачей, так и меньшей вероятностью искажения программы при ее хранении, чем при передаче.

Методы распределения данных и программ при применении таких подробных моделей ввиду целочисленности и нелинейности задачи очень громоздки и вряд ли позволяют решить задачу в приемлемые для проектирования сроки при числе станций сети порядка больше десяти. Такие модели не оправданы и в условиях неполной информации о возможных изменениях состава задач сети и характеристик наборов данных для этих задач. Поэтому в условиях промышленного проектирования нахождение оптимального распределения данных и программ необязательно и для решения задачи естественно использование субоптимальных алгоритмов. Например, при синтезе распределенной АСУ по схеме, приведенной на рис.52, станции, в которых будут выполняться те или иные операции, определяются по минимальной сумме взвешенных по показателям надежности расстояний от узла, где размещается станция обработки, до терминальных узлов, имеющих информационную связь с данной операцией.

Задача ставится следующим образом. Задан набор задач, реализующих функций управления

$$\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_l\},$$

и матрица смежности  $A$ , состоящая из нулей и единиц и характеризующая связи задач по входам и выходам. Требуется найти такое представление исходного набора задач

$$\Phi^* = \{\varphi_1^*, \varphi_2^*, \dots, \varphi_r^*\}, \quad r \leq l,$$

при котором минимизируется следующий функционал качества:

$$I = k_1 P_{св} + k_2 P_{отк} + k_3 T_{вып}, \quad (67)$$

где  $P_{св}$  - мощность связи структуры;  $P_{отк}$  - вероятность отказа системы;  $T_{вып}$  - время выполнения задач системы;  $k_1, k_2, k_3$  - весовые коэффициенты.

Величина  $r$  называется степенью децентрализации системы;  $P_{св}$  определяется суммированием элементов матрицы  $A$ . Эвристическое решение основано на целенаправленном попарном объединении задач исходного набора и проверке на экстремум функционала качества.

Для распределения данных и программ предлагается диалоговая процедура, при которой проектант проводит распределение вручную, а затем с помощью специальной программы проектирования проверяет соблюдение ограничений при данном размещении программ и данных. Пользуясь этой программой, проектант может быстро оценить загрузку процессоров, линий связи и требуемую память в каждой станции. В результате проектант получает мощное средство для модификации первого пробного распределения; затем он снова использует программу для отыскания распределения, удовлетворяющего заданным ограничениям.

Для облегчения принятия решения об изменениях в распределении задач программа проектирования предоставляет проектанту дополнительную информацию, например, о том, какая станция оказалась перегруженной или какие задачи создают основную перегрузку процессора.

### 3. Оптимальное проектирование топологии сетей ЭВМ

Оптимальному проектированию топологии сетей ЭВМ посвящена обширная литература, основываясь на которой, можно предложить методы и модели, подходящие для проектирования распределенных АСУ.

Задача топологической оптимизации вычислительной сети состоит в поиске такой топологии сети передачи данных между узлами вычислительной сети, для которой достигается оптимум некоторого критерия при соблюдении определенных ограничений на показатели функционирования. Следует отметить, что обычно под топологией сети понимается взвешенная топология, т. е. конфигурация каналов связи вместе с их характеристиками (весами).

Задачи синтеза топологии вычислительных сетей можно разбить на два класса в соответствии с рассмотрением сети ЭВМ как состоящей из двух подсетей: базовой сети, которая образуется из ЭВМ, соединенных между собой непосредственно или через связанные процессоры, и терминальной сети, которая служит для подключения терминалов пользователей к узлам базовой сети. На практике отмечается существенная связь между базовой и терминальной сетями. Однако без ущерба для эффективности решения задачи принимается, что топология терминальной сети определяется лишь числом и расположением узлов базовой сети и не зависит от конфигурации каналов связи

базовой сети, а при оптимизации топологии базовой сети достаточно учитывать лишь внешний для нее трафик от терминальной сети, не принимая во внимание топологию последней.

В распределенной АСУ базовая сеть состоит из узлов обработки, связанных между собой сетью передачи данных, а терминальная сеть - это остальная часть аппаратурного комплекса АСУ, т. е. датчики, исполнительные механизмы, пульта дистанционного управления, связанные каналами связи с узлами базовой сети.

Основными показателями при топологическом проектировании вычислительных сетей являются: время задержки передачи сообщения; надежность сети; стоимость сети.

Надежность сети ЭВМ можно определить так способность сети обеспечить непрерывную связь между всеми действующими парами узлов. Для измерения надежности сети обычно пользуются понятием связности сети. Сеть называется  $\mu$  - связной, если для каждой пары узлов существует минимум  $\mu$  связывающих их и непересекающихся путей. Иными словами, чтобы некоторая пара узлов сети оказалась несвязной, должно отказать не менее  $\mu$  вершин и (или) ребер сети.

Для распределенной АСУ характеристика надежности в виде-связности представляется недостаточной, в связи с чем для характеристики надежности выполнения функций вводятся показатели готовности и интенсивности отказов. Задача их вычисления гораздо более трудна, чем вычисление связности сети, так как связность определяется только конфигурацией сети, а для вычисления коэффициентов готовности и интенсивности отказов нужно знать и дисциплину резервирования, определяемую структурой распределенной АСУ, протоколами обмена между узлами сети, организацией программного обеспечения АСУ.

Особенности топологического проектирования распределенных АСУ по сравнению с сетями ЭВМ состоят и в разных порядках размерности соответствующих задач. Размерность терминальной сети в распределенных АСУ может иметь тот же порядок, что и в сетях ЭВМ, размерность же базовой сети может колебаться всего от нескольких узлов до нескольких десятков (последнее лишь в исключительных случаях). Поэтому средняя размерность базовой сети распределенных АСУ меньше, чем сети ЭВМ, а это может упростить оптимальное проектирование. В то же время АСУ могут повторно проектироваться для аналогичных технологических процессов, и, как правило, чем меньше сама АСУ, тем больше ее тираж, следовательно, тем тщательнее нужно добиваться оптимума при ее проектировании. Поэтому следует считать обоснованным применение строгих математических методов оптимизации при топологическом проектировании распределенных АСУ при условии, конечно, возможности решения задачи по таким методам в приемлемые для разработки сроки.

Оптимальный синтез топологии базовых сетей проводится в различных постановках, при которых один из трех вышеупомянутых показателей служит

критерием проектирования, а остальные два входят в ограничения задачи. В качестве ограничивающих могут выступать и другие параметры сети:

матрица тяготения (траффика), характеризующая интенсивность обмена сообщениями для каждой пары узлов сети;

характеристики каналов связи;

число узлов сети;

граф возможных связей между узлами сети,

При синтезе терминальных сетей задано размещение терминалов, объем траффика от каждого терминала, характеристики доступной аппаратуры связи и концентраторов. Требуется разместить пункты концентрации (входа в базовую сеть), распределить терминалы между пунктами концентрации и установить категорию канала связи от каждого терминала. Задачу синтеза терминальной сети отличает большая размерность, так как число терминалов в вычислительных сетях может достигать нескольких тысяч, но структура терминальных сетей намного проще, чем базовых. Обычно возможно представление терминальной сети в виде графа древовидной структуры, что приводит к ряду упрощений при синтезе терминальной сети. Так, распределение потоков в терминальной сети известно заранее, поскольку граф терминальной сети не содержит циклов; следовательно, между любой парой вершин существует единственный путь. В качестве критерия оптимального синтеза терминальной сети почти всегда принимают минимальную стоимость, что объясняется явным преобладанием стоимости терминальной сети в общей стоимости всей сети.

Применение строгих методов оптимального топологического проектирования вычислительных сетей крайне затруднено в связи с большой размерностью и целочисленностью задачи. Например, при числе узлов, равном 10, число переменных, входящих в модель, равно примерно  $10^4$ . Задачу целочисленного программирования такой размерности невозможно решить в реальные сроки. Наиболее распространенные строгие методы оптимизации топологии вычислительной сети основаны на принципе ветвей и границ, хотя достижение в реальное время оптимального решения этим методом может быть получено для сетей с числом узлов не более десяти. Принцип ветвей и границ, применяется для оптимизации топологии межцентральной распределенной сети передачи данных. Каждой вершине дерева ветвлений сопоставляется подмножество ветвей, уже включенных в искомую сеть; подмножество ветвей, запрещенных для включения в искомую сеть; подмножество ветвей, для которых еще не принято решение об их включении в сеть. При ветвлении из активной вершины проводятся две дуги: одна из них предопределяет включение в сеть некоторой дуги, другая запрещает его.

Ввиду перечисленных выше причин, затрудняющих применение строгих методов, развиваются эвристические методы топологической оптимизации.

Теоретической основой для оптимизации терминальной сети является задача о построении минимального каркаса. Подход состоит в следующем.

Сначала все терминалы рассматриваются как относящиеся к отдельным компонентам. Затем производится объединение компонентов путем их объединения линиями кратчайшей длины или минимальной стоимости. Объединение двух компонентов может быть осуществлено лишь при условии, что суммарный компонент не нарушает заданных ограничений.

Для оптимального топологического проектирования базовой сети с распределенной структурой предложено много подходов, основанных на многочисленных алгоритмах известной задачи о коммивояжере.

### **Контрольные вопросы**

1. Сформулируйте задачи построения распределенной АСУ, исходя из разных уровней определенности ее структуры. [Л.7, 196-197].
2. Что выбирается в качестве критерия оптимального проектирования. [Л.7, 197-198].
3. Нарисуйте дерево показателей эффективности АСУ. [Л.7, 199-200].
4. Какие методы используются при решении общей задачи оптимального проектирования вычислительной сети? [Л.7, 201-203].
5. Начертите схему алгоритма синтеза распределенной АСУ. [Л.7, 203-204].
6. Перечислите исходные данные при проектировании распределенной АСУ. [Л.7, 205-206].
7. Опишите блоки укрепленной схемы алгоритма синтеза распределенной АСУ. [Л.7, 206-207].
8. Методы распределения данных и программ. [Л.7, 207-208].
9. Что является задачей топологической оптимизации вычислительной сети. [Л.7, 207-208].
10. Что является основными показателями при топологическом проектировании вычислительных сетей. [Л.7, 208-209].

## Ключевые слова

**Директор**—ранг станции сети, содержащей несколько магистралей; станции этого ранга присваивают статус распорядителя станциями каждой магистрали и организуют обмен сообщениями между магистралями.

**Заказчик** — ранг станции магистрали, при котором станция имеет право инициативного запроса о предоставлении ей статуса инициатора.

**Инициатор** — ранг станции магистрали, при котором станция имеет право запроса данных с других станций и инициативной передачи.

**Интерфейс** — в слоистой архитектуре правила взаимодействия между функциями соседних уровней.

**Исполнитель** — ранг станции магистрали, при котором станция имеет право приема данных и передачи по запросу.

**Кадр** — набор элементов информации, который служит для передачи информации от одной станции к другой и в котором роль каждого элемента стандартизована протоколом связи. В узком смысле термин применяется к каналному уровню архитектуры связи.

**Комплекс средств для распределенных АСУ**— набор аппаратурных и программных средств, рассчитанный на совместную работу в распределенной АСУ и допускающий сочетание средств в широком диапазоне конфигураций.

**Конфигурирование** — задание параметров прикладных функций АСУ ГП ручным вводом их в память АСУ, аппаратура которой уже установлена и действует на объекте.

**Координационная станция** — станция АСУ, предназначенная для решения прикладных задач согласования работы отдельных участков технологического объекта управления в рамках одной распределенной АСУ. Координирующая станция, как правило, не имеет прямого выхода ни на технологический объект, ни на человека-оператора.

**Локальная вычислительная сеть** — вычислительная сеть, узлы которой размещаются в пределах одного или нескольких соседних зданий одной организации.

**Локальная технологическая станция** — станция, предназначенная для связи, распределенной АСУ с определенным участком технологического процесса; непосредственно соединяется с датчиками, исполнительными устройствами, местными коммутаторами и регуляторами и решает автономные задачи управления данным участком.

**Магистраль**—канал связи, соединяющий несколько станций, которые разделяют его во времени; чаще всего термин применяется к каналам с последовательной передачей данных.

**Маркер, или жезл** — группа символов, циркулирующая от станции к станции в магистральной или кольцевой сети и дающая принявшей ее станции статус инициатора.

**Операторская станция** — станция, предназначенная для связи распределенной АСУ с оператором-технологом; в ее состав входят или непосредственно соединяются с ней дисплеи, клавиатуры, печатающие устройства и т. п.

**Операция** — одно из последовательного ряда действий, выполняемых аппаратурой АСУ для реализации прикладной функции.

**Пакет** — кадр (в широком смысле) сетевого уровня архитектуры связи.

**Приемник** — ранг станции магистрали, при котором станция имеет право только принимать сообщения.

**Прикладная функция** — функция АСУ, заключающаяся в выработке управляющего воздействия на объект или сигнала, адресуемого оператору или внешнему объекту (например, другой АСУ) по данным, полученным с технологического процесса, от оператора и (или) от внешнего объекта.

**Произвольный доступ (к магистрали)**—принцип разделения магистрали во времени, разрешающий любой станции начинать передачу при условии, что магистраль свободна.

**Протокол** — в архитектуре связи правила взаимодействия между объектами одного уровня архитектуры, расположенными в разных станциях вычислительной сети.

**Распределенная АСУ** — содержащая несколько процессоров, которые выполняют различные операции, реализующие прикладные функции АСУ.

**Распорядитель** — ранг станции магистрали, дающий станции право предоставлять статус супервизора другим станциям.

**Связная станция** — станция АСУ, предназначенная для управления обмена данными по сети передачи данных АСУ или для организации связи с другой АСУ.

Слоистая архитектура — принцип построения системы ф- в виде ряда слоев, или уровней, организуемых таким образом, что функции каждого уровня для их реализации обслуживаются функциями соседнего с ним нижележащего уровня.

**Слой архитектуры** — см. Уровень архитектуры.

**Сообщение** — кадр (в широком смысле) транспортного и более высоких уровней архитектуры связи, вплоть до прикладного.

**Станция** — конструктивная единица комплекса среден для распределенных АСУ, допускающая установку па объекте в месте, которое соответствует се функциональному назначению, и имеющая собственный адрес в сети передачи данных АСУ.

**Структурный граф распределенной АСУ** — ориентированный граф, в котором вершины соответствуют узлам вычислительной сети, а дуги — линиям связи между ними.

**Супервизор** — ранг станции магистрали, дающий станции право предоставлять статус инициатора другим станциям.

**Терминальный узел** — в сети распределенной АСУ внешний узел, в котором размещается аппаратура, связанная с технологическим процессом, человеком-оператором или другой АСУ.

**Узел** — в вычислительной сети место сосредоточенного расположения аппаратуры одной или нескольких станций.

**Уровень, или слой архитектуры** — набор функций, выделенный по принципам слоистой архитектуры.

**Функциональный граф АСУ** — ориентированный граф, в котором вершины соответствуют операциям, а дуги — информационной связи между операциями. Вершины-истоки и вершины-стоки графа соответствуют терминальным узлам распределенной АСУ.

## Литература

1. С.Д.Бушуев, В.С.Михайлов. Автоматика и автоматизация производственных процессов М., «Высшая школа», 1986.
2. Проектирование систем автоматизации технологических процессов. Справочное пособие/ А.С.Клюев и др. Под ред. А.С.Клюева 2-е изд.-М., «Энергоатомиздат», 1985.
3. В.Г.Оливер, Н.А. Оливер Компьютерные сети Учебник. С.-Петербург, «Питер», 1999.
4. Стандарты по ЛВС. Справочник.-М.:Радио и связь,1990
5. Сети ЭВМ. Справочник.-М.: 1990
6. Пятибратов и др. ВС, сети и телекоммуникации –М.:ФИС,1998.
7. Высокопроизводительные сети. Энциклопедия пользователя. Киев, 1998.

## **Содержание**

<b>Лекция №1</b>	Архитектура распределенных АСУ. Функционально программная структура
<b>Лекция №2</b>	Физическая структура. Интерфейсы и протоколы
<b>Лекция №3</b>	Архитектура распределенных АСУ. Каналы связи
<b>Лекция №4</b>	Устройства связи с технологическим процессом. Устройство связи оператора с АСУ
<b>Лекция №5</b>	Функции и программы распределенных АСУ. Физический канал
<b>Лекция №6</b>	Управление магистралью. Управление сетью
<b>Лекция №7</b>	Управление сеансами обмена и представлением данных
<b>Лекция №8</b>	Операционные системы
<b>Лекция №9</b>	Проектирование распределенных АСУ. Математические модели распределенных АСУ
<b>Лекция №10</b>	Оптимальное проектирование распределенных АСУ

## **Лекция №3**

### **Аппаратура распределенных АСУ. Каналы связи**

#### **План**

1. Аппаратура современных АСУ.
2. Каналы связи
3. Аппаратура современных АСУ

#### **1. Аппаратура современных АСУ**

Аппаратуру современных АСУ можно с полным основанием отнести к новому поколению, использующему все, последние достижения технологии микроэлектроники. Интегральные микросхемы (ИМС) либо полностью вытеснили дискретные электронные схемы, либо наложили совершенно новый отпечаток на неэлектронные (механические, электромеханические, электромагнитные, оптические и др.) узлы аппаратуры и позволили в ряде случаев реализовать новые принципы.

Применение нового поколения аппаратурных средств не ограничивается собственно распределенными АСУ, а распространяется на все структуры, но именно эти новые средства, прежде всего микропроцессоры, позволили развивать распределенные структуры.

Средства передачи данных являются неотъемлемым атрибутом распределенных АСУ. К новой аппаратуре передачи данных относятся высокоскоростные коаксиальные и оптические линии связи, высокочастотная канальная аппаратура, в том числе и оптоэлектронная, и специальные цифровые ИМС, реализующие протоколы связи.

Новая аппаратура обработки данных - это универсальные и специализированные микропроцессоры. Последние применяются либо в качестве регуляторов, реализующих заранее заданные законы регулирования, либо в качестве управления устройств различными компонентами АСУ-связью с сетью, терминалами и т. п. Дешевизна микропроцессоров позволила широко оснащать ими различные терминалы - датчики, местные устройства управления, устройства общения между оператором и системой - и таким образом эффективно распределять функции АСУ между различными ее узлами.

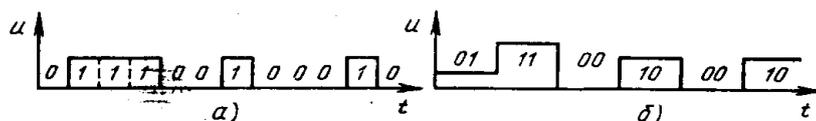
Устройства связи с процессором обновились прежде всего вследствие возможности введения ИМС в аппаратуру, работающую с микропроцессорами, обрабатывающими данные, - аналого-цифровые и цифро-аналоговые преобразователи, коммутаторы; в промежуточные преобразователи электрических сигналов, устанавливаемые между датчиками и ЭВМ; в местные устройства управления аппаратурой связи с процессом.

Наконец, практически полностью обновилась аппаратура связи с оператором. Для ввода команд почти исключительно применяются современные конструкции клавиатур и в меньшей степени средства аналогового ввода, для вывода на оператора - прежде всего дисплеи, а в качестве дополнительного средства - печатающие устройства или графопостроители. В современных комплексах средств для построения распределенных АСУ основной аппаратурной единицей является набор аппаратуры, размещаемый в одном каркасе и присоединяемый к сети передачи данных со своим адресом. Эту аппаратурную единицу чаще всего (хотя и не всегда) называют станцией. Хотя далее дается другое определение понятия станции, основанное на логической структуре сети передачи данных, известное сходство обоих понятий позволяет сохранить за нами один и тот же термин «станции».

## **2. Каналы связи**

Физический канал связи состоит из передающей линейной аппаратуры - передающего и приемного линейных узлов и соединяющей их линии. Хотя, экономически оптимальным считается равномерное распределение стоимости между оконечной аппаратурой и линией связи, линия связи в АСУ зачастую оказывается лимитирующим элементом физического канала.

Импульсы, которые передают цифровую информацию по каналу, занимают в зависимости от их формы и частоты следования определенный спектр частот. Этот спектр совпадает с полосой частот канала только в том случае, когда полоса частот, занимаемая сигналом в линии, начинается с нулевой частоты, т. е. передача осуществляется видеоимпульсами в основной полосе частот канала. Цифровой сигнал после усиления и фильтрации в передатчике посылается в линию.



**Рис.18. Дискретная модуляция:**

**а – бивалентная; б – четырехвалентная; и – уровень сигнала в линии**

Приемник, получив сигнал, распознает его дискретное состояние, а следовательно, и передаваемый им цифровой код, с помощью детектора.

В тех случаях, когда спектр сигнала не совпадает с полосой сигнала необходимо преобразование спектра сигнала, что выполняется с помощью модуляции.

Самый простой вид модуляции - бивалентная. При бивалентной модуляции сигнал, передающий бит, может принимать одно из двух возможных состояний (рис.18,а).

При мультивалентной модуляции состояние сигнала определяется не одним битом, а их сочетанием. Так, например, если состояние сигнала определяется сочетанием двух битов, то он может принимать одно из четырех значений (рис.18,б).

Использование для передачи сигнала той части полосы частот, которую пропускает физический канал, позволяет уплотнять канал по частоте, применяя его для организации связи между несколькими абонентами, присоединенными одной и той же оконечной аппаратуре. Данные по линии чаще, передаются последовательно (битами). Использование одного физического канала для обмена данными между различными станциями сети распределенной АСУ достигается только разделением времени канала. Различным станциям канал предоставляется с помощью сетевых средств.

Передачу по электрическим цепям можно вести в их основной полосе частот, но нередко из соображений, связанных с гальванической развязкой,

переходят к работе в некоторой смещенной полосе. Чаще всего при этом пользуются стандартным, телефонным каналом.

Стандартный телефонный канал в соответствии с рекомендациями Международного консультативного комитета телеграфии и телефонии (МККТТ) занимают полосу частот от 300 до 3400 Гц, но допускается и пользование каналами с суженной полосой - до 2400 или 2000 Гц. Для передачи данных отводится либо вся полоса телефонного канала (с небольшим расширением) 300 - 3480 Гц, либо полосы, остающиеся неиспользованными в верхней части телефонного канала при сужении его полосы.

Рекомендуемое нормами МККТТ деление полосы 300 - 3480 Гц на частные каналы для передачи данных приведено на рис.19 допускается шесть вариантов деления полосы: максимум на 26, 13, 8, 6, 2 или 1 канал. При передаче данных по телефонной линии непосредственно видеоимпульсами пользуются скоростями передачи 600 и 1200 бод.

Скорость передачи двоичных данных по каналу связи  $V$  измеряется числом битов, передаваемых за 1 с. Скорость дискретной модуляции  $B$ , измеряемая в бодах, совпадает со скоростью передачи по каналу только при бивалентной модуляции, а при мультивалентной она может быть ниже, чем скорость передачи. При бивалентной модуляции скорость передачи и скорость дискретной модуляции равны:

$$V=B=1/T, \quad (1)$$

$T$  - длительность интервала, занимаемого одним битом.

Для мультивалентной модуляции скорости  $V$  и  $B$  связаны соотношением

$$V= B \text{Log}_2 m \quad (2)$$

где  $m$  -валентность (число возможных дискретных состояний) сигнала.

Таким образом, повышением валентности модуляции, может существенно увеличить скорость передачи, в канале, не расширяя его полосы пропускания, а следовательно, улучшить его показатели.

Отношение между верхней границей скорости дискретной модуляции  $B_m$ , и шириной полосы канала  $df$  зависит от рода модуляции. Теоретически коэффициент  $k$  в соотношении

$$k=B_m/df \quad (3)$$

равен 2 для видеоимпульсов и 1 для амплитудной, фазовой и частотной модуляции. На практике, однако, он ограничивается значениями 1,4 и 0,7 соответственно.

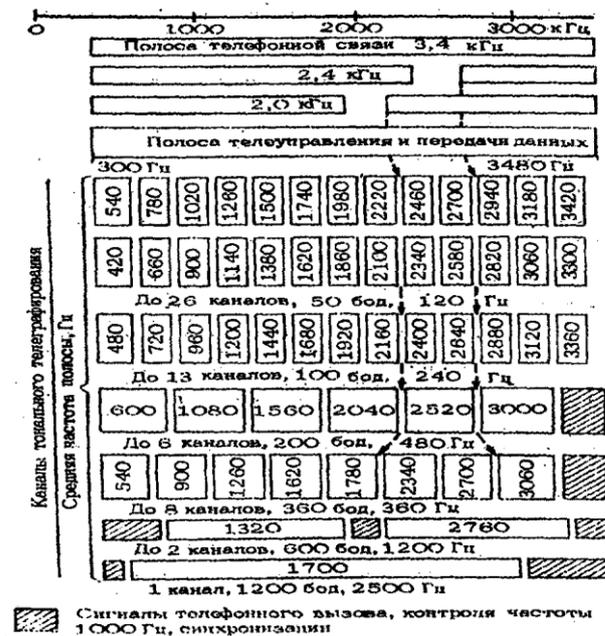


Рис. 19. Деление стандартного телефонного канала на полосы по рекомендациям МККТТ

Качество аппаратуры канала связи удобно характеризовать произведением:

$$p = dfL \quad (4)$$

где  $L$  - дальность передачи, а качество канала дискретной передачи данных в целом - произведением

$$b = BL \quad (5)$$

В каналах связи распределенных АСУ нередко применяются такие же линии, как и для телефонной связи внутри предприятия, или же просто пользуются парами проводов в кабелях, ранее проложенных для телефонной сети.

Кабели с воздушно-бумажной изоляцией, применяемые для внутризаводской телефонной связи, характеризуются волновым сопротивлением, убывающим с частотой: оно составляет 1 - 1,5 кОм на частоте 300 Гц и 300 - 400 Ом на частоте 4 кГц. Погонное затухание этих кабелей растет с частотой: с 0,5 - 1 дБ/км при 300 Гц до 2 - 3 дБ/км при 4 кГц. Погонная емкость этих кабелей незначительна: при воздушно-бумажной изоляции она составляет 40 - 80 нФ/км, а при полиэтиленовой - 45 нФ/км на частоте 800 Гц.

Качество физического канала с телефонной линией, по которой ведется передача видеоимпульсами, характеризуется показателем порядка 0,2 км Мбит/с.

Для модуляции несущей частоты на передающей стороне и демодуляции на приемной стороне применяют устройства, получившие название модемов (модуляторы - демодуляторы). Качество модемов характеризуется двумя показателями: отношением средней передаваемой мощности к мощности шума в полосе

$$df = 2/T, \quad (6)$$

где  $T$  - интервал, занимаемый одним битом, и отношением  $k$  скорости передачи  $V$  к полосе частот  $df$  по (3).

Максимальная скорость передачи данных по телефонному каналу с применением модемов достигает 9600 бит/с.

В последние годы в целях повышения быстродействия и дальности связи в распределенных АСУ улучшают качество самой линии связи. Для повышения помехоустойчивости, а следовательно, и скорости передачи применяют экранированные симметричные витые пары. Выпускаемые промышленностью радиочастотные симметричные кабели со скрученными жилами имеют погонное затухание в зависимости от марки кабеля от 0,006 - 0,013 дБ/км на частоте 1 МГц до 0,13 - 0,28 дБ/км на 100 МГц. Ввиду значительно большей стоимости этих линий связи их применяют при сравнительно небольших расстояниях или в магистральных структурах, требующих относительно небольшой затраты проводов. Качество каналов связи на таких кабелях характеризуется показателем, достигающим  $\sigma = 1,5$  км Мбит/с. Впрочем, на практике часто в целях повышения помехоустойчивости на указанный верхний предел не выходят, ограничиваясь значением  $\sigma = 0,8$  км Мбит/с.

Военный стандарт США 1553В предусматривает применение для последовательной передачи данных экранированной скрученной пары проводов. Требуется не менее 13 витков на метр, погонная емкость не более 100 пФ/м.

Оба конца кабеля замыкаются на сопротивление от 70 до 85 Ом. Кабель соединяется с терминалом через два резистора сопротивлением по 55 Ом, последовательно с которыми ставится разделительный трансформатор 1: 1,4.

Суущественно лучшие электрические характеристики имеет коаксиальный кабель, у которого электромагнитное поле не выходит, за пределы цилиндрической экранированной оболочки, т. е. полностью лежит внутри кабеля. Коаксиальные кабели широко применяются в последние годы в качестве линии связи в промышленных АСУ. В отечественной практике пользуются преимущественно кабелем РК-75, т. е. с волновым сопротивлением 75 Ом. Погонная емкость этого кабеля 67 пФ/м.

В США пользуются коаксиальными кабелями двух типов: с волновым сопротивлением 93 Ом (так называемый кабель высшего качества) и телевизионным кабелем с волновым сопротивлением 75 Ом. Видеоимпульсы, как правило, передают по кабелю первого типа, кабель второго типа служит для передачи модулированных сигналов.

Типичным примером применяемого в распределенных АСУ коаксиального кабеля может служить Cellflex 1/4" Cu24 с волновым сопротивлением 75 Ом, погонной емкостью 54 пФ/м, диаметром внутреннего провода 1,5 мм, наружного проводника 10 мм; затухание составляет 2,5 дБ/км на частоте 500 кГц и 3,5 дБ/км на частоте 1 МГц. Реально достигнутый показатель качества физического канала при применении такого кабеля составляет  $\sigma=1,5$  км Мбит/с.

Стандарт ЕСМА-80 на линию связи для локальной сети с передачей в основной полосе предусматривает применение коаксиального кабеля с волновым сопротивлением 50~2 Ом. Затухание кабеля на длине 500 м до 8,5 дБ на частоте 10 МГц до 6 дБ на частоте 5 МГц. Передача ведется в основной полосе с рабочей скоростью 10 Мбит/с.

Каналы передачи данных по коаксиальному кабелю характеризуются показателем  $\alpha$ , примерно равны 1 - 5 км Мбит/с, хотя могут быть получены значения до 600 км Мбит/с, как, например, в системе передачи данных американской фирмы Interactive Systems Inc.

Заметим, что линии из одинарного или двойного коаксиального кабеля используются в большинстве современных промышленных распределенных АСУ.

При построении распределенных АСУ для управления бортовыми системами корабля использован триаксиальный кабель, состоящий из трех концентрических проводов этот кабель позволил обеспечить передачу данных со скоростью 10 Мбит/с на расстояния до 300 м, что соответствует показателю качества дискретного канала  $\sigma=3$  км Мбит/с.

Оптические линии передачи данных являются сравнительно новым техническим средством. Носителем сигнала здесь служат близкие к монохроматическим световые колебания, которые модулируются передаваемым сигналом по амплитуде. Свет может проходить только по прозрачному веществу - стеклу или прозрачной пластмассе. К достоинствам оптических каналов передачи данных по сравнению с электрическими относятся: нечувствительность к внешним электромагнитным полям, колебаниям температуры, отсутствие коротких замыканий; меньшие габариты по сравнению с медными проводами; высокая пропускная способность (более 30 Гбит/с); большое значение показателя  $\rho$  (более 1 ГГц км). Для преобразования электрического сигнала в оптический в передающем линейном узле используют либо светодиоды, либо полупроводниковый лазер.

Световая мощность, отдаваемая светодиодом, пропорциональна модулирующему электрическому току и в современных приборах не превышает 0,1 мВт. Светодиоды стоят сравнительно дешево, имеют длительный срок службы, однако по сравнению с лазерами страдают серьезным недостатком: они излучают свет не в одном направлении, а под весьма широким углом.

Полупроводниковые лазеры обеспечивают генерирование когерентных световых колебаний с очень узким спектром. Светоотдача лазера не пропорциональна модулирующему электрическому току: она растет быстрее после некоторого, порогового значения. Средняя отдаваемая мощность лазера составляет 0,6 мВт при длине волны 820 нм.

Основным материалом для изготовления световодов служит двуокись кремния (кремнезем), являющаяся основой многих минералов или входящая в их состав, а потому в отличие от меди отнюдь не являющаяся дефицитным сырьем. Различают три вида световодов: со ступенчатым профилем, градиентные и мономодные. Световод со ступенчатым профилем (рис.20,а) имеет стержень из материала с показателем преломления  $n_1$  и оболочку с показателем преломления  $n_2$ . Световые лучи распространяются в световоде, многократно полностью отражаясь от границы между стержнем и оболочкой. Однако полностью отражаются только лучи, падающие на границ под углом удовлетворяющим соотношению:

$$\cos \varphi \leq n_1/n_2 \quad (7)$$

Следовательно, распространяться по световоду будут только лучи входящие в световод под углом  $\varphi \leq 2\varphi_1$ , где угол  $\varphi_1$  определяется коэффициентом преломления на входе в стержень световода.

Этим условиям хорошо удовлетворяют лучи, испускаемые лазером, но в меньшей степени - лучи от светодиода.

Величина  $\sin \varphi_1$ , носящая название **числовой апертуры**, является существенной характеристикой световода.

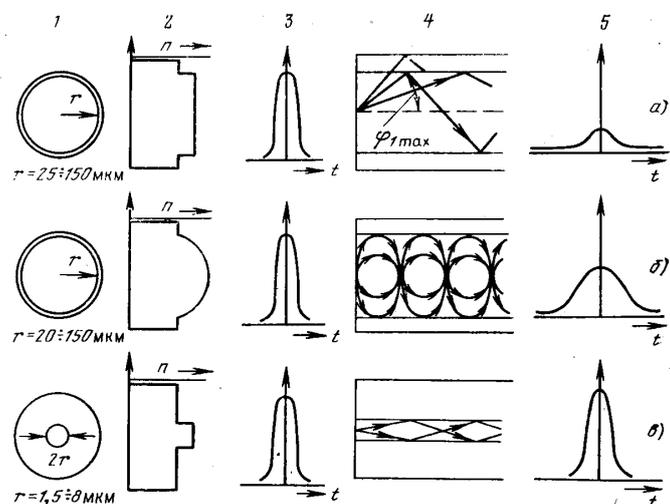


Рис.20. Принцип распространения луча в световодах:

**а** – ступенчатый профиль световода; **б** – градиентный световод; **в** – мономодный световод; **1** – поперечный разрез световода; **2** – изменение показателя преломления по диаметру; **3** – входной импульс; **4** – канал передачи; **5** – выходной импульс

В градиентных световодах (рис20,б) полное отражение не используется: здесь лучи непрерывно отклоняются в сторону оси стержня так что они распространяются почти по синусоидальным траекториям. Показатель преломления стержня изменяется в функции радиуса:

$$n(r) = n_1 / (1 - d(r^2/a^2)), \quad (8)$$

где  $n$  - показатель преломления оболочки;  $d$  - нормированная разность показателей преломления:

$$d = (n_1 - n_2) / n_1, \quad (9)$$

$a$  - радиус световода.

Наконец, в мономодном световоде (рис.20,в) возможно распространение только отдельных световых лучей. Так как внутренний стержень этого световода имеет диаметр всего 2-4 мкм, световод пропускает практически только осевые лучи. Мономодные световоды превосходят по пропускной способности световоды первых двух типов, но на практике предпочитают применять градиентные световоды ввиду того, что их изготовление более просто.

В своем распространении по световодам лучи претерпевают затухание и размывание. Затухание сигнала, входящего в световод, обусловлено в основном рассеянием и поглощением в световоде. Рассеяние в световоде, называемое также рассеянием Рэлея, пропорционально  $\lambda^{-4}$  (где  $\lambda$  - длина волны света). Кроме того, затухание имеет место в сопряжениях на входе в световод и на выходе из него. Поэтому затухание при передаче данных тем больше, чем из большего числа сопряженных между собою отрезков световодов состоит линия передачи. Затухание определяется (в децибелах) как логарифм отношения интенсивностей сигнала на входе и на выходе:

$$\alpha = 10 \lg \frac{I_{вх}}{I_{вых}} \quad (9)$$

Общее затухание оптического канала передачи данных составляет

$$\alpha_0 = \alpha_{вх} + l_{\mu} + n\nu + \alpha_{вых}, \quad (10)$$

где  $\alpha_0$  - общее затухание;  $\alpha_{вх}$  - затухание на входном сопряжении в световоде;  $l$  - длина световода;  $\mu$  - затухание на единицу длины световода;  $n$  - число промежуточных сопряжений;  $\nu$  - затухание в промежуточном переходе;  $\alpha_{вых}$  - затухание в сопряжении на выходе из световода.

У лучших из выпускаемых в настоящее время типов стеклянных световодов затухание  $\mu$  не превышает 1дБ/км, а это означает, что на длине световода 1 км теряется 20% мощности. В [7] указывается, что если бы морская вода была столь же прозрачна, то можно было рассмотреть дно океана в самом глубоком его месте.

Затухание  $\mu$  в материале световода зависит также от длины волны света. По мере увеличения длины волны затухание убывает, причем при некоторых значениях существуют так называемые окна, где затухание особенно мало; такое окно имеется, например, в диапазоне  $\lambda \approx 820$  нм, следующее - при 1040 нм. С этим обстоятельством естественно связано стремление так конструировать передатчики, чтобы они имели максимальное излучение, а приемники - так, чтобы они имели максимальную чувствительность на этих волнах.

Размывание сигнала в световодах обусловлено двумя причинами: одна из них связана со свойствами материала, другая - с временем распространения света. Материал влияет на размывание сигнала в связи с тем, что свет, излучаемый лазером или светодиодом, не является чисто монохроматическим, а представляет собой смесь разных частот в некотором диапазоне волн, составляющем 2 нм для лазера и 40 нм для светодиода. Поскольку показатель преломления  $n_2 \sim$  зависит от длины волны  $\lambda$ , различные составляющие по-разному преломляются. При применении лазера размывание сигнала, обусловленное свойствами материала, незначительно.

Так как световые лучи входят в световод под различными углами в пределах  $\beta \leq 2\phi_1$ , возникает разница в длине проходимого лучами пути, а следовательно, во времени распространения. Для световода со ступенчатым профилем эта разница составляет

$$dt = (l/c) n_2 d \quad (11)$$

Для градиентных световодов

$$\Delta t = (l/c) n_2 d / 2 \quad (12)$$

Таким образом, величина  $\Delta t$  пропорциональна длине световода  $l$ . Так как пропускная способность световода зависит от  $\Delta t$ , то она зависит и от длины световода. Разница во времени распространения у градиентных световодов меньше, чем у световодов со ступенчатым профилем, отчего первые находят значительно более широкое применение на практике.

Затухание выпускаемого в продажу световодного кабеля  $d_2 \geq 4$  дБ/км. Если необходимо связать с помощью волоконной оптики (без применения промежуточных усилителей) устройства, отстоящие больше чем на 10 км, то приемный линейный узел должен быть способен регистрировать очень малые сигналы. Кроме того, он должен иметь максимальную чувствительность на волне, излучаемой передающим линейным узлом. На практике в качестве приемных линейных узлов применяют фотодиоды с р-і-п- и-структурой или же лавинные фотодиоды. Фотодиоды с р-і-п- и-структурой состоят из одной полупроводниковой, области с р-проводимостью и одной с п-проводимостью (кремний), разделенных высокоомной нелегированной областью. Падающий световой поток вызывает в диоде электрический ток, пропорциональный интенсивности света. Фотодиоды с р-і-п-структурой очень надежны и дешевы, но менее чувствительны, чем лавинные диоды.

В лавинном диоде падающий световой поток вызывает появление свободных электронов, причем в результате цепной реакции число носителей заряда растет лавинообразно. Эти диоды очень чувствительны.

Дальность действия оптической системы передачи данных зависит от мощности передающего узла, затухания в кабеле, чувствительности приемного узла, скорости передачи (в битах в секунду). Важнейшим фактором является затухание кабеля, ибо оно, как правило, налагает ограничение на дальность передачи. При некоторых идеальных предпосылках, а именно при применении

полупроводникового лазера в передающем линейном узле (мощность больше 10 мВт), градиентного световода (погонное затухание до 2 дБ/км) и лавинного диода в приемном линейном узле, можно уверенно передавать данные со скоростью  $10^6$  бит/с на расстояние 70 км. В реальных условиях (погонное затухание около 5 дБ/км) передача с такой скоростью возможна на расстояние 10 - 15 км.

В зарубежных промышленных АСУ оптические каналы передачи данных находят с каждым годом все более широкое применение, постепенно заменяя каналы с коаксиальным кабелем. На коротких расстояниях нередко пользуются низкокачественной волоконной оптикой: до 30 м - кабелями с погонным затуханием свыше 100 дБ/км, от 30 до 100 м - кабелями с затуханием от 10 до 100 дБ/км. Числовая апертура у световолокон с высокими потерями (свыше 100 дБ/км) обычно превышает 0,5, у световолокон со средними потерями (10 - 100 дБ/км) лежит в пределах от 0,25 до 0,5, а у высококачественных световодов (потери ниже 10 дБ/км) - в пределах от 0,1 до 0,25.

Весьма типичны для рынка США световолоконные кабели, выпускаемые фирмой Sicoг. В кабеле - от одного до 10 волокон. Числовая апертура равна 0,21, погонное затухание для волны 820 нм - от 6 до 10 дБ/км. Показатель  $n$  составляет 200 МГц км для каналов на стандартных кабелях и 400 МГц км для каналов на кабелях с улучшенными характеристиками.

До сих пор рассматривалась аппаратура, реализующая функции на уровне физического канала. Для решения задач более высоких уровней в последние годы применяют микропроцессоры, которые по исполняемой ими роли получили название связных процессоров. Программа их работы, реализующая протоколы связи (см. гл. 4), обычно фиксирована и хранится в постоянном запоминающем устройстве в виде микропрограмм.

С развитием технологии ИМС получила распространение программно-аппаратурная форма реализации канального протокола с повышенной долей участия аппаратурных средств. Программы, реализующие протоколы, уже заложены в аппаратуре, программирование которой заключается лишь в выборе того или иного протокола. Одно из первых устройств, разработанных для этой цели фирмой Intel в США, названо UART - Universal Asynchronous Receiver - Transmitter, т. е. «универсальный асинхронный приемопередатчик». Устройство выполнено на одном кристалле. Оно реализует асинхронное преобразование параллельного кода микропроцессорного слова в последовательный код линии и обратное преобразование кода линии в параллельный код микропроцессора.

Фирма Intel выпускает одноплатное программируемое интерфейсное устройство USART, способное реализовать большинство канальных протоколов передачи данных, как синхронных, так и асинхронных. Пользователь задает применяемый протокол в программе связного микропроцессора. Устройство USART передает в связной микропроцессор сигнал о начале передачи с данной станции и о приеме каждого знака для данной станции. Микропроцессор всегда может определить состояние, в котором находится интерфейсное устройство в данный момент.

### 3. Аппаратура обработки данных

В современных АСУ данные обрабатываются преимущественно в цифровом виде.

Аппаратура цифровой обработки различается прежде всего способом задания реализуемой ею программы. Для решения фиксированных, весьма узких логических или вычислительных - задач применяют специализированные цифровые устройства, в которых программа задана аппаратурно. Такие схемы часто называют

к о н т р о л л е р а м и.

Функционально контроллер в ответ на каждое сочетание двоичных сигналов (код), подводимое к его входам параллельно или последовательно, выдает на одном или нескольких выходах определенный набор (наборы) двоичных сигналов.

Контроллеры служат, во-первых, для решения внутрисистемных задач: управления связью, т. е. реализации протоколов одного или нескольких уровней архитектуры (контроллеры связи); управления внешними устройствами, такими как коммутаторы переменных, дисплеи, печатающие устройства, измерительные приборы и т. п. (периферийные контроллеры); наконец, диагностики системы. Кроме того, контроллеры применяют также для решения прикладных задач, особенно на массовых объектах. Среди них объекты, требующие логического управления, такие как станки, конвейеры, лифты, насосы, электрические подстанции, железнодорожные стрелки, светофоры, стиральные и моечные машины, и объекты с непрерывными процессами - устройства отопления, кондиционеры, небольшие печи, реакторы периодического действия.

Конструктивно аппаратурные контроллеры выполняются в настоящее время на ИМС, причем устройство обычно реализуется на одной или нескольких ИМС и во всяком случае на единственной плате.

Первым шагом к отказу от специализации при разработке и изготовлении контроллеров является микропрограммирование, при котором отдельные специфические операции управления процессом закладываются в устройство в виде микропрограмм, а программирование при построении АСУ заключается в задании последовательности микропрограммных операций. Примером микропрограммного автомата для автоматического регулирования процессом может служить цифровой регулятор Mudicon, реализованный на основе 16-разрядной микро-ЭВМ типа 2901.

Следует, однако, отметить, что с развитием микропроцессорной техники все большую часть функций, прежде выполнявшихся аппаратурными контроллерами, возлагают на универсальные программируемые устройства, ядром которых является микропроцессор.

Микропроцессор, дополненный запоминающими устройствами (оперативным, постоянным, полупостоянным), устройствами микропрограммного управления и ввода-вывода, образует микро-ЭВМ, которая служит основным устройством обработки данных в современных

распределенных АСУ, по крайней мере на нижнем их уровне, связанном с процессом.

Задачи, которые ранее возлагались на аппаратурные контроллеры, в настоящее время передаются программируемым контроллерам, выполненным в виде микро-ЭВМ и называемым поэтому микроконтроллерами. Разумеется, программируемые микроконтроллеры, будучи универсальными устройствами, имеют меньшее быстродействие, но зато они по этой же причине дешевле в производстве.

Если программа функционирования микроконтроллеров записывается в постоянном запоминающем устройстве (ЗУ), т. е. определяется на стадии изготовления микроконтроллера, то тогда с точки зрения пользователя программируемые микроконтроллеры не отличаются от аппаратурных.

Однако программируемые контроллеры наделены некоторыми дополнительными функциональными возможностями, которые обеспечивает входящий в их состав микропроцессор, а именно функциями арифметической обработки данных и отсчета времени.

Наряду с постоянными ЗУ для хранения программ применяются также перепрограммируемые (ППЗУ) и оперативные (ОЗУ) ЗУ, что позволяет пользователю самому вносить и исправлять программу.

В последние годы получили распространение как более надежные однокристальные 8- и 16-разрядные микроконтроллеры, причем доля последних с каждым годом увеличивается.

Число двоичных входов колеблется от 16 до 2048, но может достигать и значительно больших значений, например 64К. Емкость памяти колеблется от 2К до 64К, причем она может по-разному распределяться между постоянной, полупостоянной и оперативной частями памяти. В некоторых последних разработках память достигает 96К с возможностью расширения до 256К. При хранении программы в ОЗУ последние выполняются энергонезависимыми, например на ферритовых сердечниках или магнитной проволоке.

Характеристики микроконтроллеров даны подробно в литературе. Вычислительные возможности микроконтроллеров по существу стирают границу между микроконтроллерами и управляющими микро-ЭВМ, из-за чего микро-ЭВМ, предназначенные для регулирования ТП, нередко также называют микроконтроллерами. В качестве цифровых регуляторов ТП используются микро-ЭВМ общего назначения. Специализация микро-ЭВМ, предназначенной для автоматического регулирования ТП, проявляется лишь в том, что в ее ПЗУ или ППЗУ вносятся программы регулирования.

Цифровые регуляторы на микропроцессорах программируются для одно- или многоконтурного регулирования. Как пример одноконтурного цифрового регулятора на микропроцессорах можно назвать плату комплекса NAF-Unic шведской фирмы SAAB Scania или 5260К американской фирмы Taylor Instruments. Структурная схема регулятора 5260К показана на рис.21.

Сигнал регулируемой переменной  $x$ , пройдя через разделительный усилитель У1 и коммутатор КМ, поступает на вход аналоговой схемы сравнения СС. На второй вход схемы сравнения СС через цифро-аналоговый преобразователь

ЦАП подается задание  $x_0$ , из микропроцессора МП типа RCA1802. Знак с выхода схемы сравнения вводится в виде двоичного сигнала на микропроцессор. Управляющее воздействие, вычисленное в МП по избранному закону регулирования, выводится через ЦАП как аналоговый сигнал  $u$  на фиксатор  $\Phi$ , а оттуда - на исполнительный механизм.

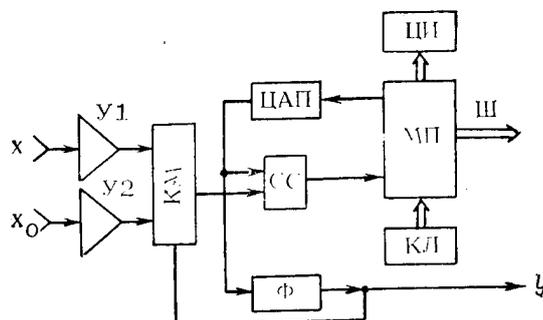
Задание  $x$ , может вводиться также помимо микропроцессора, дистанционно в аналоговой форме через разделительный усилитель У2. Задание и параметры настройки регулятора вводятся в МП с клавиатуры КЛ и контролируются по цифровому индикатору ЦИ. Микропроцессор связан с остальной системой управления многопроводной шиной Ш.

Нередко на одну и ту же микро-ЭВМ возлагают кроме вычисления управляющего воздействия также линеаризацию входных сигналов, обнаружение отклонений, реализацию связи между контурами регулирования, переключения в схеме управления, логическое (цикловое) управление, связь с магистралью; таковы, например, программы 32-контурного цифрового регулятора комплекса Centum японской фирмы «Екогава дэнки». Столь же универсальны функции микро-ЭВМ 8-контурных цифровых регуляторов комплекса Micon MDC)C200.

Цифровой регулятор комплекса P4000 английской фирмы George Kent, Ltd снабжен набором стандартных программ, записанных в полупостоянном ЗУ. В этом наборе - программы регулирования по П-, ПИ-, ПД-, ПИД-законам, с зоной нечувствительности или без нее, программы арифметических операций, например для преобразования (извлечения квадратного корня) подводимого извне задания и регулируемой переменной, ограничения задания или управляющего воздействия. Общая программа управления на конкретном объекте собирается одним из трех способов: с передней панели самого регулятора заданием адресов и настройкой регулирования с помощью переключателей с контролем по цифровому индикатору; с клавиатуры портативного терминала, который можно подключать к передней панели регулятора; с центрального пульта АСУ.

Построение регуляторов на базе микропроцессора делает их чрезвычайно гибкими и удобными в эксплуатации, расширяет их функциональные возможности. Так, устройство сигнализации отклонений и электромеханические или полупроводниковые реле для включения различных электрических аппаратов могут дополнять регулятор лишь тогда, когда это требуется на данном объекте. Цифровое управление регулятором делает его настройку более удобной, чем у аналоговых регуляторов: входные переменные, управляющие воздействия, задания, положение исполнительных органов представляются оператору в цифровой форме. Режимы управления объектом и настройки регулирования устанавливаются кнопочными (клавишными) переключателями. В некоторых цифровых регуляторах (например, в регуляторах семейства 80 американской фирмы Doric Scientific имеется возможность программного изменения задания в функции времени. Зависимость задания от времени аппроксимируется ломаной линией, содержащей до девяти отрезков. Сенсорная панель позволяет изменять

программу, включая и отрезок, обрабатываемый в данный момент времени. Оператор может изменить длительность участка, задания на концах отрезков, пропускать отдельные участки, начинать программу с любой промежуточной точки. Можно ввести в программу выдачу импульса или включение реле в определенный момент обработки запрограммированного задания.



**Рис.21. Структурная схема цифрового регулятора 5260R:**

*KL* — клавиатура; *KM* — коммутатор; *МП* — микропроцессор; *CC* — схема сравнения; *U1*, *U2* — усилители; *Ф* — фиксатор; *ЦАП* — цифро-аналоговый преобразователь; *ЦИ* — цифровой индикатор; *Ш* — шина;  $x$  — регулируемая переменная;  $x_0$  — уставка регулирования;  $y$  — управляющее воздействие

В цифровых регуляторах удобно вводить задания кодом из управляющей ЭВМ. Во избежание резких изменений управляющего воздействия на выходе регулятора при введении нового задания регулятор отслеживает положение исполнительного механизма, благодаря чему достигается переход на работу по новому заданию.

Еще одна функция, реализуемая программным путем в цифровых регуляторах, - регулирование отношения двух технологических переменных. Оператор может наблюдать в цифровой форме либо заданное отношение, либо задание ведомому контуру регулирования. Возможен также режим, при котором задания двум контурам регулирования находятся в определенном постоянном отношении.

Иногда большие программы управления сами хранятся во внешних ЗУ, преимущественно на гибких магнитных дисках. Однако чаще в целях повышения надежности стремятся исключить гибкие диски из АСУ, более же надежные и весьма перспективные внешние ЗУ на цилиндрических магнитных доменах только начинают получать применение в АСУ.

Наряду с цифровыми устройствами обработки технологических данных успехи микроэлектроники позволяют развивать и аналоговые устройства, в первую очередь аналоговые регуляторы. Эти устройства включаются в комплексы средств, предназначенных для построения распределенных АСУ, либо с расчетом на более или менее автономную работу, либо с возможностью связи с системой.

Так, в составе комплекса PIDCOM, выпущенного в 1977 г. американской фирмой Beckman, имелись одноконтурные аналоговые регуляторы типа 8800 и

микропроцессоры, связывающие эти регуляторы с магистралью; один микропроцессор рассчитан на соединение с магистралью до 64 регуляторов.

Аналоговые регуляторы двух видов - для ПИ- и для ПИД-регулирования - входят также в состав комплекса Р4000. Каждый регулятор выполнен в виде вставной платы. Платы могут содержать схемы для простых аналоговых вычислений - умножения, деления, извлечения квадратного корня. В этом комплексе тоже предусмотрена связь регуляторов с магистралью системы путем цифрового преобразования аналоговых сигналов.

### Контрольные вопросы

1. Расскажите о современной аппаратуре, используемой в АСУ. [Л.7, 35-36].
2. Из какой аппаратуры состоит физической канал? [Л.7, 36-37].
3. Перечислите виды модуляции. [Л.7, 37-38].
4. Напишите формулы, связывающие скорость дискретной модуляции при бивалентной и мультивалентной модуляции. [Л.7, 38-40].
5. Что улучшают в целях повышения быстродействия и дальности связи в последние годы? [Л.7, 40-43].
6. Достоинства оптических каналов передачи данных по сравнению с электрическими. [Л.7, 44-47].
7. Преимущественно в каком виде обрабатываются данные в современных АСУ? [Л.7, 48-49].
8. Роль микро-ЭВМ в обработке данных АСУ. [Л.7, 48-49].
9. Расскажите о современных цифровых регуляторах. [Л.7, 50-51].
10. Применение аналоговых регуляторов в распределенных АСУ. [Л.7, 52-53].

### Лекция №4

#### Устройства связи процессом. Устройства связи оператора с АСУ

##### План

1. Устройства связи с процессом.
2. Схемы аналого-цифрового преобразователя.
3. Устройства связи оператора с АСУ

#### 1. Устройства связи с процессом

К устройствам связи АСУ относятся устройства ввода в систему сигналов, соответствующих значениям технологических переменных, и устройства вывода управляющих воздействий. Принципы построения и тех и других устройств в основном сформировались на предшествовавших этапах развития АСУ и переход к микропроцессорам и распределенным структурам АСУ наложил отпечаток лишь на отдельные их особенности.

Аналоговые устройства связи с процессом подробно освещены в литературе.

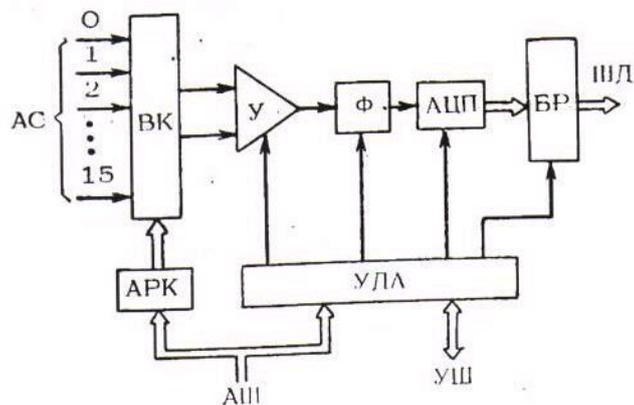
Для передачи аналоговых сигналов от датчиков, связанных с процессом, в аппаратуру обработки еще шире, чем прежде, применяют в качестве унифицированного токовой выходной сигнал в виде постоянного тока 0-5, 1-5 или 4-20 мА. При этом сигнал передается по двум проводам, *не* подвергаясь существенному воздействию промышленных помех.

Местный промежуточный преобразователь сигнала датчика, устанавливаемый непосредственно около него, может служить только для получения унифицированного токового сигнала или выполнять функцию измерительного преобразователя, т. е. линеаризовать и масштабировать сигнал. В первом случае можно обойтись без местного источника питания, располагая последний на приемной стороне; это относится, в частности, к резистивным датчикам, включаемым в мостовую схему. При установке датчиков в агрессивной или взрывоопасной среде его вместе с местным промежуточным преобразователем заключают в герметизированный корпус, который одновременно служит экраном.

Датчики переменного тока обычно соединяются с устройством обработки четырьмя проводами: по двум на датчик передается напряжение переменного тока для питания, по двум другим снимается сигнал измерения.

Иногда пользуются датчиками с частотным выходом. Частотный сигнал хорошо подходит для передачи по световодам и в связи с этим рассматривается как весьма перспективный.

Типичная структурная схема ввода аналоговых величин в АСУ с микро-ЭВМ показана на *рис.22*.



**Рис.22. Типичная структурная схема ввода аналоговых сигналов в микро-ЭВМ: АРК - адресный регистр с коммутатором; АС - аналоговые сигналы; АЦП - аналого-цифровой преобразователь; АШ - адресная шина; БР - буферный регистр; ВК - входной коммутатор аналоговых сигналов; У — измерительный усилитель; УДА — устройство дешифратора адреса и логического управления; УШ — управляющая шина; Ф - фиксатор; ШД — шина данных**

Аналоговые сигналы *АС* поступают на входной коммутатор *ВК*, который рассчитан, как правило на 16 однопроводных (несимметричных) или 8 двухпроводных (дифференциальных) входов; часто предусматривается возможность увеличения числа входов до 64. С входного коммутатора модулированный по амплитуде импульсный сигнал поступает на фиксатор *Ф*, хранящий аналоговую величину между операциями ее цифрового преобразования, выполняемыми аналого-цифровым преобразователем *АЦП*. Код выводится на выходной буферный регистр *БР*, откуда через шину данных *ШД* вводится в микро-ЭВМ.

Операциями ввода управляет микро-ЭВМ, посылая коды адреса по адресной шине *АШ* и управления - по управляющей шине *УШ*. Код адреса поступает на адресный регистр коммутатора *АРК* и устройство декодирования адреса и логического управления *УДА*. Это последнее устройство, получающее также код управления, задает коэффициент усиления и, возможно, коэффициенты линеаризации в усилителе *У*, а также управляет работой фиксатора *Ф*, аналого-цифрового преобразователя *АЦП* и выходного буферного регистра *БР*. Аналого-цифровой преобразователь обычно имеет 12 двоичных разрядов, шина данных - 8-разрядная.

Конструктивно узлы ввода оформляются в виде модульных плат с гибридными ИМС. В одних случаях гибридные платы рассчитаны на работу с микро-ЭВМ определенного семейства (например, АОС0816 фирмы National Semiconductor), в других требуют, чтобы пользователь дополнил их соответствующей интерфейсной схемой.

Четко проявляется тенденция к выпуску монолитных ИМС, полностью обеспечивающих функцию ввода группы аналоговых сигналов (например, от 8 до 64) в микро ЭВМ.

Главной проблемой при вводе аналоговых величин является обработка малых сигналов, особенно в промышленных условиях, характеризующихся высоким уровнем помех. Иногда в целях борьбы с помехами и дрейфом применяют

индивидуальные аналого-цифровые преобразователи для каждой технологической переменной. Это позволяет снизить быстродействие преобразователя, т.е. уменьшить его стоимость, или увеличить частоту ввода данных; обойтись без входного коммутатора и фиксатора; обеспечить при последовательной передаче данных полную гальваническую развязку цепи аналогового входа и цифровой системы (например, путем применения оптоэлектронной схемы развязки).

Американская фирма American Industrial Measurements выпускает местные преобразователи для непосредственного подключения к терморезисторам, а также к термометрам сопротивления и другим мостовым датчикам. Местный промежуточный преобразователь выдает модулированный по амплитуде импульсный сигнал с частотой до 1 МГц, передаваемый по двухпроводной линии. На приемной стороне амплитуда импульсов преобразуется в цифровой код.

Располагая индивидуальные промежуточные преобразователи у датчиков, можно полнее реализовать принцип местной обработки данных, столь характерный для распределенных АСУ. В общем случае, однако, применение индивидуальных промежуточных преобразователей обходится дороже, чем общий коммутируемый аналого-цифровой преобразователь. Входные коммутаторы в системах с микро-ЭВМ выполняются либо на герконах (герметизированных реле, называемых также капсульными или язычковыми), либо на полупроводниках - полевых транзисторах или комплементарных МОП - структурах (КМОП). Герконы дают чрезвычайно малую погрешность коммутации, но имеют ограниченное быстродействие (порядка миллисекунд) и надежность. Полевые транзисторы отличаются от КМОП-структур более высоким быстродействием и низкими токами утечки, однако КМОП-структуры лучше совместимы с транзисторно-транзисторной логикой (ТТЛ), имеют меньшее рассеяние мощности, надежнее разрывают входную цепь при отключении питания.

Ко входу коммутатора подводится напряжение постоянного тока, падающее на сопротивлении в токовой цепи или на мосте; это напряжение может принимать значение от 10 мВ до 10 В.

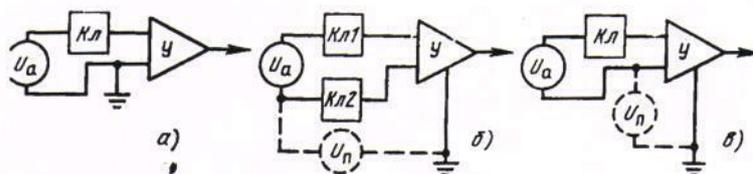
Для защиты транзисторных ключей от перенапряжений во входных коммутаторах устанавливаются диоды, ограничивающие входное напряжение в пределах от +20 до +30 В; ток на входе ограничивается резисторами. Типичное значение входного сопротивления закрытого ключа на КМОП-структурах -  $10^9$  Ом при токе 5 нА, у ключей на полевых транзисторах оно на порядок выше.

Управление входным коммутатором обеспечивает небольшой сдвиг во времени между разрывом цепи подключенного аналогового сигнала и замыканием цепи следующего сигнала, чтобы исключить возможность даже самого кратковременного подключения к входу аналого-цифрового преобразователя двух сигналов.

Входные коммутаторы на ИМС выпускаются в двух модификациях: с устройством декодирования адреса, включенным в общую ИМС, и без такого устройства. Как правило, они допускают три способа подключения входного сигнала (рис.23). Однополюсное подключение применимо, если можно пренебречь отношением продольной помехи - напряжения между источником сигнала и землей усилителя - к полезному сигналу. Двухполюсное симметричное

подключение позволяет полностью использовать способность усилителя ослаблять продольную помеху. Это типичный способ подключения источников сигнала низкого уровня. Каждый источник может иметь произвольный потенциал относительно земли системы в пределах допустимого диапазона продольной помехи, например +30 В.

По сравнению с однополюсным подключением число входных переменных, обслуживаемых коммутатором, сокращается при двухполюсном подключении вдвое.



**Рис.23. Схемы подключения входного аналогового сигнала:**  
**а** — однополюсное подключение; **б** - двухполюсное симметричное;  
**в**-псевдодифференциальное; Кл, Кл1, Кл2 - ключи; У - усилитель;  $U_a$  - напряжение полезного аналогового сигнала;  $U_n$  - напряжение продольной помехи

Третий способ подключения носит название псевдодифференциального. Он предпочтителен, когда все источники аналогового сигнала находятся под одинаковым потенциалом относительно земли системы. При этом способе экономится число ключей входного коммутатора и вместе с тем существенно снижается влияние продольной помехи.

Измерительный усилитель повышает входной сигнал низкого уровня (например, милливольты с выхода термопары) до уровня шкалы на входе аналого-цифрового преобразователя, обычно от 0 до 10 В. Одновременно усилитель ослабляет продольную помеху до пренебрежимо малого уровня. К измерительному усилителю предъявляются многочисленные и не всегда легко выполнимые требования: малый дрейф; высокая степень ослабления продольной помехи (от 90 до 120 дВ и выше); высокое входное сопротивление поперечному (дифференциальному) и продольным (относительно земли) сигналам; высокое допустимое значение напряжения продольной помехи на входе; очень высокая линейность и стабильность запаздывания; короткое время установления выходного сигнала, например достижение 99,99 % установившегося значения за 2 мкс; малый собственный шум. У лучших образцов измерительных усилителей коэффициент ослабления продольной помехи достигает 150 дВ (типичное значение 60 - 80 дБ, иногда 120 дВ), а входное сопротивление продольному сигналу - более  $10^{11}$  Ом. Полоса пропускания усилителя обычно лежит в пределах от 30 Гц до 50 кГц, входное сопротивление 100 МОм.

Высококачественные измерительные усилители строят в виде двухкаскадных операционных усилителей на одном кристалле. Отделение входов по постоянному току от земли осуществляется трансформаторами или оптоэлектронными устройствами. При требовании очень малого дрейфа применяют измерительные

усилители с модуляцией входного сигнала, усилением на переменном токе и последующей демодуляцией. Однако в большинстве применений допустима работа с усилителями без внутренней гальванической развязки между каскадами.

Коэффициент усиления измерительного усилителя должен изменяться ступенями, принимая значения  $2^p$ , где целое  $p$  изменяется от 0 до 10, путем задания управляющего слова программой микро-ЭВМ. Таким образом можно устанавливать свой коэффициент усиления и шкалу для каждой технологической переменной. Управление усилителем можно упростить, введя в схему ввода аналоговых данных оперативную или полупостоянную память, хранящую значения настройки для каждой переменной. К этой памяти регистр адреса переменной обращается - тогда же, когда он обращается к схеме декодирования адреса входного коммутатора.

Фиксаторы являются необходимым элементом системы ввода с аналого-цифровым преобразователем, действующим по принципу поразрядного уравнивания. На входе такого преобразователя сигнал должен оставаться неизменным в течение всего периода преобразования. Требование к постоянству напряжения на фиксаторе тем выше, чем шире полоса входного сигнала и чем больше время одного цифрового преобразования.

Для цифрового преобразования применяются устройства, основанные на всех трех издавна установившихся принципах: параллельного и поразрядного уравнивания и развертывающего время-импульсного преобразования. Чаще всего пользуются преобразователями с поразрядным уравниванием. Преобразователи с параллельным уравниванием применяют лишь в случае особенно высоких требований к быстродействию, а время-импульсные - при низких требованиях к быстродействию или при использовании индивидуальных преобразователей для каждой входной переменной.

## **2. Схемы аналого-цифрового преобразователя**

В составе схемы аналого-цифрового преобразователя должен иметься прецизионный источник эталонного напряжения. В однокристалльных преобразователях стабилизаторы напряжения часто выполняют внутри кристалла. Внешние источники используют обычно тогда, когда цифровой преобразователь служит для получения кода отношения между двумя аналоговыми сигналами.

На выходе аналого-цифрового преобразователя с односторонней входной шкалой, например 0 - 10 В, получают отсчет в двоичном или двоично-десятичном коде, а у преобразователей с двусторонней шкалой (например, от - 5 до +5 В) - в дополнительном двоичном коде со смещением либо в двоично-десятичном коде со знаком. Совместимый с вводом в микропроцессор цифровой преобразователь должен иметь на выходе регистр, допускающий вывод байтами на 8-битную шину. При установке преобразователя на расстоянии от обрабатывающего микропроцессора и применении двухпроводной связи

целесообразна организация последовательно, но одному биту, вывода кода в линию передачи данных.

В целом современные системы аналогового ввода с 8-разрядными аналого-цифровыми преобразователями имеют при 25 °С погрешность 0,2 - 0,4 %, а с 12-разрядными преобразователями - 0,025-0,05 %. С ростом коэффициента усиления погрешность возрастает: например для 12-разрядного преобразователя при коэффициенте усиления 1000 - до 0,2 %.

Частота ввода для большинства промышленных устройств колеблется от 0,03 до 3 Гц, однако при большом числе опрашиваемых переменных она может достигать и 25 кГц. В последнем случае применяют аналого-цифровые преобразователи с параллельным или поразрядным уравниванием, результат преобразования вводится в память по каналу прямого доступа. Для подавления помех при столь большом быстродействии приходится прибегать либо к внешним фильтрам, либо к программной цифровой фильтрации.

Вывод аналоговых управляющих воздействий на технологический процесс на микро-ЭВМ осуществляется с помощью цифро-аналоговых преобразователей, которые преобразуют код в аналоговую величину непосредственно или с промежуточным преобразованием, обычно в последовательность модулированных по ширине прямоугольных импульсов.

В цифро-аналоговом преобразователе прямого действия код управляет током в резисторной цепочке, выполненной тонкой пленкой 81Сг на поверхности интегральной пластины. Эта же ИМС часто содержит эталонный источник напряжения и входной буферный регистр. Последний может загрузиться последовательно (в случае размещения преобразователя на расстоянии от процессора) или по байтам.

Цифро-аналоговые преобразователи с промежуточной широтно-импульсной модуляцией реализуются аппаратурно значительно проще, они не нуждаются в прецизионных резисторных цепочках, однако время преобразования у них сравнительно велико - как правило, более 1 мс.

Платы аналогового вывода выдают от одного до восьми сигналов, причем для каждой переменной применяется свой цифро-аналоговый преобразователь. Выходным сигналом обычно служит напряжение либо ток в унифицированной шкале 4-20 мА (в отдельных случаях 0 - 5 мА). Погрешность 8-разрядных преобразователей составляет около 0,4% для 12-разрядных преобразователей она уменьшается до 0,01 - 0,05 %. Задачей устройств ввода и вывода двоичных сигналов является преобразование логических уровней 0 и 1. В аппаратуре, непосредственно связанной с ТП, этим уровням могут соответствовать напряжение на замкнутом или разомкнутом электронном ключе; замкнутый или разомкнутый контакт реле; сравнительно мощный сигнал 8, 24, 30, 48 или 60 В при токе 2, 10, 100 или 200 мА или нулевой уровень. Устройства выполняются как платы, рассчитанные на ввод или соответственно вывод 16 или 32 двоичных сигналов. В платах предусматриваются регистры для хранения вводимых (выводимых) двоичных наборов.

### 3. Устройства связи оператора с АСУ

Понятие распределенности АСУ относится в первую очередь к размещению аппаратуры. Однако все технические средства, позволяющие оператору воздействовать на процесс, стремятся сосредоточить в одном центральном пункте, здесь же концентрируются и средства, представляющие результаты контроля процесса. В местных узлах устанавливаются лишь минимальные средства контроля за процессом и, возможно, органы настройки контуров регулирования.

В современных АСУ основным средством визуального вывода стал дисплей, гибкость которого повышается вместе с его «интеллектуальностью», растущей по мере оснащения дисплея местными микропроцессорными средствами обработки данных. Традиционные специализированные средства вывода данных - стрелочные шкалы, сигнальные лампы, панели цифровой индикации, графопостроители, печатающие устройства - играют теперь лишь более или менее второстепенную роль.

Преобладающим средством воздействия на процесс в системе управления со стороны оператора стала клавиатура дисплея, лишь в отдельных случаях пользуются клавиатурой печатающих устройств, аналоговый же ввод применяется только на специальных объектах.

Поскольку дисплеи снабжаются клавиатурой, устройство в целом, обеспечивающее общение оператора с АСУ называется дисплейным терминалом.

Современные дисплеи, как правило, строятся на основе электронно-лучевых трубок с цветным воспроизведением.

По принципу образования элементов на экране дисплеи делятся на штриховые и растровые. В штриховых дисплеях элементы изображения прочерчиваются отдельными штрихами, а в растровых - точками, которые проходятся лучом по строкам, как в телевизионных приемниках.

Штриховые дисплеи дают изображение более высокого качества, чем растровые. При применении в штриховом дисплее обычной электронно-лучевой трубки изображения нуждаются в периодическом восстановлении со стороны схемы возбуждения. Этому недостатка лишены запоминающие трубки; однако последние не позволяют получать цветные изображения и передавать быстрые изменения, они менее ярки, имеют более короткий срок службы.

В растровых дисплеях изображение на экране обычно состоит из 256X512 точек, хотя выпускаются дисплеи и с более высокой разрешающей способностью, например 1024X1280 у дисплея американской фирмы Genisco.

По характеру выводимой информации различают алфавитно-цифровые (или символьные) и графические дисплеи. Кроме того, имеются промежуточные по ориентации устройства - так называемые алфавитно-цифровые дисплеи с графическими возможностями. Программы формирования стандартных элементов, подлежащих воспроизведению, -отрезков точек; дуг окружности и других кривых,

символов и т.п., - хранятся в постоянной памяти, и хранящиеся ассортименты элементов отражают - специализацию дисплея.

Самым совершенным (как, впрочем, и самым дорогим) средством общения между оператором и АСУ сегодня является цветной графический терминал. Такой терминал оснащается большой оперативной -памятью (до нескольких сотен миллионов битов) для хранения выводимого изображения - так называемой п а м я т ью о б н о в л е н и я (refresh memory) - и дает возможность получить миллионы цветовых оттенков (например, цветовой терминал американской фирмы Comtal дает 16 млн. оттенков цвета). Изображение хранится в памяти обновления в трех «срезах», каждый из которых соответствует одному цветовому компоненту. Память обновления позволяет дисплею работать автономно от главной ЭВМ. Эта память однажды загружается из главной ЭВМ, а затем к ней обращаются электронные схемы дисплея, обеспечивая непрерывное обновление изображения на экране. Память обновления выполняется как динамическое или статическое ЗУ с произвольным доступом. Успехи в технологии ЗУ делают эту память все более миниатюрной, быстродействующей, дешевой, экономичной по потребляемой мощности. Однако поскольку быстродействие памяти все еще отстает от требуемых скоростей обновления изображения, обычно в память обновления обращаются одновременно за 8 или 16 элементами изображения. Наиболее популярна память обновления на 512X512 8-битовых элементов изображения; имеются, впрочем, и запоминающие устройства на 4096X4096 элементов от 1 до 16 битов в каждом.

К настоящему времени цветными графическими дисплеями оснащены все современные комплексы средств для построения распределенных АСУ. Ранее разработанные и более простые комплексы, предусматривают применение черно-белых алфавитно-цифровых дисплеев с ограниченными графическими возможностями. Для хранения постоянной информации, подлежащей выводу на экран дисплея, пользовались гибкими дисками.

Дисплеями в распределенных АСУ оборудуются все пункты контроля. Они входят в состав станций контроля и управления либо присоединяются к ним как автономные терминалы. Так, в комплексе Centum, выпускаемом японской фирмой «Екогава дэнки», каждой станции оператора, рассчитанной на 32 контролируемые переменные, может быть придано до четырех дисплеев с электронно-лучевой трубкой размером 50,8см. На каждом экране формируется до семи изображений, имитирующих аналоговые пульта управления соответствующими участками процесса. Обращаясь к этим изображениям, оператор получает информацию о состоянии процесса и системы управления, а также воздействует на процесс, характеристики управления и способы контроля.

Информация о процессе, выводимая на экран дисплея, различается шириной охвата и подробностью описания объекта; кадры представления упорядочиваются некоторой иерархической системой, содержащей до четырех уровней.

Хотя дисплеи являются универсальным средством общения оператора с АСУ, в комплексах для распределенных АСУ еще используются и более традиционные

средства визуального вывода и регистрации. Так, в комплексе Micro-Z Французской фирмы Controle Bailey центральная станция, рассчитанная на 240 контуров регулирования, в дополнение к дисплеям и лампам сигнализации допускает подключение печатающего устройства, многоточечных самописцев, регистраторов тенденции и устройств фотокопирования экрана дисплея. Иногда предусматривается сохранение традиционной мнемосхемы, и тогда в комплекс входит специальное устройство с микропроцессором, управляющее лампами мнемосхемы (комплекс Damatic финской фирмы Valmet).

Среди новых специализированных средств визуального вывода в последние годы получили распространение линейные аналоговые указатели значений контролируемых величин.

Электронные линейные указатели сочетают лучшие свойства цифровых индикаторов - якорь и четкость изображения - и аналоговых индикаторов - легкость восприятия. Этим и объясняется их широкое распространение. По сравнению со стрелочными указателями электронные линейные указатели не подвержены влияниям вибрации, ударов, практически безынерционны, могут иметь гораздо меньшие размеры. По сравнению с цифровыми индикаторами линейные указатели предпочтительнее, когда достаточна точность 1 %, они в отличие от цифровых индикаторов не страдают от колебаний отсчета. Пример изображения на экране линейного указателя показан на рис. 24. Переход переменной за заданную границу или нарушение режима при применении линейных указателей удобно отображать переходом через отметку или изменением цвета. Гибкость в генерировании выводимого изображения в современных линейных указателях легко достигается применением микропроцессоров. Точность отсчета в случае необходимости легко может быть повышена добавлением к линейному отсчету также цифрового указания. Линейные указатели выполняются на плазменных или электронно-лучевых приборах, светодиодах или жидких кристаллах.

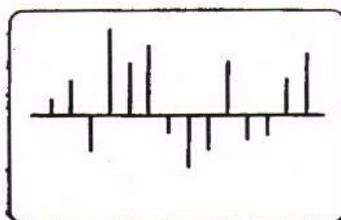


Рис.24. Изображение на экране линейного указателя

Для линейной индикации выпускают специальные электроннолучевые трубки. Цвет индикации - зеленый

Клавиатура - одно из самых старых устройств ввода в ЭВМ, совершенствование ее непрерывно продолжается в направлении повышения аппаратной и функциональной надежности.

Весьма экономно применение виртуальной клавиатуры. При этом функция каждой клавиши программируется. Чтобы оператор знал, каково назначение каждой клавиши в данный момент, расшифровка функций дается в изображении клавиатуры на некотором участке экрана дисплея.

Иногда, как, например, в комплексе Micro-Z, предусматривается возможность пользования небольшой переносной панелью с клавиатурой и цифровым индикатором. Панель присоединяется к станции управления штекером.

Ввод и вывод речевых сигналов при взаимодействии оператора с системой является реальной перспективой развития ближайшего будущего.

#### Контрольные вопросы

1. Какие устройства относятся к устройствам связи АСУ процессом? [Л.7, 53-54].
2. Расскажите о передаче аналоговых сигналов от датчиков. [Л.7, 54-55].
3. Что является главной проблемой при вводе аналоговых величин? [Л.7, 55-56].
4. Объясните о двух модификациях, выпускающих входные коммутаторы на ИМС. [Л.7, 57-58].
5. Сколько способов имеются подключения входного сигнала к входным коммутаторам на ИМС? [Л.7, 58-59].
6. Что должно иметься в составе схемы аналого-цифрового - преобразователя? [Л.7, 59-60].
7. Расскажите о цифро-аналоговых преобразователях. [Л.7, 59-60].
8. Что является основным средством визуального вывода в современных АСУ [Л.7, 60-61].
9. Перечислите лучшие свойства цифровых и аналоговых индикаторов. [Л.7, 62-64].
10. Какие недостатки имеют светодиоды? [Л.7, 64-66].

## Лекция №5

### Функции и программы распределенных АСУ. Физический канал. Информационный канал

#### План

1. Функции и программы распределенных АСУ
2. Физический канал.
3. Информационный канал.

#### 1. Функции и программы распределенных АСУ

Выше была представлена архитектура распределенной АСУ причем ее основу составляли функции и программы системы. Функции, реализующие их алгоритмы и программы, расслаиваются по уровням, каждый из которых имеет свое назначение. Самый верхний уровень - прикладной - содержит функции контроля и управления процессом которых и создается АСУ.

Функции и программы, находящиеся в состоянии исполнения, принято называть процессами. Прикладные процессы для своего осуществления должны обмениваться данными, и этой последней цели и служат остальные, нижележащие уровни функционально-программной архитектуры. Функции этих уровней будем называть служебными.

Нижние четыре архитектурных уровня предназначены для организации обмена между процессами по сети, а пятый и шестой (снизу) уровни организуют обмен данными между прикладными процессами независимо от места их исполнения, будь то разные или один и тот же процессор.

Принципы архитектуры предполагают независимость решений на каждом ее уровне (при условии соблюдения интерфейсов), а для этого необходимо разделять по уровням также все программные и аппаратные средства. Однако нынешние системы еще далеки от строгого следования предписаниям слоистой архитектуры. В настоящее время требования архитектуры лучше выполняются при реализации нижних архитектурных уровней. В последних разработках для распределенных АСУ уже стремятся четко разделить аппаратуру по сетевым архитектурным уровням, однако вряд ли можно ожидать в обозримом будущем расслоения аппаратуры по трем надсетевым уровням: по-видимому, функции этих уровней по-прежнему будут реализовываться в основном на одном процессоре с общими ресурсами по вычислениям и памяти. Подобным же образом устройства связи с оператором и связи с технологическим объектом выполняют функции всех трех надсетевых уровней, в то время как функции сетевых уровней в них выполняют отдельные встроенные блоки. В практику построения распределенных АСУ уже вошли ИМС, реализующие аппаратным или микропрограммным путем протоколы физического, канального и даже магистрального уровня.

Изложение функций и программ распределенной АСУ здесь ведется «снизу вверх»: оно начинается с нижнего, физического уровня и заканчивается прикладным.

Согласно архитектуре распределенных АСУ процессы на каждом уровне пользуются услугами средств нижележащего уровня. Ресурсы каждого уровня ограничены, в связи с чем необходимо распределять ресурсы между запрашивающими их процессами. Для распоряжения ресурсами применяются программы управления (которые на нижних уровнях могут быть реализованы аппаратурно), обычно входящие в состав операционной системы. Операционные системы для вычислительных сетей чаще всего сами оказываются распределенными между станциями сети. Операционные системы распределенных АСУ характеризуются ниже.

## 2. Физический канал

На уровне управления физическим каналом линейный узел передающей станции определяет все параметры физического сигнала, который служит для передачи информации по каналу, а линейный узел приемной станции обрабатывает поступающие сигналы. Элементами сигнала, соответствующими двоичному разряду кода, по большей части являются электрические импульсы, и лишь по оптическим каналам пропускаются импульсы света.

Параллельную передачу данных между узлами распределенной АСУ применяют редко и лишь на расстояниях до нескольких десятков метров.

На уровне физического канала определяется закон измерения во времени сигнала в линии; характеризуя этот закон, часто пользуются термином «линейный код», что, по-видимому, методически ошибочно.

При двоичной кодово-импульсной модуляции нулевому и единичному значениям бита соответствуют два различных элемента сигнала. Различают два вида двоичной модуляции; при одном из них сигнал непосредственно переходит от уровня, соответствующего данному биту, к уровню, соответствующему следующему биту, при другом сигнал в промежутке между двумя элементами, соответствующими соседним битам, принимает определенный уровень, который можно считать соответствующим паузе между битами.

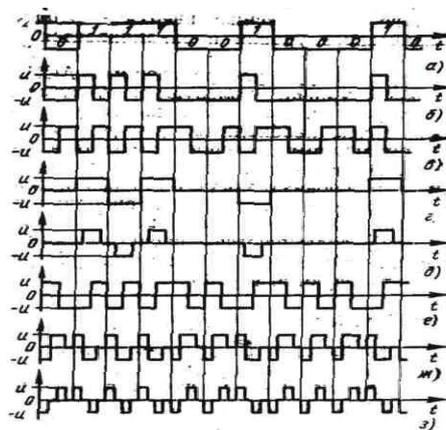


Рис.25. Сигналы в линии передачи данных:

- а - беспauseный знакопеременный; б - знакопеременный с pauseй; бимпульсный;
- г - квазитроичный; д - квазитроичный с импульсами, занимающими половину такта; е – беспauseный
- двухфазный (манчестерский); ж - двухфазный (манчестерский) с pauseй;
- з – двухфазный (манчестерский) с двумя pauseми

В результате модуляции первого вида получают беспазузный сигнал, который нередко называют (опять-таки методически неправильно) кодом без возвращения к нулю (по-английски NRZ - non - return - to - zero). Беспазузный сигнал иллюстрируется на рис. 3.1,а. Беспазузная двоичная модуляция требует меньшей полосы частот в канале, чем модуляция с паузой (рис.25,б), но при ней несколько затрудняется синхронизация приема каждого бита. При передаче по линии видеосигнала существенно отсутствует составляющая в линии, что облегчает трансформаторную связь передающей и приемной аппаратуры с линией. Отсутствие постоянной составляющей в линии может быть гарантировано только в том случае, когда отсутствует постоянная составляющая в сигнале, соответствующем каждому биту. Таковую возможность предоставляют некоторые специальные виды двоичной модуляции, о которых речь пойдет ниже. Одновременно при этом сигнал, как правило, приобретает свойства, позволяющие проверить его правильность при приеме.

В бывшем СССР наряду с беспазузным сигналом в линии передачи данных стандартизованы также биимпульсный (его называют также дифференциальным манчестерским кодом) и квазитроичные сигналы. Каждому биту в линии связи соответствует элемент сигнала, который на каждом такте, отведенном для передачи бита, изменяет уровень в середине такта с и на - и или наоборот. При этом при передаче единичного бита уровень меняется также в начале такта, а при передаче нулевого бита уровень сохраняется (рис.25,в). При приеме биимпульсного сигнала считаются ошибочными элементы сигнала, в которых отсутствует переход через нуль; этот переход может использоваться для синхронизации по каждому элементу сигнала при приеме.

Квазитроичный сигнал принимает три уровня значений: и, 0 и - и нулевой бит всегда передается нулевым уровнем, а единичный - уровнем и или - и; при этом знак импульса, передающего единичный бит, всегда изменяется на противоположный по сравнению с предшествовавшим импульсом (рис.25,г). В квазитроичном сигнале допускается также применение импульсов, соответствующих единице, шириной в половину такта (рис.25,д). Хотя постоянная составляющая каждого импульса в квазитроичном сигнале не равна нулю, он практически гарантирует равенство нулю постоянной составляющей в линии.

Знакопеременный беспазузный сигнал на рис.25,а, называемый сигналом низкого уровня, предлагается для синхронной или асинхронной связи со скоростью передачи до 20 кбит/с, биимпульсный - для синхронной связи по 4-проводным линиям на скоростях от 1,2 до 48 кбит/с или выше, квазитроичный - для передачи со скоростью свыше 48 кбит/с. Уровень и на передающей стороне не должен превышать 1 В для скоростей до 72 кбит/с, 2 В для скоростей 72 - 144 кбит/с и 3 В для скоростей свыше 192 кбит/с; при передаче по многопарному кабелю, в разных парах которого передача может идти одновременно, ограничение напряжения составляет 0,4В. На входе в приемный узел уровень сигнала не должен быть меньше 0,02 В для сигнала низкого уровня и 0,1 В для биимпульсного и квазитроичного сигналов; передача по многопарному кабелю повышает допустимый уровень на входе приемного узла до 0,05В.

В зарубежной аппаратуре передачи данных видеосигналами широко применяют двоичный сигнал, называемый *д в у х ф а з н ы м* или *м а н ч е с т е р с к и м* кодом. Сигнал бита принимает значения обоих уровней *и* и *и*, и значение бита передается направлением перехода элемента сигнала с одного уровня на другой. Например, нулевой бит передается переходом от *и* к *-и*, а единичный - переходом от *-и* к *и* (рис. 25,е). Обычно переход совершается в середине такта, отведенного для передачи бита. Манчестерский сигнал гарантирует нулевую постоянную составляющую и имеет те же достоинства в отношении синхронизации и контроля правильности приема элементов сигнала, что и биимпульсный сигнал.

Наряду с беспauseной формой манчестерского сигнала применяют манчестерский сигнал с pauseй. Такт элемента сигнала делится на три равных отрезка (рис.25,ж). Нулевому значению бита отвечает значение *-и* сигнала на первом отрезке, *и* на втором и нуль на третьем, а единичному значению *и*, *-и* и *0* соответственно.

В магистрали Niway комплекса ТВС2000 каждый бит передается за 4 мкс двумя импульсами - положительным и отрицательным, чередующимися с pauseями (Рис.25,з). Если в каком-либо бите принятого слова нет импульсов обеих полярностей, все слово отбрасывается как ошибочное.

При широкополосной передаче по коаксиальному кабелю в современных локальных сетях пользуются фазовой манипуляцией с плавным изменением фазы (в течение 100 нс) между значениями 3,75 и 6,25 МГц, причем рабочая скорость передачи составляет 1 Мбит/с. Такая манипуляция по биимпульсному закону принята в проекте Proway C.

Физические каналы передачи данных в зависимости от того, синхронизируются ли в них элементы сигнала, представляющие биты, делятся на асинхронные и синхронные. В асинхронных каналах каждый элемент сигнала передается в произвольное время по мере его генерации передающей аппаратурой, тогда как в синхронных каналах момент передачи элемента сигнала задается непрерывно работающим тактовым генератором.

В стандарте Proway под линейным протоколом понимаются правила преобразования кадра из формата, принятого для представления внутри станции, в сигналы, передаваемые по каналу связи, и обратного преобразования сигналов в линии в кадр станции. Эти функции возлагаются на специально выделяемый блок линейного перехода.

В подготовленной первой части стандарта Proway устанавливаются лишь функции линейного протокола (или, что то же самое, блока линейного перехода) без указания на способ их выполнения. Это следующие функции:

- преобразование уровней *и* (или) форматов сигнала;
- электрическое разделение цепей станции и линии связи;
- контроль качества сигнала;
- генерирование и детектирование синхронизирующих сигналов;
- синхронизация кадра (если она не предусматривается канальным протоколом);
- обнаружение состояния линии связи (свободна, занята, неисправна);
- синхронизация инициатора, исполнителя и приемника.

Функции физического уровня для локальной сети с состязанием имеют определенные особенности, которые нашли свое отражение в стандарте ЕСМА-81. Магистраль по стандартам ЕСМА строится на коаксиальном кабеле. Передача ведется манчестерским кодом с рабочей скоростью передачи  $10 \text{ Мбит/с} \pm 0,01 \%$ .

Физический уровень выходит на канальный уровень сигналами, указывающими на поступление из линии очередного элемента сигнала (соответствующего биту), из ошибку в линейном сигнале, на наличие несущей в линии, а также запросом очередного бита, подлежащего передаче. Ошибка в линейном сигнале обнаруживается по повышенному уровню сигнала в линии, например постоянной составляющей. С канального уровня физический уровень получает сигнал каждого бита кадра, подлежащего передаче, сигнал об окончании передаваемого кадра и запрос на передачу данных со станции. Стандарт ЕСМА-81 ограничивает запаздывание в выполнении каждой из функций, выражая допустимое значение числом интервалов между следующими друг за другом элементами сигнала (битами).

На уровне управления физическим каналом нет задачи распределения ресурсов для поддержания связи: эта задача решается лишь начиная с уровня информационного канала. Поэтому здесь протоколы не классифицируются по признакам, введенным в лекции 1.

### **3. Информационный канал**

К уровню информационного канала (иногда говорят просто об уровне канала) относятся задачи установления связи между двумя станциями, распознавания кадров, адресованных принимающей станции, синхронизации кадров, обнаружения и исправления ошибок в кадрах канала, наконец, преобразования параллельных кодов в последовательные и обратно, если применяется последовательная передача данных.

В то время как последние из перечисленных задач не связаны с распределением сетевых ресурсов и являются более или менее сходными во всех сетях, установление связи между двумя станциями может предполагать доступ к одному или нескольким физическим каналам. Принцип решения этой задачи имеет фундаментальное значение при построении сети передачи данных. Вместе с тем для установления связи между двумя станциями требуется, чтобы они были готовы к связи, т. е. свободны от других работ, так как установление связи нельзя полностью свести только к доступу к физическому каналу. Тем не менее, сначала выделим процедуры протоколов, выполняемые после получения доступа к физическому каналу, а затем уже рассмотрим вопросы использования физического канала. Заметим, что в архитектуре соединения открытых систем МОС обе рассматриваемые проблемы - установления связи и доступа к каналу - относятся к одному (канальному) уровню, а в архитектуре Proway первая проблема отнесена к канальному, а вторая - к лежащему над ней магистральному уровню.

Задача доступа к физическому каналу отсутствует в сетях радиальной структуры, однако поскольку радиальные каналы являются скорее исключением, нежели правилом в распределенных АСУ перейдем к рассмотрению

канальных протоколов магистралей. Параллельная передача данных между станциями применяется в распределенных АСУ в настоящее время все реже, так как ее вытесняет последовательная передача, быстродействие которой существенно повысилось благодаря использованию коаксиального кабеля и волоконной оптики. Параллельная передача сохраняет свое значение лишь как средство связи между устройствами операторского пункта на небольших расстояниях. Канальные протоколы оперируют с логическими элементами, передаваемыми от станции к станции. Таким логическим элементом может быть двоичный разряд (бит) или группа двоичных разрядов (байт). Последний чаще всего содержит 8 битов (тогда он называется *о к т е т о м*); как правило, при употреблении термина «байт» без оговорки имеют, в виду именно октета. Байт, передаваемый по каналу, состоит из следующих четырех частей (рис. 26):

начального бита;

от 6 до 8 битов данных в зависимости от применяемого кода (чаще всего применяют 8-битовый код А8СП);

бита контроля по четности (не обязательного); одного-двух концевых битов.

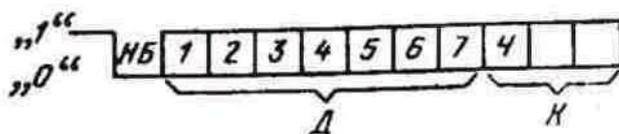


Рис. 26. Формат слова при асинхронной передаче данных: Д - данные; К - концевик; НБ - начальный бит; Ч - бит контроля по четности

Обычно для скорости передачи до 110 брд (10 знаков/с) применяют два концевых бита, а для более высокой скорости - один. Передача байтов может вестись асинхронно, с паузой, произвольной, длительности между байтами, либо синхронно, с фиксированной тактовой частотой. При асинхронной связи начало байта задается начальным битом, который переводит сигнал в линии из состояния 1 в состояние 0, а конец - одним или двумя концевыми битами. После концевого бита линия остается в состоянии 1 до поступления начального бита следующего байта. При синхронной связи обязательна синхронизация на физическом уровне, а счет импульсов, тактирующих биты, позволяет синхронизировать и байты. Если тактовая частота известна приемнику, то принятый сигнал может быть идентифицирован (рис.27). Хотя приемное устройство ожидает синхронного последовательного поступления данных, начало приема должно быть идентифицировано. Этой цели служат один или два синхронизирующих байта. Данные в посылке байта содержат от 5 до 8 битов, которые могут сопровождаться битом контроля по четности. Во время ожидания приемное устройство проверяет каждый поступающий байт, выясняя, не является ли он стандартным синхронизирующим байтом. С момента начала синхронной последовательной передачи данные поступают в линию с заданной скоростью. Если данные, подлежащие передаче, не обновляются с частотой передачи, то передатчик вставляет в передаваемую последовательность синхронизирующие

байты до тех пор, пока не появится байт, действительно подлежащий передаче (рис.28).

Асинхронное установление связи начинается с запроса, посылаемого либо с передающей, либо с приемной станции. Кадры, содержащие информацию (данные), передаются сразу же после получения запроса передающей станцией или после ответа о готовности станции к приему.



Рис. 27. Интерпретация битов при синхронной передаче данных



Рис. 28. Формат канальных кадров синхронной передаче: Д - данные; СИН - синхронизирующая посылка

Контроль правильности переданного кадра на канальном уровне производится почти исключительно с помощью циклического избыточного кода. Значения циклического  $(n, k)$  кода образуются сочетанием строк так называемой производящей матрицы, последние же получаются из некоторой производящей комбинации, содержащей  $n$  двоичных разрядов, путем  $l$  циклических перестановок. Циклический код принято описывать многочленом степени  $n^{-1}$ .

$$G(x) = \sum_{i=0}^{n-1} \delta_i x^i$$

где  $\delta_i$  равен 0 или 1 в зависимости от значения  $i$ -го разряда кода.

Последний, 20-й бит каждого кадра служит для контроля кадра на канальном уровне по четности.

В магистральной Hiway комплекса TDC-2000 применяются два формата кадра фиксированной длины 31 бит: для команд и для кадров с данными. Команда указывает операцию, подлежащую выполнению, станцию и адрес в памяти станции. Кадр с данными имеет поля: кода операции (биты 2 - 4), адреса источника (биты 5 - 10), данных (биты 11 - 26) и проверочных символов (биты 27 - 31). Контроль достоверности ведется с помощью 5-разрядного кода Бозе-Чаудхури-Хокенгема (БЧХ), обнаруживающего кратные ошибки в кадре.

Среди бит-ориентированных канальных протоколов наибольшее распространение как в вычислительных сетях, так и в сетях АСУ получил протокол HDLC (High-level Data Control - управление каналом передачи данных на верхнем уровне), принятый как стандартный МОС, МЭК в составе стандарта Proway и МККТТ.

На подуровне логического управления каналом выдается запрос на передачу канального кадра. Здесь адреса получателя и источника канального уровня соединяются в соответствующий формат и передаются на кадровый подуровень. После

выдачи подуровнем логического управления запроса на передачу кадровый подуровень строит кадр из спускаемых ему данных, добавляя в случае необходимости значащие байты («балласт»), чтобы кадр не оказался меньше минимально установленной длины 64 байта.

При приеме кадровый подуровень проверяет, относится ли адрес поступающего кадра к данной станции, и если это так, то он пропускает содержательную часть кадра на подуровень логического управления каналом, где правильность приема контролируется по циклическому избыточному коду с генераторным полиномом

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{11} + x^{10} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1,$$

а также по тому, целое ли число байтов содержится в кадре. Если принятое сообщение признается верным, генерируется соответствующий код состояния.

Прием, при конфликте идет независимо от того, обнаружена ли ошибка в сигнале. Однако принятые в этом случае кадры всегда короче самого короткого правильного кадра в 64 байта и отбрасываются на подуровне управления линией связи.

Формат кадра по стандарту ЕСМА-82 показан на Рис.29, где слева от соответствующего поля указана его длина в байтах.

Преамбула, представляющая собой серию битов, вводится канальным уровнем в начало каждого кадра, передаваемого в линию, с тем чтобы цепи станции успели войти в стационарный режим. В качестве преамбулы пользуются байтом 10101010 (первой передается единица, записанная слева), который повторяется 8 раз. Флаг начала информационного поля - байт 10101011.

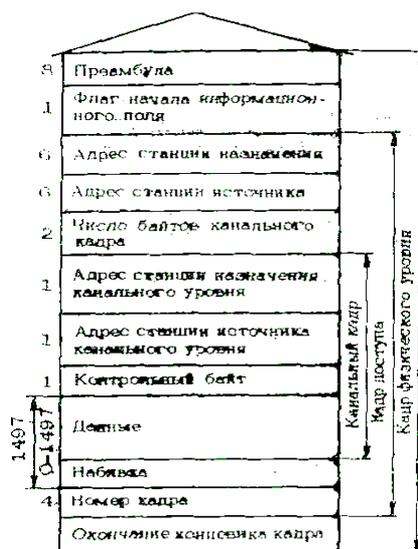


Рис.29. Формат кадра по каналному протоколу ЕСМА-82

Считается, что на окончание концевика кадра нужно отвести от 3,5 до 5 периодов частоты следования элементов сигнала в линии.

В [7] отмечается, что в магистралях можно упростить обмен между станциями, если в заголовке кадра будут содержаться два адреса: передающей и приемной станций.

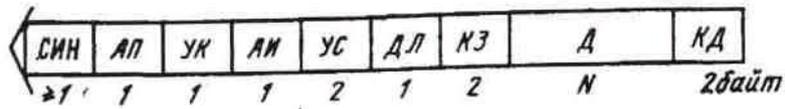


Рис.30. Формат кадра по каналному протоколу согласно [7]:

АИ - адрес источника; АП - адрес получателя; Д - данные; ДЛ - длина кадра; КД - поле контроля данных; КЗ — поле контроля заголовка; СИИ — синхронизирующий байт; УК — поле управлений каналного уровня; УС - поле управления сетевого уровня

Учитывая это и другие факторы, связанные с требованиями АСУ в [7] предложен каналный протокол, рассчитанный на промышленные АСУ, где велика вероятность появления ошибок при передаче по линии связи. На каналном уровне предлагается подтверждать правильный прием кадра.

Формат кадра предлагаемого протокола показан на Рис.30.

На каналном уровне формируется флаг, отмечающий начало кадра, собирается заголовок и генерируется циклический избыточный код, позволяющий обнаруживать ошибки. При обнаружении ошибки в заголовке кадр игнорируется, если же ошибки нет, то принятый заголовок сразу же подвергается обработке, а тем временем собирается и проверяется часть кадра, содержащая данные. На каналном уровне применяются два типа кадров - информационные и командные. Для обработки кадров в соответствии с процедурой, иллюстрируемой упрощенной схемой рис.31, служит поле управления кадра, показанное на рис.32.

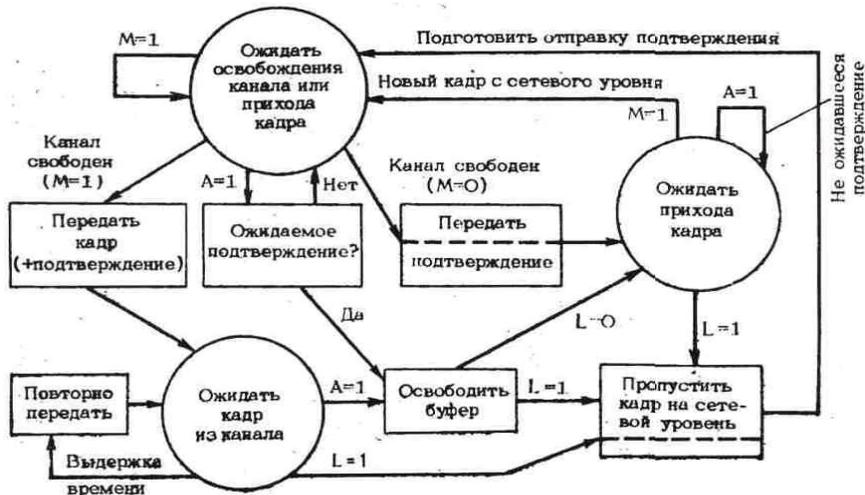


Рис. 31. Диаграмма каналного протокола:

A=1 - кадр содержит подтверждение; L=1 - принят кадр из канала; M=1 - новый кадр для передачи с сетевого уровня

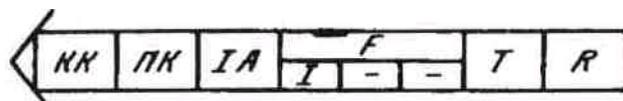


Рис. 32. Поле управления кадра, изображенного на рис. 30:

КК - признак командного кадра; ПК - признак подтверждения команды; Р - команда; I - признак информационного кадра; К - номер ответного кадра; Т - номер передаваемого кадра

Информационный кадр ( $I=1$ ) содержит информацию для более высоких уровней сетевой архитектуры вплоть до прикладного уровня. В нем содержится порядковый номер кадра  $T$ , который изменяет свое значение с 1 на 0 или обратно с каждым новым кадром: если число неподтвержденных кадров не может превышать одного, то принципиально достаточно для нумерации кадров одного бита (четный или нечетный кадр). Номер  $T$  сравнивается на приемной станции с ожидаемым порядковым номером кадра. Когда прием информационного кадра подтверждается, принятый порядковый номер кадра  $T$  передается в поле порядкового номера ответа  $K$ , а бит  $1A$ , подтверждающий прием информации, в исходящем заголовке обращается в 1.

По возможности это подтверждение сочетается с передачей новой информации, ожидающей своей очереди. Когда приходит кадр с подтверждением правильного примера, т.е. с  $1A=1$  и  $R=LT$ , где  $LT$  - порядковый номер последнего переданного кадра, буферная память, занятая подтвержденным кадром, освобождается.

Командные кадры ( $C=1$ ) служат для приведения в определенное состояние информационного канала, а возможно, и всей системы связи на станции. Эти кадры всегда должны приниматься, чтобы не нарушался контроль номеров информационных кадров, но они построены таким образом, чтобы их повторение не сказывалось на работе канала. Команда, помещенная в поле  $P$ , выполняется только на конечной станции назначения. Принятые команды подтверждаются установкой значения  $CA=1$ , и в каждый момент времени может выдаваться только одна команда. Очевидно, что биты  $C$  и  $1$  не могут одновременно равняться единице, но оба могут быть равны нулю. Протокол предусматривает четыре команды, хотя для целей диагностики и проверки состояния этот набор может быть расширен.

Канальный протокол магистрали PDV, разработанной специально для АСУ внес в протокол HDLC полезные для АСУ усовершенствования.

Кадр в магистрали PDV строится из 8-битовых полей четырех типов:  $A$  - адресного,  $P$  - функционального,  $8$  - контрольного и  $D$  - данных. В начале и в конце каждого кадра передаются синхронизирующие знаки ( $SYN$ ).

Особенность формата кадра PDV заключается в том, что в заголовок наряду с адресным и функциональным полями введено также контрольное поле. Это позволяет проверять адрес и команду до начала передачи данных, что особенно полезно при передаче длинных сообщений, и допускает аппаратную реализацию контроля.

Форматы кадров PDV показаны на рис. 33.

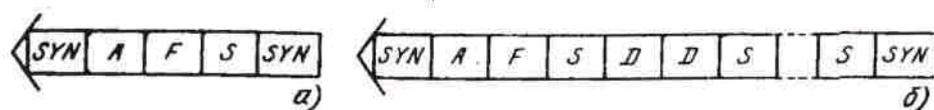


Рис. 33. Форматы канальных кадров по проекту PDV:

а - командный кадр; б - информационный кадр; А - адрес; D - данные; F - функциональное поле; 8 - проверочный байт; S - проверочный байт; SYN - синхронизирующий байт

Каждый шаг обмена по магистрали PDV состоит из команды и ответа; ответа не требуют только команды по групповым адресам.

Станция, устанавливающая связь по магистрали PDV посылает другой станции кадр с командой о передаче или приеме данных. Заголовок кадра содержит адрес, команду и контрольный байт. В адресное поле кадра помещается главный адрес вызываемой станции: это тот же адрес, который ставится затем в адресном поле ответного кадра. Если 8 битов не хватает для прямой адресации, возможно расширение адреса введением дополнительного адресного поля AI. Дополнительный адрес в ответный кадр не включается.

В стандарте Proway устанавливается, что канальный протокол этой магистрали для промышленных АСУ должен решать следующие задачи: преобразование параллельных кадров в последовательные (при передаче) и последовательных в параллельные (при приеме); введение и исключение знаков синхронизации кадра; обнаружение ошибок в синхронизации кадра; обнаружение ошибок в формате кадра; распознавание кадров, адресованных данной станции; установление необходимой выдержки времени перед началом передачи со станции; генерирование и контроль кодов, обнаруживающих и исправляющих ошибки, в целях обеспечения высокой достоверности данных; обеспечение эффективной работы с кадрами магистрали, длина которых колеблется в широких пределах (например, от 2 до 1024 байт); переключение в случае необходимости на резервный физический канал связи.

К удовлетворению всех этих функциональных требований в настоящее время близки проект PDV, а также новые проекты Proway.

### Контрольные вопросы

1. Что принято называть функции и программы, находящиеся в состоянии исполнения? [Л.7, 66-67].
2. Расскажите о принципах архитектуры. [Л.7, 66-67].
3. Почему редко применяют параллельную передачу данных между узлами распределенной АСУ [Л.7, 67-68].
4. Сколько видов различают в двоичной модуляции? [Л.7, 68-69].
5. Что такое манчестерский код? [Л.7, 68-69].
6. Объясните асинхронные и синхронные передачи данных в физических каналах. [Л.7, 69-72].
7. Имеется ли задача распределения ресурсов для поддержания связи на уровне управления физическим каналом? [Л.7, 72-74].
8. Опишите задачи, решаемые на уровне информационного канала. [Л.7, 74-75].
9. Проиллюстрируйте интерпретацию битов при синхронной передаче данных. [Л.7, 75-78].
10. Какие задачи должен решать канальный протокол в стандарте Proway? [Л.7, 79-84].

## Лекция №6

### Управление магистралью. Управление сетью

#### План

1. Управление магистралью.
2. Управление сетью.
3. Протоколы в цепных структурах.

#### 1. Управление магистралью

Магистраль представляет собой единый физический канал, который поочередно используется множеством станций. Задачей протокола, управляющего магистралью, является распределение ее между разделяющими ее информационными каналами; поэтому часто говорят в данном случае об управлении доступом к магистрали.

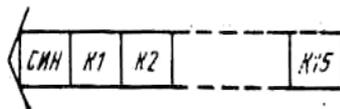
Заметим здесь, что одни и те же протоколы управления доступом к каналу могут применяться к различным физическим структурам - и к магистрали, и к кольцу, если только в последнем не используется промежуточное хранение передаваемых пакетов, и хотя далее будем говорить о доступе к магистрали, то же будет относиться и к доступу к кольцу. Однако кольца в отличие от магистралей обязательно требуют синхронизации передаваемых одновременно от разных станций пакетов; поэтому в кольце всегда выделяется отрезок времени для передачи одного пакета, называемый временным окном или просто окном.

Рассмотрение доступа к физическому каналу начнем с классификации магистральных протоколов по принципам, в соответствии с которыми они распределяют этот подчиненный им сетевой ресурс.

Протоколы со статическим, т. е. принятым раз и навсегда при создании системы, разделением физического канала предусматривают обычно частотное или временное разделение. Частотное разделение практически не применяется в распределенных АСУ. В промышленных АСУ и других локальных сетях применяют временное разделение физического канала. Статическое временное разделение заключается в том, что по одному физическому каналу организуется  $n$  информационных каналов, причем каждый информационный канал устанавливается периодически, с периодом  $T$ , на отрезок времени  $T$ , соответствующий временному окну. Чтобы не связывать таким образом каналами между собой все станции сети, т.е. не устанавливать слишком много каналов, связь  $n$  станций осуществляется через некоторую центральную станцию, внутри которой происходит переадресация сообщений и передача их опять по общему физическому каналу на станцию назначения..

Протокол передачи по магистрали со статическим разделением времени предусматривает в силу заданного размера временного окна фиксированный формат информационного кадра; для системы с  $n=15$  этот формат иллюстрируется на рис. 34. Кадр центральной станции состоит из 16 частичных

кадров, первый из них - синхронизирующий знак СИН, остальные 15 - кадры, адресуемые каждой из 15 станций или подставляемые станциями для приема центральной станцией. Содержание каждого информационного кадра К1 - К15 уже не связано с магистральным протоколом и было рассмотрено в лекции 5.



**Рис.34. Кадр центральной станции в магистрали со статическим разделением канала: К1-К15 - кадры, адресуемые подчиненным станциям; СИН - синхронизирующая посылка**

Иногда применяют более сложные варианты протокола со статическим временным разделением. С учетом различной загрузки магистрали отдельными станциями либо разным станциям отводят окна разной длительности, либо отдельные станции получают не одно, а несколько окон в каждом цикле опроса.

Протоколы со статическим разделением физического канала весьма просты, но неэффективно используют канал, а потому применимы только при относительно низкой загрузке канала, либо в связи с использованием сверхбыстродействующего канала (на коаксиальном кабеле или волоконной оптике), либо на объектах управления, порождающих редкий поток сообщений.

Лучше используется магистраль в протоколах с опросом, которые отдают магистраль только станциям, выставившим запрос на передачу данных. Здесь фиксирован лишь порядок возможного доступа станций к магистрали. При централизованном управлении доступом опрос производится одной станцией - контролером магистрали. Станции опрашиваются циклически; как и в протоколах со статическим разделением, возможен более частый (несколько раз за цикл) опрос отдельных станций.

При децентрализованном управлении все станции равноправны и право доступа передается от одной станции к другой в заранее заданном цикле. Такой протокол называют *передачей маркера* (token passing). Получив в свое распоряжение магистраль, станция передает сообщение целиком или же только часть его, если протокол предусматривает передачу доступа к магистрали по истечении определенного отрезка времени.

Протокол с передачей маркера, предусматривающий возможность исключения станций из сети и введения новых станций в сеть без нарушения ее работы, подробно разработан в проекте Proway C, предложенном в начале 1984 г.

Более эффективны, чем протоколы с опросом, но зато и более сложны протоколы с произвольным поступлением запросов от станций по их инициативе.

Стандарт Proway устанавливает, что к задачам магистрального протокола относятся контроль работы магистрали и управление ею, включая исправление ошибок и управление доступом к каналу. Предусматривается шесть функций, связанных с работой на магистрали. Они располагаются по рангам старшинства. Самая младшая функция - прием. Она заключается в приеме всех

правильных кадров, представляющих интерес для станции, которой они предназначены. Станция, способная осуществлять прием, называется **п р и е м н и к о м (listener)**

Следующая по рангу функция - исполнение, или ответ. Это - прием кадров в сформулированном выше смысле, дополненный ответом на них и немедленным извещением об этом станции, пославшей кадры. Станция, наделенная способностью давать ответ, носит название исполнителя (**responder**).

В стандарте Proway предполагается, что станция старшего ранга, как правило, способна выполнять функции всех младших рангов, и это будет иметься в виду, если не приводится оговорка, утверждающее обратное. **И н и ц и а ц и я** заключается в решении следующих задач: ответа на опрос активного супервизора (см. ниже) запросом доступа к магистрали; передачи кадров станциям, наделенным функцией приема; выбора исполнителя для одного шага обмена данными путем передачи кадров с адресом; обнаружения нарушения приема кадرا исполнителем и инициирование процедур исправления, которые по возможности должны выполняться автоматически и без чрезмерной задержки других шагов обмена.

Станция, выполняющая инициацию и, как правило, функции всех низших рангов, называется **и н и ц и а т о р о м (initiator)**. Если ранг станции не закреплен и может меняться в процессе функционирования магистрали, то следует говорить о статусе станции.

Станция, имеющая право на получение некоторого статуса, но еще не получившая его, а лишь выставившая запрос на его получение, квалифицируется как **п р е т е н д е н т** на данный статус, например претендует на статус инициатора (**candidate initiator**). Станция же, получившая статус, на который она претендовала, будет называться **а к т и в н о й** в данном статусе, например активным инициатором.

Следующая по рангу функция - заказ - относится к передаче по магистрали запроса, выставленного претендентом на статус инициатора с целью получить доступ к магистрали ввиду неотложной необходимости срочно передать сигнал о некотором только что совершившемся событии, например об аварии или создании ее угрозы. Этот запрос может передаваться по выделенным линиям или путем возбуждения специальных состояний физического канала. В последнем случае заказ не должен искажать кадры, передаваемые по магистрали. В АСУ может не применяться функция заказа, если требования по отклику в реальном времени удовлетворяются иным способом. Станция с функцией заказа называется **з а к а з ч и к о м (demandor)**.

Функция контроля выполняется супервизором (**supervisor**). В задачи этой станции входят: управление доступом к каналу путем установления активного инициатора для каждого шага обмена данными, разрешение состязаний между активными заказчиками и (или) претендентами на статус инициатора за время, приемлемое в конкретной АСУ, контроль работы инициатора в целях обнаружения и обработки ошибок;

ограничение длительности и количества шагов обмена данными в целях исключения перегрузки магистрали каким-либо инициатором; обеспечение непрерывности работы магистрали при отказе какого-либо активного инициатора;

контроль эффективности действующего информационного канала;

включение в надлежащей ситуации, например при возникновении в действующем информационном канале неустранимых ошибок или повреждении действующего физического канала, другого возможного информационного канала.

В типичной АСУ время доступа для активного заказчика не должно превышать 2 мс (если время передачи кадра не превышает 1,5 мс), а время доступа для претендента не статус инициатора - 20 мс (если время передачи кадра не превышает 5 мс).

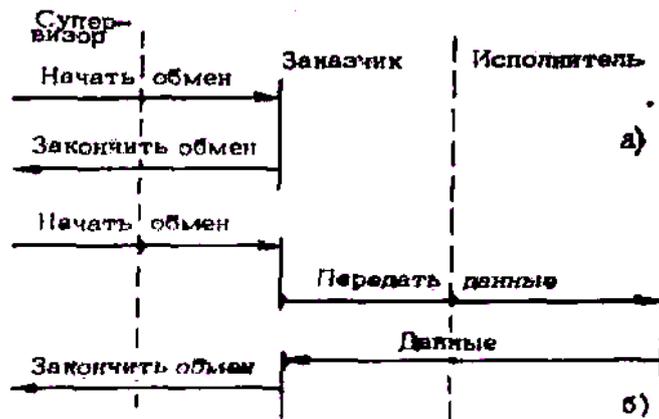
Наконец, высший ранг в функциях магистрального протокола принадлежит **распорядителю** магистралью. Распоряжение должно: передавать управление магистралью путем назначения активного супервизора; разрешать состязания между претендентами на статус супервизора за время, совместимое с требованиями конкретной АСУ (в типичной АСУ время доступа для претендента на статус супервизора не должно превышать 1с); обеспечивать непрерывность работы магистрали при отказе активного супервизора.

Станция, распоряжающаяся магистралью и именуемая **распорядителем** (manager), должна также контролировать эффективность работы магистрали (а возможно, и всех подключенных к ней станций) и регистрировать показатели этой эффективности.

Протоколы с произвольным порядком поступления запросов от станций (в ранге инициатора) это централизованные протоколы, реализуемые с участием супервизора, который устанавливает очередь на занятие магистрали. Запросы на доступ к магистрали здесь могут передаваться супервизору либо по отдельному физическому каналу, либо по временному окну, выделяемому супервизором. Поскольку при инициативном поступлении запросов супервизор должен иметь сведения не только о станции-источнике, но и о станции назначения каждого пакета, устанавливаемого в очередь, здесь требуется увеличивать длину кадров запроса и полей адресации в информационных кадрах.

По принципу произвольного порядка поступления запросов построен протокол доступа к магистрали РВУ. Здесь предусматриваются станции трех рангов: исполнителя, заказчика и супервизора (в определениях Proway, в стандарте PDV применены названия передающей, заказывающей и управляющей станции соответственно). Предполагается, что исполнитель и супервизор способны также к приему, а заказчик - к приему и ответу. Кроме того, могут применяться специальные заказчики, в которых отсутствует функция ответа.

Схема использования магистрали РВУ подчиненными станциями иллюстрируется на рис.35. В одном случае (рис.35,а) заказчик, получивший магистраль в свое распоряжение, тут же отказывается от пользования ею.



**Рис. 35. Общая схема протокола доступа к магистрали PDV:**  
 а - отказ заказчика от пользования магистралью; б - заказчик получает данные от исполнителя

По весьма сложному протоколу работает одна из первых промышленных магистралей АСУ – магистраль Hiway комплекса TDC-2000. В магистрали, управление которой централизовано, имеются станции следующих рангов: супервизор (называемый директором магистрали), заказчики, инициаторы и исполнители. Общее число станций (не считая директора) может достигать 28. Поскольку сеть TDC-2000 обычно состоит из нескольких магистралей, директор как главная станция всей сети выполняет также функции управления сетью, характеризующиеся далее. Ранг заказчика присвоен станциям пультов оператора и ЭВМ верхнего уровня. Заказчики упорядочены по приоритетам доступа к магистрали в пределах четырех приоритетных уровней. Заказы на статус инициатора передаются не по магистрали, а по специальным проводам, число которых может составлять от 4 до 16.

Статус инициатора не может быть отобран у станции-заказчика до истечения 80 мкс после окончания инициированного ею обмена.

Инициаторы могут стать активными только после освобождения магистрали всеми заказчиками; тогда супервизор начинает цикл опроса претендентов на статус активного инициатора. За один шаг опроса, занимающий 8 мкс, опрашивается одна станция. Пока магистраль свободна, опросы повторяются с периодом 10 мс; если в течение 30 мс супервизор не может повторить цикл опроса, выдается сигнал тревоги, указывающий на отказ магистрали.

По стандарту ЕСМА-82 доступом к магистрали управляет подуровень канального уровня, названный подуровнем управления линией связи. Стандарт различает два режима протокола этого подуровня: нормальный и с состязанием. В нормальном режиме, когда подуровень логического управления каналом выдает запрос собственной станции на передачу кадра, кадровый подуровень конструирует кадр доступа, добавляя к данным, полученным с верхнего подуровня, номер кадра, используемый затем для контроля, и кадр по битам спускается на подуровень управления линией связи и передается в линию. При приеме подуровень управления линией связи получает с физического уровня сигнал о приеме каждого бита; биты преамбулы, относящиеся только к кадру

физического уровня, отбрасываются. По окончании кадра доступа исчезает несущая, и тогда принятый кадр передается на кадровый подуровень.

При возникновении конфликта передача не прекращается сразу же после его обнаружения; вместо этого конфликт «усиливается», для того чтобы дать время на заведомое обнаружение всех станций, участвующих в конфликте. «Усиление» достигается передачей дополнительных битов так называемой пробки (jam) - от 32 до 48 бит.

Если попытка передачи не удалась станции из-за конфликта, то станция повторяет ее до успешной передачи, но не более 16 раз. Повторные попытки подчиняются рандомизированному расписанию, получаемому процессом, который назван разветвленным двоичным экспоненциальным отступлением (truncated binary exponential backoff). Время отступления, т. е. задержки следующей попытки передачи, составляет  $\gamma T$ , где  $\gamma$  - случайное целое число, равномерно распределенное от 0 до  $2^k$ , причем  $A$  не может превышать 10. Отрезок времени  $T$ , называемый длительностью окна, кратен времени, требуемому для передачи 512 бит.

Предложен ряд усовершенствованных протоколов доступа с состязанием, предварительно сокращающих вероятность конфликтов, благодаря чему уменьшаются потери времени, связанные с повторением передачи. Одно из них содержится в популярном протоколе доступа к сети в варианте, приспособленном для магистрали. Станция перед началом передачи контролирует сигнал в магистрали и не приступает к передаче, пока магистраль занята. Попытка передачи повторяется спустя некоторое время. Контроль сигнала в линии продолжается и после начала передачи, и если в процессе передачи обнаруживается конфликт, то передача прекращается, с тем чтобы попытка возобновилась через наперед заданное время.

В последние годы предлагались и адаптивные протоколы доступа, в которых параметры цикла или временных окон меняются в зависимости от того, как занималась магистраль или отдельные окна в одном или нескольких предыдущих циклах.

Наконец, возможны протоколы доступа, сочетающие различные принципы, изложенные выше. Так, в протоколе [7] все станции разделены на группы. Цикл обмена разделен на окна, и для каждой группы отведено фиксированное число окон. Станции одной группы состязаются за доступ к свободным окнам своей группы по принципу занятия свободных окон. Благодаря разделению станций на группы снижается время ожидания и исключается возможность полной блокировки группы, как это могло бы произойти при состязании между всеми станциями магистрали.

## 2. Управление сетью

К управлению сетью относятся задачи прокладки маршрута в сети и продвижения пакета по маршруту.

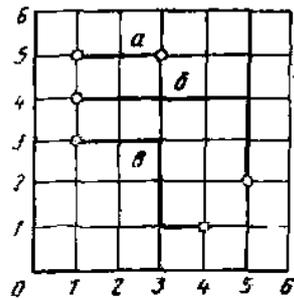
Вычислительные сети, применяемые в промышленных АСУ, отличаются от сетей другого назначения сравнительной простотой, маршрут передачи пакета в них между станциями, как правило, однозначен, а это обычно приводит к

вырождению функций сетевого уровня АСУ. Существуют лишь отдельные предложения о развитии функций сетевого уровня в промышленных АСУ. Маршрутизация же применяется не в функции текущей загрузки станций сети, как это имеет место в больших сетях ЭВМ, например ARPANET, а лишь в связи с нарушениями целостности первоначальной сети.

Одна из основных форм управления на сетевом уровне связана в промышленных АСУ с тем, что применяемая сеть состоит из нескольких магистралей, и задача протокола сетевого уровня - организация связи между магистралями. Стандартом Proway функция управления сетью, состоящей из нескольких магистралей, возлагается на станцию, имеющую ранг директора. Согласно стандарту к функциям директора относятся: установление ответственности за распоряжение каждой магистралью сети путем присвоения статуса активного распорядителя одной из станций магистрали; разрешение состязания между претендентами на статус распорядителя за время, удовлетворяющее требованиям данной АСУ, обеспечение функционирования каждой магистрали при отказе ее активного распорядителя; организация передачи данных между станциями (при необходимости - с промежуточным хранением пакетов).

Сеть комплекса TDC-2000 может содержать до трех ветвей, из которых каждая состоит из двух параллельных магистралей, соединяющих одни и те же станции. Все магистрали соединены со станцией, имеющей ранг директора сети. Эта станция управляет обменом между ветвями, ретранслируя пакеты, адресованные из одной ветви в другую. Сеть состоит из матрицы магистралей, так что каждая станция соединена с двумя магистралями - горизонтальной (строкой матрицы) и вертикальной (столбцом). Для организации обмена между станциями заголовок пакета должен содержать поля с адресом источника и получателя. Адреса в такой матричной сети задаются номерами строк и столбцов. Совпадение одного из адресов (строки или столбца) источника и получателя означает, что они связаны между собой одной магистралью, горизонтальной или вертикальной, и по этой магистрали и передаются пакет и подтверждение о его приеме. Такой обмен иллюстрируется маршрутом а на рис. 36 - между станциями (1, 5) и (3, б).

Протокол, сконструирован таким образом, что при несовпадении ни одного из адресов источника и получателя протокол выбирает одну из двух магистралей, соединенных с источником, и посылает по ней пакет. Пусть в сети на рис.36 пакет адресован со станции (1, 4) на станцию (5, 2) и избрана сначала горизонтальная магистраль 4. На уровне магистрального протокола адрес получателя равен 5, поэтому кадр по магистрали будет доставлен на станцию (4, 5). Но на сетевом уровне адрес по вертикали равен 2, поэтому станция (1, 4) после передачи пакета по горизонтальной магистрали 4 не получит подтверждения и станция (5, 4) ретранслирует пакет по вертикальной магистрали 5. После приема пакета получателем (5, 2) подтверждение пойдет обратно к источнику (1,4) по тому же маршруту.



**Рис. 36. Маршрут в матрице магистралей**

На случай нарушения целостности сети протокол предусматривает, что при неполучении подтверждения источником передача от источника повторяется, причем адрес на уровне первой магистрали смещается на соседний относительно адреса получателя, а при очередной неудаче происходит переход с горизонтальной магистрали на вертикальную или обратно. Пример маршрута, прокладываемого при отказе вертикальной магистрали со станции (1, 3) на станцию (4, 1), представлен ломаной в на рис.36. Технически в рассматриваемом сетевом протоколе отсутствие одного из адресов приводит к тому, что пакет адресуется всем станциям магистрали (например, горизонтальной при отсутствии адреса по горизонтали), а отсутствие обоих адресов - к тому, что пакет адресуется всем станциям сети. Этот прием и используется при поиске обходных маршрутов при нарушении целостности матричной сети. Во избежание повторного приема пакеты нумеруются. Наряду с построением сетей из нескольких магистралей в промышленных АСУ прибегают также к соединению колец. При этом весьма типична фиксированная маршрутизация. Примером может служить выбор маршрута по так называемому семафору - части адреса назначения, указывающей, по какому из направлений, исходящих от данной станции, должен быть направлен пакет. Управление конфигурацией сети при нарушении ее целостности рассмотрим на примере сетевого протокола RDC (Really Distributed Control System - «истинно распределенная АСУ»). Сеть состоит из двойного кольца: по одному кольцу все станции проходятся в одном направлении, по другому - в противоположном (рис.37,а). Ключ, шунтирующий выход каждой станции на сеть, позволяет преобразовать участок кольца в магистраль.

Сетевые функции выполняются в RDC двумя расслоенными протоколами. Протокол нижнего уровня SUP выясняет состояние сети, а протокол верхнего уровня KFP в зависимости от этого состояния изменяет ее конфигурацию. При отказе отрезка одного из колец организуется обмен по сохранившемуся кольцу (рис.37,б). При выходе из строя станции сеть реконфигурируется в одно или несколько одинарных колец, дважды проходящих в разных направлениях через каждую входящую в их состав станцию (рис.37,в). В такого рода кольцах осуществляется так называемый маятниковый обмен - из конца в конец и обратно (например, С5 - С6 - С7 - С6 - С5 и т. д. на рис. 37, в).

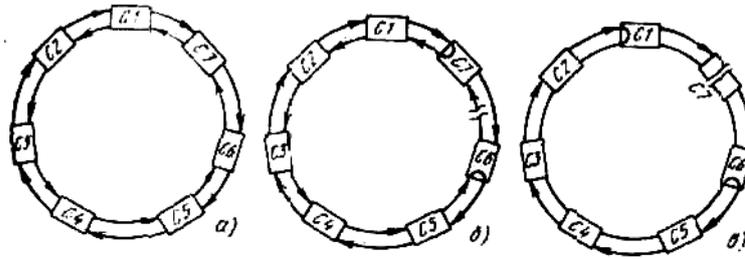


Рис. 37. Двойное кольцо с противоположными направлениями передачи:  
 а - исходная структура; б - реконфигурация при отказе звена; в - реконфигурация при отказе станции

К функциям сетевого протокола относится также определение пути передачи пакета по сквозному (транспортному) каналу, устанавливаемому в сети и проходящему через несколько промежуточных станций. Между каждой парой соседних станций сквозного канала устанавливается информационный канал (link, т. е. звено), по которому передача ведется в соответствии с протоколом канального уровня. Построение же сквозного канала относится к функциям сетевого протокола.

В магистральных физических структурах сквозной канал между станциями отправления и назначения строится непосредственно, без участия промежуточных станций, поэтому имеет смысл выделение магистрального уровня, сопрягающегося сверху с прикладными уровнями, в первую очередь с уровнем сеансов. Протоколы магистрали были рассмотрены выше.

### 3. Протоколы в цепных структурах

В цепных структурах, в том числе в кольцевой, сквозной канал устанавливается однозначно еще на физическом уровне, поэтому сетевой протокол для цепи обычно сводится к тому, что когда станция получает право воспользоваться единственным информационным каналом, ведущим к следующей по цепи станции, она передает ей пакет, избранный транспортным протоколом для передачи из образовавшейся на ней очереди. Получив пакет, станция анализирует содержащийся в нем адрес получателя и, если пакет не адресован данной станции, пропускает его дальше, к следующей станции цепи. В цепных структурах, как и в магистральных, протокол должен устанавливать порядок доступа различных станций к информационным каналам передачи данных.

Поскольку практическое применение получили только кольцевые цепные структуры, соответствующие сетевые протоколы разработаны лишь для колец. Они различаются прежде всего тем, предусматривается ли в них промежуточное хранение пакетов (feedforward transmission – передача с промежуточным хранением), т. е. задержка пакетов на транзитных станциях. Если такая задержка не предусматривается, протокол управления доступом к кольцу, за исключением отдельных специальных деталей, не отличается от управления доступом к магистрали. Об этом уже говорилось выше, где характеризовались

соответствующие протоколы, причем делались необходимые оговорки, когда речь шла о кольце. Промежуточное хранение может потребоваться, если на станцию одновременно поступают из разных источников два пакета, требующие дальнейшей передачи по одному информационному каналу. В кольце подобный конфликт возникает между пакетом, поступившим с предыдущей станции кольца, и пакетом, источником которого является данная станция. Другое решение может заключаться в том, что промежуточное хранение отсутствует, и конфликт между двумя пакетами разрешается в пользу одного из них, тогда как другой пакет уничтожается, а станция-источник, не получив подтверждения о приеме от конечного получателя, повторяет попытку передачи. Существует, впрочем, мнение, что для АСУ подходят протоколы с неопределенными исходами конфликтных ситуаций из-за трудностей испытания таких АСУ на соответствие техническим требованиям по времени передачи сообщений. Однако с таким мнением вряд ли можно согласиться. Чаще всего кольцевые протоколы предусматривают промежуточное хранение пакетов, хотя имеются и кольца без промежуточного хранения.

Протоколы управления доступом к кольцу могут основываться на присвоении приоритета пакетам. Действие приоритета может распространяться на все кольцо (глобальный приоритет) или ограничиваться только одной станцией (локальный приоритет). При применении глобальных приоритетов станция отдает предпочтение передаче пакета с самым высоким приоритетом и тогда пакеты с более низким приоритетом приходится сохранять на станции до освобождения очередного окна. Если же приоритет локальный, то он касается только очередности выхода пакетов со станции, циркуляция же пакетов по кольцу не нарушается.

Приоритеты могут быть статическими, т. е. не изменяющимися в процессе работы кольца, или динамическими, т. е. изменяющимися. Их можно присваивать по разным признакам: в зависимости от типа сообщения; расстояния, которое остается пройти пакету по кольцу до станции назначения; длины сообщений; времени ожидания пакетом отправления и др.

Примером приоритетного протокола может служить кольцевая сеть DLCN. Кольцо управляется по так называемому принципу в в е д е н и я р е г и с т р а . Требуемый передачи пакет загружается в регистр, который вводится в кольцо в промежутке между прохождением двух последовательных пакетов либо когда в кольце вообще нет пакетов.

Протоколы с произвольным порядком запросов и приоритетные протоколы страдают тем недостатком, что в них отдельные станции при резком увеличении нагрузки могут оказаться заблокированными, так что они в течение длительного времени не будут иметь доступа к кольцу, а это может привести к переполнению их буферной памяти.

Статическое временное разделение, опрос, захват окна и приоритетное кольцо DLCN неэффективны при большом числе станций, так как приводят к значительным потерям времени на ожидание передачи.

Кольца со статическим временным разделением, с опросом, с захватом окна менее надежны, чем остальные, ввиду централизованного принципа управления,

наибольшая надежность - у колец с состязанием за право доступа. К самым дешевым по реализации следует отнести кольца со статическим временным разделением, с произвольным порядком запросов и с состязанием, к самым дорогим - кольца с захватом окна. Наконец, легче других допускают расширение с вводом новых станций кольца с произвольным порядком запросов, с захватом окна, с состязанием, с приоритетными протоколами, труднее других - кольца со статическим временным разделением и с опросом.

К функциям транспортного уровня АСОС, включенного стандартом Proway в сетевой уровень, относятся: формирование из сообщений прикладного уровня пакетов, подлежащих передаче по сети; упаковка их в транспортный конверт, содержащий адреса станций назначения и отправления, данные для контроля; извлечение пакета из транспортного конверта после приема на станции назначения; контроль правильности передачи по сквозному (транспортному) каналу. Следует, впрочем, отметить неоднозначность применяемой терминологии; так, уровень, называемый сетевым в архитектуре соединения открытых систем, некоторые авторы называют транспортным, а транспортный уровень этой архитектуры - уровнем сквозного канала.

В магистральных информационный канал устанавливается непосредственно между станциями отправления и назначения, поэтому здесь упаковка в конверт и контроль на канальном уровне исключают нужду в соответствующих операциях на транспортном уровне и за последним остается лишь задача согласования длины сообщения прикладного уровня с окном, предоставляемым магистралью.

Это относится также к кольцам с синхронной передачей. Однако и в кольцах с асинхронной передачей, и в сетях более общей структуры транспортный протокол должен выполнять все предписанные ему функции.

Если станция снабжается буфером, накапливающим ожидающие передачи сообщения, то управление очередью относится к прерогативам транспортного протокола. Эта задача решается в сетях.

Охарактеризуем транспортный протокол АСУ в сети общего вида. Структура поля управления сетевым уровнем, входящего в кадр на рис.30, показана на рис.38.

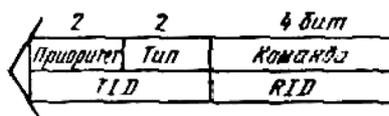


Рис. 38. Структура поля управления сетевого уровня в протоколе

На транспортном уровне различаются четыре типа пакетов, которые на канальном уровне все передаются информационными кадрами.

Информационные пакеты пропускаются на прикладной уровень как сообщения, если они содержат данные или ответы. Четыре бита команды определяют: ожидается ли ответ, если ожидается, то иницируется отсчет времени; содержит ли пакет данные, если содержит, то вводится в действие поле TID с опознавательным

номером передаваемого пакета; должен ли пакет генерировать извещение о неправильном приеме; является ли пакет ответом, если да, то вводится в действие поле RID с номером ответа, используемым для прекращения отсчета времени и идентификации ответов. Извещения о неправильном приеме посылаются с промежуточной станции, но источнику они представляются поступающими со станции назначения и имеют значение номера RID, равное TID отправленного пакета. Характер ошибки описывается полем команды. Другого рода извещения о неправильном приеме или ожидаемые ответы генерировать нельзя, ибо тогда существовала бы опасность того, что отказ порождал бы бесконечную последовательность извещений о неправильном приеме.

Транспортный протокол ЕСМА (по стандарту ЕСМА-72) первоначально относился к сети общего вида, тем не менее предполагается его применение и в локальной сети. К функциям транспортного протокола по этому стандарту относятся установление транспортного соединения, передача данных и разрыв соединения. При установлении транспортного соединения выбираются ссылки, идентифицирующие партнеров, вступающих в соединение; выбираются сетевые услуги; решается вопрос о целесообразности мультиплексирования транспортных соединений, т. е. Поочередного установления нескольких транспортных соединений внутри одного сетевого соединения, и о целесообразности расщепления транспортного соединения между несколькими сетевыми соединениями; устанавливается оптимальная длина транспортного пакета; транспортные адреса преобразуются в сетевые.

Передача данных сопровождается (если это предусмотрено) мультиплексированием транспортного соединения или его расщеплением, управлением потоком пакетов по транспортному соединению, включая внеочередную посылку пакетов экстренных сообщений, обнаружением ошибок, заданием начала и конца каждого транспортного пакета.

Транспортное соединение реализует тот или иной набор функций, определяемый отнесением транспортных услуг к одному из пяти классов (0 - 4). Класс услуг определяется при установлении транспортного соединения в процессе «переговоров» между двумя станциями, входящими в соединение. В ходе переговоров устанавливаются класс протокола и длина транспортного пакета.

Стандарт ЕСМА-72 задает только четыре класса: 0, 2, 3 и 4. Класс 0 обеспечивает лишь передачу данных с их сегментированием по кадрам и сообщение об ошибках в исполнении протокола. Далее с ростом номера класса функции наращиваются. В классе 2 выполняется мультиплексирование транспортных соединений в одном сетевом соединении, имеется возможность управлять потоком транспортных пакетов и вне очереди передавать экстренные сообщения. Класс 3 позволяет исправлять ошибки, обнаруженные на сетевом уровне, что достигается хранением копий переданных транспортных пакетов для получения подтверждения об успешном приеме. Наконец, класс 4, который служит для работы с сетями, имеющими недопустимо высокую интенсивность остаточных ошибок, обнаруживает ошибки, пропущенные на сетевом уровне, и исправляет их; к таким ошибкам относятся потеря или повреждение

транспортного пакета, повторный Прием уже принятого ранее пакета, прием пакетов с нарушением порядка их передачи.

На уровне транспортного протокола устанавливалась очередность передачи пакетов по кольцевой сети, в зависимости от их приоритета. В пределах одного обращения пакетов по кольцу передаются пакеты только одного приоритетного уровня, после чего уровень приоритета снижается на единицу и запускается новый цикл обращения по кольцу.

Следует, впрочем, отметить, что в большинстве распределенных АСУ(транспортный уровень архитектуры не выделяется из сетевого, на котором выполняется контроль правильности передачи; в этом случае функции упаковки в транспортный конверт и распаковки конверта сами собой отпадают. Возможность обойтись без специальных функций транспортного уровня объясняется прежде всего применением в них одинаковых процессорных станций.

### Контрольные вопросы

1. Что такое магистраль? [Л.7, 85-86].
2. Что является задачей протокола, управляющего магистралью? [Л.7, 87-88].
3. Расскажите о протоколах передачи по магистрали со с статическим разделением времени. [Л.7, 89-90].
4. Объясните о рангах в функциях магистрального протокола. [Л.7, 91-93].
5. Начертите общую схему протокола к магистрали РВУ. [Л.7, 93-94].
6. Какие задачи относятся к управлению сетью? [Л.7, 94-95].
7. Приведите пример к типичной фиксированной маршрутизации. [Л.7, 96-98].
8. Расскажите о кольцевых цепных структурах. [Л.7, 99-102].
9. Приведите пример для приоритетного протокола. [Л.7, 103-105].
10. Стандарт ЕСМА-72 (транспортный протокол ЕСМА). [Л.7,106-107].

Лекция №7  
**Управление сеансами обмена и представлением данных.  
Прикладные функции**

План

1. Управление сеансами обмена и представлением данных.
2. Способы реализации функций представления данных.
3. Прикладные функции.

**1. Управление сеансами обмена и представлением данных**

Управление сеансами обмена и представлением данных относится в архитектуре распределенной АСУ к надсетевым уровням и поэтому в отличие от систем с традиционной архитектурой характерно для распределенных АСУ лишь постольку, поскольку этого требует новый подход к созданию АСУ обусловленный современными средствами обработки и представления данных..

Сеанс представляет собой процедуру информационного взаимодействия прикладных процессов для целей обмена данными при соблюдении временных ограничений.

Различают односторонний, двусторонний разновременный и двусторонний одновременный сеансы обмена. В первом случае один процесс постоянно является источником, а другой - приемником данных. При двустороннем обмене оба прикладных процесса могут быть в одно и то же время или поочередно источниками и приемниками данных.

Для осуществления сеанса обмена необходимо выполнить две функции: синхронизацию двух прикладных процессов и передачу данных.

Под синхронизацией понимают обеспечение временной взаимозависимости процессов, вводимой в целях достижения заданной очередности протекающих явлений.

Все разнообразие известных алгоритмов синхронизации можно свести к двум механизмам, известным как отношения производитель - потребитель и взаимное исключение.

В первом случае временное ограничение состоит в том, что потребитель не может использовать данные до тех пор, пока производитель их не подготовит. Производитель формирует сообщение, снабжая его при необходимости адресом получателя, и помещает в некоторый буфер. При наличии сообщения потребитель начинает обработку данных. В противном случае освобождаются ресурсы в пользу других потребителей (обработка данных приостанавливается), которые действуют аналогичным образом. Этим обеспечивается требуемая

последовательность процессов обработки информации производителем и потребителем, т. е. их синхронизация.

В том случае, когда несколько процессов обращаются к одним и тем же данным или устройствам, возможна ситуация, когда последние на какое-то время оказываются не подготовленными для использования. Такое состояние может возникнуть, например, при обновлении данных или при одновременном обращении двух процессов к устройству ввода-вывода с одним буфером, и тогда процесс, получивший доступ к ресурсу, должен воспрепятствовать всем другим процессам воспользоваться тем же ресурсом. Иначе говоря, один процесс должен исключить все другие процессы.

Распространенный способ решения задачи взаимного исключения известен как семафор. Общим данным (устройствами) ставится в соответствие некоторая целочисленная переменная (семафор). Каждый процесс, захватывающий ресурс, начинает работу с увеличения содержимого  $s$  на единицу и заканчивает ее вычитанием единицы по завершению работы. Очередной процесс-пользователь получает доступ к данным только в том случае, если  $s = 0$ . Частным случаем семафора является механизм флагов, при котором переменная может иметь только два значения: 1 (флаг выставлен) и 0 (флаг снят).

Комитет ТС 8 Европейской рабочей группы по промышленным вычислительным системам (EWICS) предложил универсальный механизм, сочетающий преимущества обоих способов. Он базируется на принципе «почтового ящика», представляющего собой буфер, для которого предусмотрены операции записи (INC) и чтения (DEC) данных по определенным правилам. Буфер называют синхронизационным элементом, а записываемые (считываемые) данные - информационным элементом.

Операция INC (S,S) помещает информационный элемент с именем 1 в синхронизационный элемент с именем 5. Соответствующая операция DEC (5,1) считывает информационный элемент, 1 из синхронизационного элемента 5. Если элемент 1 содержится в 5, то операция чтения завершается, а процесс, в котором она осуществляется, получает данные 1, и его исполнение может продолжаться дальше. Если запрашиваемого элемента 1 в синхронизационном элементе 5 нет, процесс, пытающийся считать эти данные, дальше исполняться не может. Он прерывается, т. е. освобождает занятые ресурсы в пользу других процессов. Происходит переключение процессов. Операция DEC завершается позже, когда элемент 1 окажется в буфере 8. Таким образом, с помощью операций INC и DEC осуществляется синхронизация процессов. Конкретный механизм, реализуемый этой парой операций, зависит от свойств обоих элементов. Если синхронизационный элемент является буфером типа магазина (стека), а информационный - сообщением, то реализуется механизм производитель - потребитель. Если синхронизационный элемент оперирует с данными типа шаблона, то реализуемый механизм соответствует семафору. Информационный элемент в этом случае является своеобразным паролем, по которому синхронизационный элемент разрешает доступ к общим переменным.

В однопроцессорной системе информационный элемент передается через память, общую для всех процессов. В многопроцессорных комплексах для этой цели

может использоваться также дополнительный вход блока памяти, через который информация записывается в локальную память конкретного процессора. Процессор анализирует запись в подходящий для него момент времени. Возможно также использование механизма прерывания: процессор-получатель прерывает работу по сигналу процессора-отправителя, принимает данные, а затем анализирует их в подходящий момент времени.

В многомашинных системах не существует общей памяти, в которой можно было бы располагать синхронизационные элементы. Поэтому при выполнении операции синхронизации сначала решается вопрос, где находится соответствующий объект. Если он расположен в собственном узле, операция выполняется традиционно. В противном случае заказ передается через локальный синхронизационный элемент в сеть передачи данных. При этом процесс-отправитель указывает адресатов (синхронизационный элемент и процесс-получатель) по именам, т.е. независимо от их местонахождения в сети. Заказ от локального синхронизационного элемента, переданный по сети передачи данных, проводится описанными ранее методами через все уровни сети, достигая через устройства ввода-вывода и каналы связи нужного узла сети. Здесь он обрабатывается процессом-адресатом, который таким образом синхронизируется с процессом-отправителем. Если отправитель не ждет ответа, то работа на этом завершается. В противном случае процесс-отправитель выполняет операций DEC над локальным синхронизационным элементом, существующим для каждого процесса. Когда операция синхронизации завершается, процесс-отправитель получает ответ в локальной памяти из синхронизационного элемента. С помощью операций INC и DEC реализуются также более сложные алгоритмы синхронизации, используемые в языках реального времени.

Для упрощения сеансов обмена в распределенных АСУ прикладные процессы делят на активные и пассивные. Пассивные лишь готовят информацию для других процессов, т.е. являются производителями данных. Активные процессы, наоборот, сами разыскивают предназначенные для них данные. Для организации сеанса обычно используется принцип почтового ящика. Сеанс начинается с формирования сообщения пассивным процессом. Сообщение снабжается адресом активного получателя и помещается в буфер (почтовый ящик). Активные процессы с большой частотой опрашивают буфер, проверяя наличие в них сообщений для себя. Обнаружение такого сообщения соответствует соединению процесса-отправителя с процессом-получателем, после этого данные из буфера передаются процессу-получателю.

Обмен данными после соединения (синхронизации) прикладных процессов осуществляется с использованием функций представительного уровня.

Задачей уровня является предоставление взаимодействующим прикладным процессам данных в требуемом формате. Уровень представления выполняет по отношению к вышележащему прикладному уровню две основные функции; получение информации данным прикладным процессом от других процессов, в том числе и от процессов ввода; передачу информации от данного прикладного процесса другим процессам, в том числе и процессам вывода. Работу уровня можно представить в виде четырех последовательных фаз. В фазе инициализации

выполняются процедуры подготовки к сеансу связи прикладных процессов. Затем следует фаза управления отображением, в которой выбирается форма отображения данных, после чего в фазе передачи данные или команды преобразуются в нужную форму и воспринимаются процессом-получателем. Фаза окончания сеанса завершает работу уровня представления. Выполнение этих операций обеспечивается, во-первых, сервисными функциями нижележащего уровня сеансов обмена и, во-вторых, внутренними функциями самого уровня представления, которые охватывают накопление, хранение, преобразование форматов, защиту данных, организацию доступа к ним и т. п.

## **2. Способы реализации функций представления данных**

Способы программной реализации функций уровня представления зависят от объема, состава и размещения данных в системе. В состав данных, перерабатываемых в распределенной АСУ, входят: 1) сообщения о событиях; 2) текущие значения измеряемых и расчетных технологических переменных; 3) данные о текущем состоянии (статусе) аппаратуры управления; 4) параметры настройки (конфигурирования) программ контроля, управления и регулирования, определяющие режим работы технологического оборудования; 5) параметры настройки, определяющие форматы представления данных на средствах отображения; 6) внутренние данные станции.

В АСУ Отданные размещают в основном в соответствии с местом их возникновения, иначе говоря, данные распределены по станциям, где они вводятся или вычисляются

Текущие значения переменных, характеризующих состояние соответствующей части процесса, хранятся в локальных технологических станциях. Там же находятся параметры настройки, касающиеся данного участка объекта управления. Таблицы статуса аппаратуры отражают состояние технических средств всего комплекса и дублируются во всех станциях. Эти сведения нужны для мероприятий по реконфигурации системы в аварийных ситуациях, реализуемых на уровне локальных станций. Сообщения, как правило, не хранятся постоянно в локальных станциях, а теряются после обработки (передачи в операторскую станцию или вывода с квитиowaniem их оператором). Для восстановления данных в аварийных ситуациях (при рестартах и реконфигурациях) применяется копирование отдельных таблиц, буферов и т. п. в различных локальных станциях. В проекте RDS каждая локальная станция содержит помимо данных своего участка аналогичную информацию (кроме расчетных переменных) еще одного участка, обслуживаемого соседней локальной станцией. Однако обновляются здесь лишь те данные, которые в случае реконфигурации (т. е. при отказе локальной станции) невозможно получить с объекта управления.

Состав данных, хранимых в операторской станции, не является постоянным. В каждый момент времени здесь находятся копии таблиц измеряемых переменных и параметров двух станций, состояние которых отражается на экранах видеотерминалов. Помимо этого во внешней памяти операторской станции хранятся данные от всех локальных технологических станций, но лишь в объеме



**Рис. 39. Принцип изображения данных в различных моделях:  
а - реляционная модель; б - иерархическая модель; в - сетевая модель**

Различают три модели данных, которые отличаются числом и семантикой операторов языка запросов. Рассмотрим различия между ними на простом примере. Пусть имеются две технологические переменные - расход и температура газа, характеризующиеся именем, текущим численным значением, допустимыми граничными значениями, формой, цветом и положением на экране символа, индицирующего текущее состояние переменной. Возможны три различных принципа изображения этих данных. Наиболее естественным представляется реляционная модель, в основе которой лежит табличное представление как специальная форма математического отношения (рис. 39, а). Все данные, касающиеся обеих технологических переменных, представлены в форме двух взаимосвязанных таблиц. В иерархической модели данные имеют древовидную структуру, базирующуюся на отношении «отец - сын». Иерархическая модель для рассматриваемого примера показана на рис. 39, б.

Сопоставление древовидной структуры с таблицами на рис.39,а показывает, каким образом одна модель преобразуется в другую. Рисунок 39,в иллюстрирует третью возможную модель данных -сетевую. Она базируется на отношении «предшественник -последователь» и строится с помощью ссылок.

Уже из приведенного примера видно, что все три модели данных принципиально пригодны для систем управления. Выбор той или иной модели зависит от требований, предъявляемых к средствам управления данными в распределенных АСУ со стороны прикладных процессов. Эти требования сводятся в основном к следующему: независимости данных от прикладных процессов, независимости программ от размещения данных на физических носителях информации, целостности данных, защите данных и эффективности.

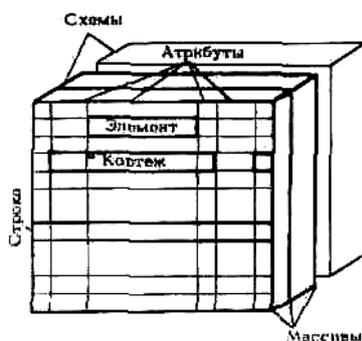
Возникающая в распределенных АСУ на уровне представления при применении реляционных банков данных, состоит в необходимости согласования логической структуры данных с физической структурой объекта управления.

Пример реляционной базы данных приведен на рис.40. Таблица здесь именуется массивом и задает некоторое отношение над данными. Она идентифицируется именем и состоит из и строк и т столбцов. Столбцы называется а т р и б у т а м и и идентифицируются именами, На рис.39,а в первой таблице атрибутами являются имя, текущее и нормативное значения, размерность и т. п. Каждый атрибут одного и того же массива имеет одинаковое число элементов, и все элементы (они лежат на пересечении столбца и строки массива) одного и того же атрибута имеют одинаковую длину. Атрибут, у которого все элементы попарно различны, используется в качестве ключевого. Значения элементов такого атрибута однозначно идентифицируют строки массива. Строки разыскиваются по заданному ключевому значению с помощью специальных процедур путем определения положения ключевого значения. Совокупность отношений называют с х е м о й . Схема совместно с массивами составляет базу данных и

обозначается именем. База данных может объединять массивы с различным числом строк и столбцов. Совокупность программ обслуживания баз данных составляет систему управления базами данных (СУБД), которая совместно с базами данных образует банк данных АСУ.

Набор данных одной и той же строки называется *к о р т е ж е м*. Он составляет основной адресуемый элемент данных. Кортеж идентифицируется именем базы данных, массива, ключевого атрибута, именами других атрибутов и, наконец, ключевым значением (элементом атрибута). Сообщив эти данные через структурно-независимый интерфейс, прикладной процесс получает доступ к кортежу независимо от его размещения в базе данных.

Доля затрат, приходящихся на систему управления базой данных (СУБД) локальной станции, составила: по памяти для хранения программ 38%; по памяти для хранения данных 57%, по затратам машинного времени 16% (при общей средней загрузке ЭВМ 40%).



**Рис. 40. Структура реляционной базы данных**

Выполнение требований независимости и защиты данных предопределяет централизацию управления ими. Между тем полная централизация, т. е. сосредоточение приема и обработки всех запросов в одной станции распределенной АСУ нецелесообразна как с точки зрения эффективности, так и с точки зрения надежности. Поэтому систему управления данными разделяют на части таким образом, чтобы в каждой технологической станции распределенной АСУ находилась локальная база данных, составляющая часть общей базы. Этот подход реализован, например, в аппаратно-программном комплексе фирмы ВВС. Комплекс предназначен для управления объектами, находящимися на сравнительно большом удалении друг от друга (до 100 узлов на площади примерно до 100000 км<sup>2</sup>).

Эти данные были получены в основном опытным путем. По оценкам разработчиков объем программ сильно зависит от числа дополнительных функций (контроля доступа, защиты данных и т. п.). Затраты памяти существенно увеличиваются по сравнению с системами без СУБД из-за дополнительных таблиц

и буферов. В целом затраты оперативной памяти (за исключением места для хранения самих данных) не выше, чем для реализации системы управления файлами.

Вместе с тем приведенные выше цифры свидетельствуют о сравнительно высоких затратах на реализацию СУБД. По этой причине в комплексах с магистралями, перекрывающими лишь несколько километров и имеющими высокую скорость передачи данных, шире применяется структурно-зависимый интерфейс СУБД.

### 3. Прикладные функции

Как и на уровнях сеансов и представления, функции и программы прикладного уровня в распределенных АСУ имеют лишь ограниченную специфику и в основном сходны с функциями и программами АСУ традиционной архитектуры. Тем не менее обновление аппаратной базы АСУ создание новых комплексов программно-аппаратурных средств оказали известное влияние на состав и характер прикладных программ, реализуемых в распределенных АСУ

Прикладные функции рассмотрим, следуя классификации, данной.

Функции класса «объект-объект» - это автоматическое регулирование объекта и логическое управление им. Основной операцией в этих прикладных функциях является выработка управляющего воздействия в зависимости от задания и состояния объекта. Локальные станции вырабатывают управляющее воздействие программным путем в цифровой форме. В набор законов регулирования (вычисления управляющего воздействия) обычно входит непрерывное регулирование - пропорциональное (П), пропорционально-интегральное (ПИ) и пропорционально - дифференциальное (ПИД), регулирование с компенсацией запаздывания в объекте, с ограничением управляющего воздействия, по производной, многомерное регулирование. Предусматривается возможность фиксированной уставки, ее изменения как заранее заданной функции времени, каскадного регулирования, регулирования отношения между двумя технологическими переменными (одна переменная - независимая, другая - функция первой). Кроме того, имеются программы операций для двух- и трехпозиционного регулирования. В некоторых комплексах сложные законы регулирования - многомерного, каскадного и т. п. - нельзя реализовать в локальной станции, не привлекая к участию в программе координирующую станцию. Программы вычисления управляющего воздействия хранятся в локальной технологической станции чаще всего в виде модулей в постоянной памяти.

Операции, необходимые для непрерывного управления по возмущению, экстремального и адаптивного управления, можно реализовать только на координирующих станциях, допускающих свободное программирование для каждого объекта.

Логическое управление требует выполнения операции получения некоторых логических функций от набора двоичных аргументов на входе. Для прикладных функций этой группы может потребоваться также счет событий, отсчет времени, преобразование кодов.

К прикладным функциям класса «объект - оператор» технологических переменных, с вычислением функций от технологических переменных относятся также функции контроля - сигнализация о переходе переменной или ее производной за установленные границы и о других событиях, определяемых в виде логических функций двоичных переменных.

Кроме того, принято, чтобы сообщения о существующих нарушениях всегда выводились на экран дисплея независимо от вызванного в данный момент оператором кадра, обычно на красном фоне. Если нарушений в данный момент существует больше, чем помещается в выделенном для этого поле экрана, то показываются самые ранние нарушения, и с каждым нажатием клавиши самой ранней клавиши самое раннее нарушение убирается с экрана, последующие сдвигаются и вводится сообщение о первом из не уместившихся на экране нарушений.

Для непрерывного измерения важнейших переменных выделяются цифровые и (или) аналоговые индикаторы.

Функции измерения и контроля по вызову оператора в современных распределенных АСУ чрезвычайно многообразны. Более простые способы получения сведений о состоянии процесса - это пользование цифровыми и аналоговыми индикаторами и самописцами.

Вызов переменной на индикацию или запись требует, чтобы оператор установил соответствие между вызываемой переменной и прибором; в большей части систем это соответствие может быть произвольным, т. е. на любой прибор можно вызвать любую переменную. Чаще всего за каждым прибором закрепляется клавиатура вызова, на которой оператор набирает вызываемую переменную.

Функциональные клавиши выделяются только для самых важных контуров, остальные контуры вызываются алфавитно-цифровой клавиатурой, на которой оператор набирает мнемонический код контура (например, T12 - температура в 12-м контуре).

Самописцы, предназначенные для получения кривых, описывающих ход интересующих оператора технологических переменных, обычно многоточечные и многоцветные. Для выбора пера и цвета они снабжаются соответствующими функциональными клавишами.

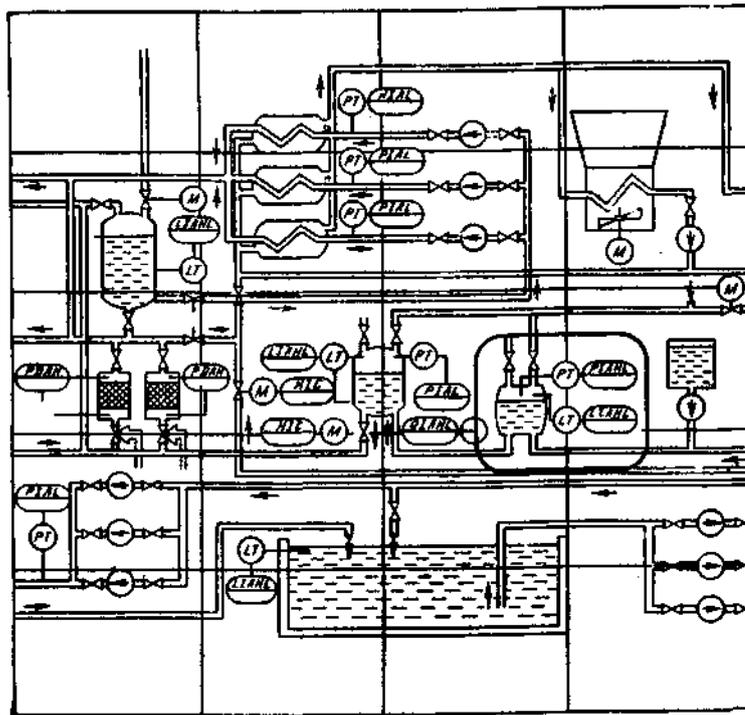
Гораздо более широкие и разнообразные возможности по представлению процесса оператору дают дисплеи с электронно-лучевыми трубками, особенно цветными. В современных АСУ принято иерархическое представление информации. Верхний уровень иерархии занимает обзорная картина, на которой виден весь процесс но без деталей, и охватывающая 150 - 250 контуров регулирования; нижний уровень - детальное представление данных по нескольким или даже одному контуру регулирования.

Обзорный кадр дает оператору примерно ту же информацию, какую он получает с традиционного щита с приборами, глядя на него с расстояния 4 - 5 м; кадр промежуточного уровня - групповой, содержащий данные по шести контурам, - дает информацию, получаемую с расстояния около 1 м от щита.

Кадры с графиками изменения переменных формируются путем нажатия специальных клавиш в сочетании с клавишами вызова переменных. При одновременном вызове на экран нескольких кривых

их различают цветом. Иногда предусматривают возможность выбора масштаба по оси времени, т. е. отрезка, к которому относятся графики, и введения фильтрации для исключения случайных колебаний.

Нестандартные кадры, формируемые специально для каждого процесса, широко используют графические возможности дисплея и позволяют представить на экране цветные мнемосхемы процесса в целом и отдельных его частей. Графические изображения обычно имеют две иерархические ступени. Обзорная картина, охватывающая весь процесс разделена на фрагменты (например,  $4 \times 4 = 16$  фрагментов, как показано на рис.41). При возникновении любого нарушения сообщение о нем появляется в верхней части кадра. Затем оператор может обратиться к кадру второй иерархической ступени - фрагменту, вызвав его в увеличенном виде на весь экран. На фрагменте мнемосхемы рядом с условным изображением аппарата или установки указываются значения переменных с применением цветовых выделений. Часть кадра может быть занята графиком изменения избранной переменной, вызванным из архива.



**Рис. 41. Кадр обзора процесса на дисплее, разделенный на 16 фрагментов**

Для вызова кадра применяют три способа: нажатие клавиш, подведение маркера (специальной метки на экране дисплея) к имени переменной или прикосновение световым пером к интересующему оператора фрагменту обзорного кадра или же к имени переменной в таблице нарушений на экране дисплея.

Имеются специальные клавиши, нажатием которых можно листать фрагменты, поочередно их вызывая на экран в прямом или в обратном порядке.

Прикладные функции класса «оператор - объект» - воздействия на процесс со стороны оператора - имеют много общего с функциями класса «объект - оператор», поскольку здесь также нужно выбирать величину, относящуюся к контуру регулирования или объекту управления, но в целях воздействия на величину.

Прикладные функции класса объект - архив (накопления архива) включаются либо автоматически - в функции времени или событий в процесс либо по вызову оператора. В последнем случае для вызова используется способ, аналогичный применяемому для вызова измерения. При фиксации «аварийного архива» особое внимание уделяется обеспечению приема сигналов с интенсивностью, многократно превышающей нормальный режим.

При регистрации в зависимости от выбранного регистрационного протокола фиксируют дату, время, мнемоническое имя переменной, ее полное название, измеренное значение со знаком и названием единиц измерения, состояние переменной (в нормальной области или вне ее), граничные значения, знак производной, моменты наступления и окончания

нарушения и т. д. Форматы регистрации устанавливаются при проектировании АСУ ФИ либо выбором из нескольких стандартных, т. е. заранее запрограммированных, форматов, либо программированием.

Требования, предъявляемые к прикладным функциям АСУ надежность, точность, допустимое запаздывание исполнения. Понятие точности относится только к функциям с выходной величиной, измеряемой числом; при выходе в форме логического суждения следует говорить о его достоверности. Связь показателей, характеризующих эффективность исполнения отдельных прикладных функций, с эффективностью АСУ (Щ) в целом рассмотрена далее.

### Контрольные вопросы

1. Что представляет собой сеанс? [Л.7, 107-109].
2. Расскажите о всех разнообразиях известных алгоритмов синхронизации. [Л.7, 109-110].
3. На какие группы делятся прикладные процессы для упрощения сеансов обмена в распределенных АСУ ф^? [Л.7, 111-114].
4. От чего зависят способы программной реализации функций представления данных? [Л.7, 114-115].
5. Особенности реализации систем управления данными в распределенных АСУ (!)ф [Л.7, 116-117].
6. Проиллюстрируйте принцип изображения данных в различных моделях. [Л.7, 117-118].
7. Приведите пример для реляционной базы данных. [Л.7, 117-119].
8. Объясните о системах управления базами данных (СУБД). [Л.7, 119-120].
9. Прикладные функции классов «объект-объект» и «объект-оператор». [Л.7, 120-123].
10. Способы получения сведений о состоянии процесса. [Л.7, 124-127].

## Лекция №8

### Операционные системы

#### План

1. Место операционной системы в АСУ.
2. Процессоры и запоминающие устройства (ЗУ).
3. Типичные функции операционных систем

#### **1. Место операционной системы в АСУ**

Под операционной системой (ОС) понимают совокупность компонентов общего программного обеспечения, служащих для эффективного и упорядоченного выделения ресурсов отдельным пользователям программно-аппаратурной системы (рис.42). Ресурсом является любой компонент внутри или вне вычислительной системы, применяемый несколькими пользователями поочередно. В понятие ресурса включают помимо физических устройств (процессоров, памяти, устройств ввода-вывода, средств передачи данных и т. п.) также и логические компоненты (отдельные данные, файлы, программы, в частности трансляторы, редакторы, системы управления файлами и базами данных, поскольку эти компоненты могут поочередно применяться несколькими пользователями). В роли основных пользователей ресурсов выступают прикладные процессы, управляющие технологическим оборудованием в реальном масштабе времени. В рамках ОС процесс определяют как совокупность операций, выполняемых строго последовательно. Операции задаются в форме программ.



Рис.42. Место операционной системы в АСУ

Сама процедура выделения ресурса может выглядеть как запуск подпроцесса по требованию прикладного процесса, открытия доступа к данным, необходимым прикладному процессу передачи данных по каналу связи с объектом.

Характерной чертой распределенной АСУ является большое разнообразие ОС, применяемых в программно-аппаратурных комплексах. Оно объясняется прежде всего различиями в проблемной ориентации станций, что отражается на составе управляемых ресурсов. Большинство станций распределенных АСУ не содержит стандартных устройств ввода-вывода и внешней памяти, поскольку прикладные задачи решаются на распределенной сети процессоров, выполняющих ОЗУ-резидентные программы. В связи с этим упрощается, например, управление вводом-выводом, отпадает необходимость в организации выполнения диск-резидентных программ и т.п. Универсальные ОС, настраиваемые на заданную конфигурацию аппаратуры, не нашли широкого применения в комплексах для построения распределенных АСУ из-за чрезмерной избыточности. В связи с этим возникла настоятельная необходимость в обобщении индивидуальных реализаций.

Работа по унификации ОС мультипроцессорных программно-аппаратурных комплексов ведется в рамках комитета ТС8 европейской рабочей группы по промышленным вычислительным системам EWICS (European Workshop on Industrial Computer Systems) .

## 2. Процессоры и запоминающие устройства (ЗУ)

Главными аппаратными ресурсами являются процессоры и ЗУ. Поэтому определяющим фактором при построении ОС оказывается выбор архитектуры памяти и способа распределения прикладных программ между процессорами.

В распределенном программно-аппаратурном комплексе АСУ обычно присутствует глобальная память, располагающаяся в центральной операторской станции и обслуживающая всю систему. В ней размещаются данные и программы, необходимые в той или иной форме всем локальным технологическим станциям (например, для целей конфигурирования, рестарта и т. п.). Обычно глобальная память имеет большую емкость и из экономических соображений располагается на дисках, так что время доступа к ней сравнительно велико.

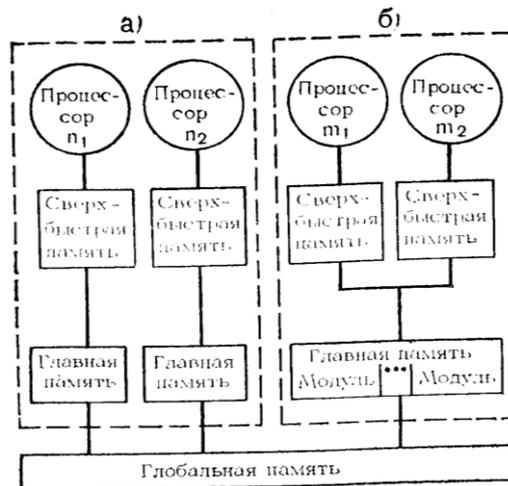
Поскольку глобальная память может удовлетворить лишь несколько десятков запросов в секунду, а процессор способен обратиться к ЗУ несколько миллионов раз в секунду, возникает противоречие, которое снимается введением более быстрых промежуточных ЗУ. К ним относится главная (рабочая) память, базирующаяся на блоках оперативной, постоянной или полупостоянной памяти, в которых хранятся данные и программы, требуемые процессорам. Полупроводниковые ЗУ обеспечивают обслуживание до 2 - 3 млн. обращений в секунду. Если этого недостаточно, вводится сверхбыстродействующая память ограниченной емкости, конструктивно объединяемая с процессором. Совместно с регистрами процессора эти виды ЗУ образуют вертикальную иерархию памяти. В системах со многими параллельно работающими процессорами на каждом уровне возможно дополнительное расчленение памяти по горизонтали на локальные элементы. Такими элементами являются ЗУ, принадлежащие одному процессору. Регистры, естественно, всегда локальны, глобальная память, наоборот, всегда общая. Для сверхбыстрой и главной памяти возможно несколько схем. Объединение блоков сверхбыстрой памяти целесообразно при числе процессоров не более четырех, иначе снижается быстродействие. Для общей главной памяти и локальных сверхбыстрых блоков это число повышается до восьми (рис.43,б). Системы с общей главной памятью называют жестко связанными мультипроцессорными или просто мультипроцессорными системами, поскольку доступ к памяти здесь возможен только по фиксированным маршрутам.

Если процессоры располагают локальной главной памятью, система называется *многочастиной*, или *свободно связанной* (рис.43,а). Выбор пути коммуникации в такой системе осуществляется в зависимости от состояния аппаратуры.

Возможно также применение комбинированной схемы

распределения памяти. Система строится в целом как многомашинная, но содержит подсистемы с жесткими связями, в которых доступ к общей главной памяти координируется сетью связи (рис.43,а и б).

Наличие общей главной памяти существенно упрощает реализацию ОС распределенной программно-аппаратурной конфигурации, однако мощности восьми процессоров, которые может объединить такая схема, в общем случае недостаточно для построения распределенной АСУ. Это ограничение снимается применением ЗУ, имеющих дополнительный вход и допускающих независимое обращение к одной памяти двух процессоров одновременно. В этом случае в комплексе можно выделить (статически или динамически) привилегированный процессор, имеющий прямой доступ к локальным ЗУ, подчиненным другим процессорам, в обход этих процессоров. При необходимости подчиненный процессор сам анализирует, не поступили ли новые данные в его локальную память и представляют ли они для него интерес.



**Рис.43. Схемы расположения АСУ:**

**а – многомашинная (свободно связанная) подсистема;**

**б – мультипроцессоры (жестко связанная) подсистема**

При статическом распределении каждый процессор получает в точности одно задание из общего алгоритма управления и только он один ответствен за его выполнение. Процессор высшего уровня инициирует выполнение заданий, передавая подчиненному процессору исходные данные и принимая от него результаты, контролирует ресурсы и поддерживает связь с объектом.

При динамическом распределении все процессоры имеют доступ к устройству связи с объектом управления и работают параллельно. Процессора высшего уровня не существует; каждый свободный процессор может выполнять очередную программу.

В распределенной АСУ локальные технологические станции обычно жестко связаны через датчики и исполнительные органы с

отдельными, как правило, удаленными друг от друга частями объекта управления, которые являются источниками и потребителями данных. Поэтому большинство программ обработки данных измерения и выдачи управляющих воздействий жестко закреплено за технологическими станциями. Точно так же программы взаимодействия с оператором-технологом закреплены за операторскими станциями. Таким образом, в распределенной АСУ преобладает статическое распределение программ. Однако принцип статического распределения затрудняет использование одного из главных преимуществ многопроцессорного комплекса - возможности реконфигурации программ и аппаратуры при отказах и нарушениях, поэтому иногда применяются компромиссные варианты.

В архитектуре открытых систем функции ОС определены как административные. Они решают задачи распределения ресурсов и выдачи заданий на управление ресурсами. В отличие от терминологии, административные процессы являются частью прикладных, будем различать прикладные программы и ОС. Поскольку ресурсы программно-аппаратурной системы (процессоры, память, данные, обслуживающие программы) распределены по уровням архитектуры, компоненты ОС также присутствуют на каждом уровне, хотя прямое соответствие между функциями операционной системы и функциональными уровнями распределенной АСУ установить трудно, так как архитектура открытых систем отражает прежде всего структуру связи компонентов, а архитектура ОС - структуру управления компонентами (ресурсами). Представляется целесообразным использовать для описания функций ОС принципы, положенные в основу архитектуры открытых систем.

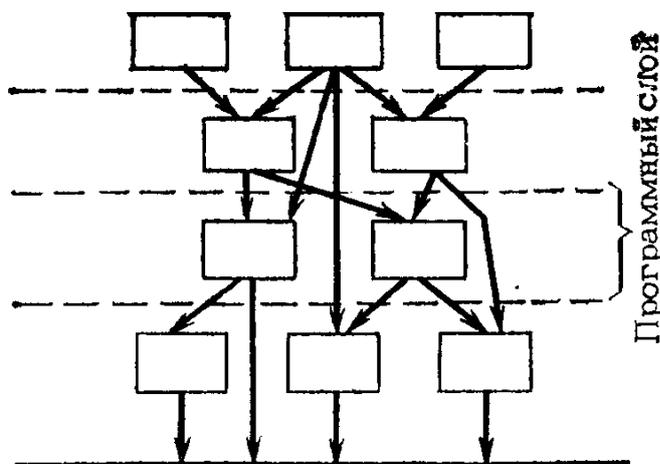
Предлагаемое далее описание уровней ОС распределенных АСУ опирается на единичные работы, посвященные послойному проектированию операционных систем, и поэтому должно рассматриваться лишь как попытка систематизации функций ОС.

Программные компоненты могут быть как частями прикладных программ, написанных пользователем, так и частями ОС, причем доля компонентов ОС увеличивается от верхних уровней к нижним. Сообразно этому прикладной процесс разбивается на подпроцессы, выполняемые с использованием услуг нижележащего уровня, которые в свою очередь состоят из подчиненных подпроцессов, и т. д. Результаты выполнения каждого подпроцесса передаются вызывающему процессу и в конечном итоге оказываются на прикладном уровне. Иерархия параллельных и последовательных процессов составляет логическую структуру АСУ. Процессы являются абстрактными объектами, которые реализуются физически с помощью аппаратуры, данных и программ (ресурсов). В распределенных АСУ один и тот же прикладной процесс может быть реализован различными ресурсами.

### **3. Операционная система организует**

Операционная система организует выполнение прикладных процессов путем выделения ресурсов. В зависимости от видов ресурсов и возможностей управления ими возникает ряд типичных функций ОС. Для их представления воспользуемся методом послойного описания. Его можно свести к следующим трем принципам:

1. Слои выделяются по функциональному признаку, т. е. независимо от конкретной реализации ОС.
2. В отдельный слой объединяются программные компоненты, не связанные друг с другом по управлению, но объединенные в силу первого принципа общим целевым назначением (рис.44). Отсюда вытекает, что компоненты одного уровня не обращаются непосредственно друг к другу.

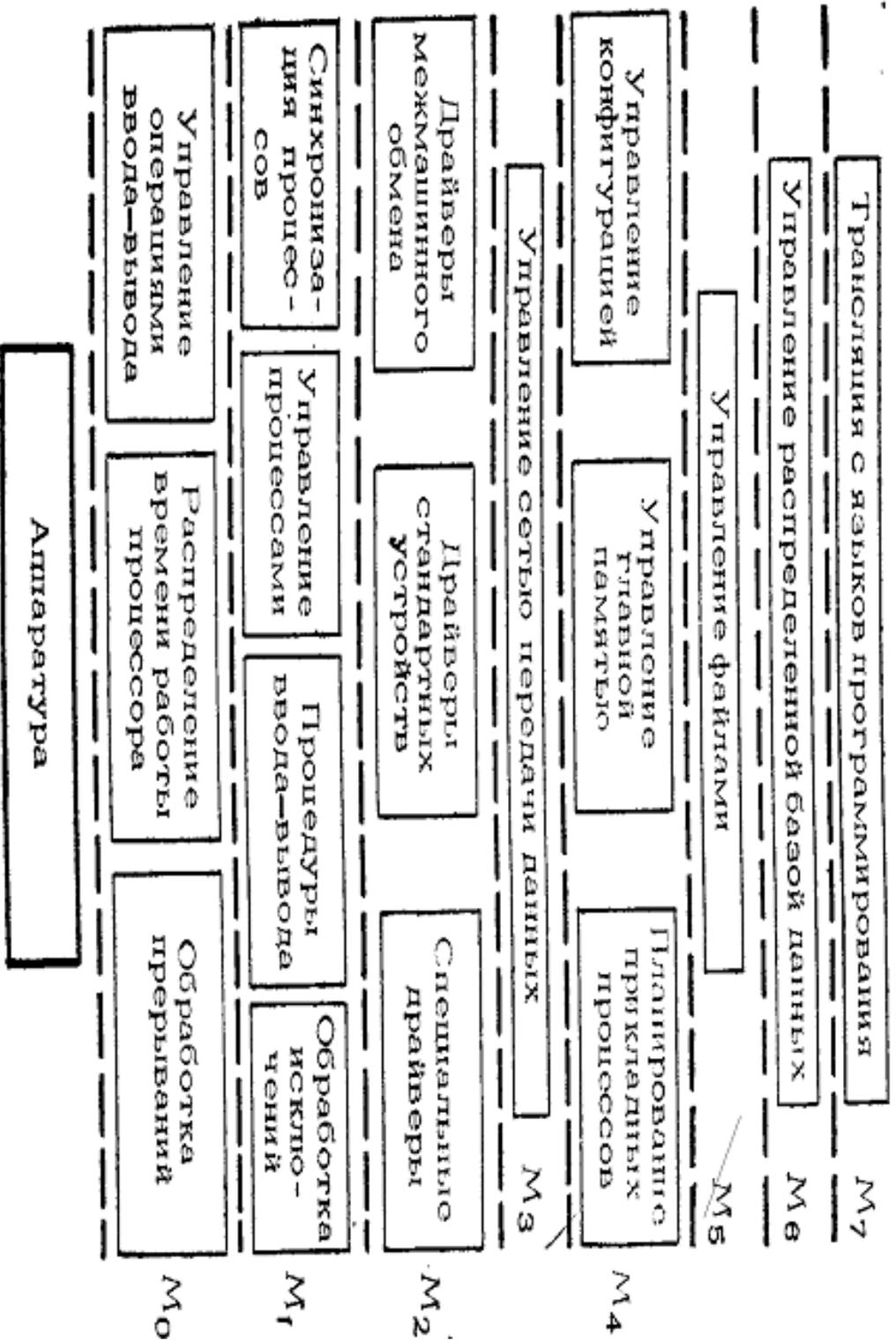


**Рис.44. Иерархически структурированная программная система**

3. Каждому компоненту, доступному из другого (верхнего) слоя, ставится в соответствие некоторая псевдо-команда, характеризующая выполняемую им функцию. Это позволяет рассматривать каждый слой как псевдомашину с определенным набором команд.

На рис. 45 представлен один из возможных вариантов многослойной функциональной структуры ОС.

Задача машины  $M_0$  состоит в распределении времени работы физических устройств между запросами, поступающими с верхних уровней. Распределение осуществляется путем блокирования прерываний, т. е. других запросов (при обращении к процессору), или запуска операций (при обращении к периферийной аппаратуре) с фиксацией момента их завершения. Машина  $M_0$  захватывает ресурс жестко, т. е. получение времени физического устройства возможно только через ее функции. Назначение машин  $M_1$  состоит в организации параллельного выполнения взаимодействующих процессов. Взаимодействие параллельных процессов базируется на обмене информацией.



Рис