

В.В. Елисеев, В.А. Ларгин, Г.Ю. Пивоваров

ПРОГРАММНО-ТЕХНИЧЕСКИЕ КОМПЛЕКСЫ АСУ ТП

Киев 2003

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

АРМ – автоматизированное рабочее место;
АСУ ТП - автоматизированная система управления технологическим процессом;
АЦП - аналого-цифровой преобразователь;
АЭС - атомная электростанция;
БД – база данных;
БПВр-1 - блок приема сигналов времени из сети единого времени;
БСО - блок связи с объектом;
ВВЭР-1000 – водо-водяной энергетический реактор мощностью 1000 MW;
ИБ - информационная база;
ИВМ - информационная вычислительная машина;
ИВК - информационно-вычислительный комплекс;
ИВС - информационно-вычислительная система;
ИР - интерфейс резервированный;
ИС МСКУ - исполнительная система микропроцессорного субкомплекса контроля и управления;
ИСПП - интегрированная система подготовки программ;
КМп - контроллер микропроцессорный;
КСв - контроллер связи;
КСО – комплекс связи с объектом;
КТС – комплекс технических средств;
ЛВС - локальная вычислительная сеть;
МАПС - модульная асинхронная перестраиваемая сеть;
МСКУ М - микропроцессорная система контроля и управления;
НЦУ - непосредственное цифровое управление;
ОБД - оперативная база данных;
ОС - операционная система;
ОС РВ - операционная система реального времени;
ПЛК – программируемый контроллер;
ПО - программное обеспечение;
ПСд - панель соединительная;
ПТК - программно-технический комплекс;
РВ - реальное время;
РМО – рабочее место оператора;
САУ - система автоматического управления;
САУ ГПА - система автоматического управления газоперекачивающим агрегатом;
СВРК-М - система внутриреакторного контроля модернизированная;
ТП – технологический процесс;
ТЭО - технико-экономическое обоснование;
ТЭС – тепловая электростанция;
ТОУ - технологический объект управления;
ТЭП – технико-экономические показатели;
УВК - управляющий вычислительный комплекс;
УВМ - управляющая вычислительная машина;
УВС - управляющая вычислительная система;
УСО - устройство связи с объектом;
УС МСКУ - управляющая система МСКУ;
ФСО - функционально-структурная организация;
ЦЩУ - центральный щит управления;
ЦАП – цифро-аналоговый преобразователь;
ЭВМ – электронно-вычислительная машина.

ВВЕДЕНИЕ

Управление предполагает наличие управляемого объекта или группы объектов и органа управления, который воздействует на объект, изменяя его состояние в нужном направлении. Управление представляет собой набор воздействий, направленных на поддержание или улучшение функционирования управляемого объекта в соответствии с заданной целью управления. Управление должно быть оптимальным, т.е. осуществляться наилучшим образом.

Оптимальное управление заключается в выборе наилучших по некоторому критерию эффективности управляющих воздействий из множества возможных с учетом имеющихся ограничений и информации о состоянии управляемого объекта и внешней среды. Основным инструментом для решения проблем управления производством служат автоматизированные системы управления (АСУ). АСУ – это человеко-машинная система, обеспечивающая автоматизированный сбор и обработку информации, необходимой для оптимизации управления в различных сферах человеческой деятельности. По типу объектов управления различают АСУ предприятием – АСУП и АСУ технологическими процессами – АСУ ТП [1].

АСУ ТП – это АСУ для выработки и реализации управляющих воздействий на технологический объект управления в соответствии с принятым критерием управления. В АСУ ТП важную роль играет человек, который принимает решения по управлению технологическим объектом. Операции по сбору и обработке информации выполняются автоматическими устройствами. Целью функционирования АСУ ТП является оптимизация работы объекта путем соответствующего выбора управляющих воздействий. В АСУ ТП выработка решений по управлению и воздействие на объект выполняются в том же темпе, что и протекающие технологические процессы. Такой режим работы АСУ ТП называется *режимом реального времени*. В АСУ ТП важное значение имеют процессы сбора измерительной информации, ее оперативного отображения и выдачи управляющих воздействий на исполнительные средства ТОУ.

Технологический объект управления представляет собой совокупность технологического оборудования и реализуемого на нем технологического процесса производства. В качестве ТОУ могут рассматриваться технологические агрегаты и установки; автономные производства, имеющие законченный цикл; производственный процесс всего промышленного предприятия, если управление заключается в выборе и согласовании режимов работы взаимосвязанных агрегатов, участков и производств. Примерами ТОУ в энергетике могут быть: энергоблоки различной мощности тепловых (ТЭС) и атомных (АЭС) электростанций, генераторы энергоблоков ТЭС и АЭС, турбины, котлы; в нефтехимии и химии: нагревательные печи, производство кокса, производства химической продукции; в газовой промышленности: газоперекачивающие агрегаты, компрессорные станции и цеха, газотранспортные предприятия; в металлургической промышленности: доменные печи, прокатные станы металлургических заводов и другие.

Управление технологическим процессом (ТП) это - управление режимами работы технологического оборудования. Под термином "*управляемый ТП*" понимают процесс, для которого: определены входные воздействия, установлены зависимости между входными воздействиями и выходными параметрами объекта, реализованы автоматические измерения входных воздействий, выходных параметров и управление процессом.

Задачу, выполняемую в системе ТП - АСУ ТП, можно сформулировать следующим образом. По полученным данным о ТП составить прогноз его хода, а также составить и реализовать такой план управляющих воздействий, чтобы в определенный момент времени состояние ТП отвечало экстремальному значению обобщенного критерия качества процесса [2]. Для решения этой задачи необходимо иметь математическую модель процесса.

Принцип действия АСУ ТП представлен на рис. В.1. Текущая измерительная информация о состоянии технологического процесса, протекающего в управляемом объекте, поступает в управляющую систему. Далее она контролируется и сравнивается с моделью объекта. Результаты сравнения анализируются, после чего готовятся и принимаются решения по управлению.

Управляющая система вместе с объектом управления образуют *автоматизированный технологический комплекс*. Человек является элементом рассматриваемой системы управления.

Если человек участвует в процессе принятия решения по управлению, то такую систему называют *автоматизированной системой управления*. Систему, в которой человек не участвует в процессе принятия решений, называют *системой автоматического управления* (САУ).

В зависимости от функционального назначения системы управления задачи управления могут формулироваться по-разному. Наиболее простой является задача стабилизации управляемой величины, когда требуется с заданной точностью поддерживать постоянными те или иные параметры технологического процесса. В качестве примера можно привести задачу поддержания температуры в помещении (стабилизируемый параметр), изменяющейся из-за нестабильности температуры окружающей среды (возмущающий фактор), посредством изменения расхода циркулирующего в теплоцентраль теплоносителя (управляющее воздействие). Функция человека в такой АСУ ТП сводится к заданию номинальной величины стабилизируемого параметра или к изменению расхода теплоносителя принудительно (дистанционное управление).

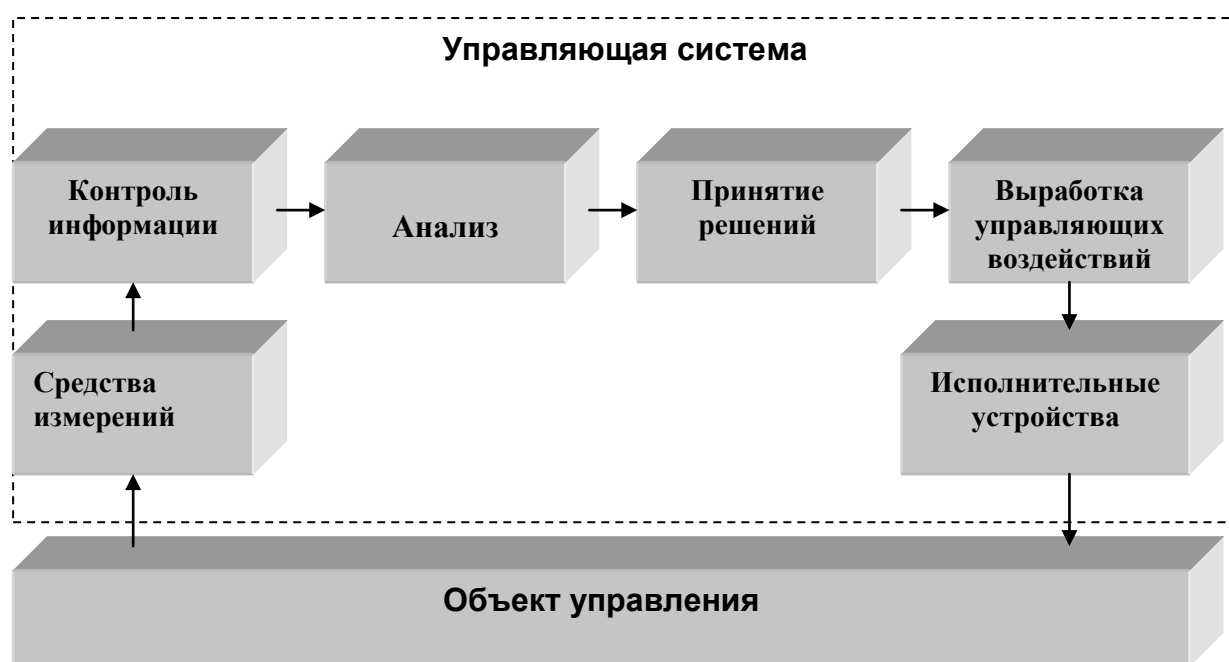


Рис. В.1. Принцип действия АСУ ТП.

В более сложных ситуациях человек, исследовав с помощью средств визуализации АСУ ТП тренды (графики изменения параметров во времени), может изменить закон регулирования регулятора с целью повышения качества (уменьшения нестабильности контролируемого параметра) или экономической эффективности (уменьшения интегрального расхода теплоносителя с целью уменьшения платы за энергоресурсы). Более сложной является задача программного управления. В этом случае управляемая величина должна изменяться по заданному закону (программное регулирование). Для рассматриваемого примера – это может быть автоматическое изменение уставки (граничного значения) контролируемого параметра во времени (ночью, в отсутствие людей, в помещении может быть прохладнее, днем – теплее и т.п.).

К наиболее сложным и совершенным относятся адаптивные системы управления. В них управляющие воздействия, или алгоритмы управления, изменяются автоматически и целенаправленно для обеспечения лучшего управления объектом. При этом характеристики объекта управления или воздействия внешней среды могут изменяться по заранее неизвестным законам. Пусть, например, температура теплоносителя в теплоцентраль подвержена существенным изменениям, а процесс теплопередачи в теплообменниках (радиаторах) нелинейно зависит от температуры. Тогда для обеспечения заданного качества регулирования настроечные параметры адаптивного регулятора должны также изменяться по определенному алгоритму с целью до-

стижения наилучшего качества (уменьшения времени переходного процесса, числа переключений и ошибки рассогласования).

Качество регулирования находится в прямой зависимости от количества измеряемых возмущающих факторов и от того, насколько полно они учитываются в законе регулирования. В рассматриваемом примере, кроме собственно стабилизируемого параметра (температуры) в помещении могли бы быть известны такие возмущающие факторы, как: температура окружающей среды, теплоносителя, степень открытия или закрытия вентиляционных отверстий, температура холодного воздуха, втекающего из системы вентиляции и его объем и другие, например, самочувствие человека в помещении или насколько он тепло одет. При этом можно было бы оптимизировать работу АСУ ТП по критерию максимума комфортности для человека. Если приложить достаточные усилия (иногда невероятно), можно было бы построить сравнительно точную математическую модель такого объекта. Скорее всего, она была бы нелинейной. К трудностям ее построения добавилась бы сложность реализации вычислительного алгоритма из-за чрезмерных требований к вычислительной мощности решающего устройства регулятора. Разрешить такую проблему могут системы управления с алгоритмами «нечеткой логики» [3].

Необходимые динамические характеристики системы управления определяются скоростью изменения состояния ТОУ. Понятие реального масштаба времени, характеризующее особенности управляющей системы, часто распространяют на систему управления в целом. Понятие "АСУ работает в масштабе реального времени" определяет следующие особенности АСУ: чрезвычайно малое время, отведенное для принятия решения (в 10 – 20 раз меньше постоянной времени переходного процесса ТОУ); практически мгновенное использование результатов решения для управления; недопустимость для некоторых объектов, как преждевременной выдачи управляющих сигналов, так и их запаздывание, т.е. привязка к точному астрономическому времени.

При проектировании АСУ важным показателем является точность управления. Этот показатель оценивается как степень отличия величины управляющих воздействий от некоторых значений, признанных идеальными в рассматриваемые моменты времени. Точность управления характеризует степень отклонения фактических моментов выдачи управляющих воздействий от оптимальных или заданных значений. С повышением точности усложняется алгоритм, повышаются требования к быстродействию системы. Точность учитывается как параметр, влияющий на выбор технических решений при проектировании АСУ ТП. Нарушение требований к точности может привести к авариям объекта управления и поэтому расценивается как отказ системы.

Для реализации задач управления с выполнением необходимых требований (к качеству регулирования, динамике, точности и пр.) необходимы соответствующие технические средства. Например, для надежной и экономичной работы энергоблоков тепловых электростанций и АЭС, необходимо контролировать и поддерживать в достаточно узких пределах с требуемой точностью тысячи технологических параметров, воздействовать на сотни исполнительных механизмов [4]. При использовании обычных средств (применявшихся 30 – 40 лет назад) для контроля работы одного энергоблока ТЭС требовалось несколько сот измерительных, десятки самопишущих приборов, около тысячи сигнальных ламп и т.д. Естественно, что человек-оператор в таких условиях не в состоянии ясно представить ход технологического процесса, а тем более и оптимально им управлять.

Средства вычислительной техники, применяемые в АСУ ТП, могут компоноваться в различные вычислительные системы с различным функциональным назначением. В технической литературе и на практике используются несколько терминов для обозначения таких систем: УВМ, ИВМ, УВК, ИВК, УВС, ИВС, ПТК и т.д.

ЭВМ, используемые для контроля и управления производственными процессами (в том числе и для непосредственного цифрового управления), относятся к классу управляющих вычислительных машин (УВМ), на базе которых строятся управляющие вычислительные комплексы технических и программных средств (УВК). ЭВМ, которые используются в основном для сбора, обработки, контроля и предоставления информации оператору, относятся к классу

информационных вычислительных машин (ИВМ), на базе которых строятся информационно-вычислительные комплексы технических и программных средств (ИВК). Термины УВС (управляющая вычислительная система) и ИВС (информационно-вычислительная система) практически тождественны терминам УВК и ИВК соответственно.

Ранние УВК и ИВК, как правило, строились на базе одной машины и имели централизованную структуру. Машина должна была обладать высокой степенью надежности, так как при ее неполадках могла выйти из строя практически вся АСУ ТП. Поэтому наиболее ответственные функции контроля и управления резервировались за счет соответствующих автономных средств. В последнее десятилетие в связи с широким производством микропроцессорной техники стала возможной реализация УВК и ИВК с распределенной структурой, включающей локальные подсистемы. Каждая локальная подсистема может быть реализована на базе промышленных контроллеров (специализированных мини-ЭВМ) или ЭВМ. Все подсистемы объединяются в единую систему с помощью общей сети передачи данных и сетевого программного обеспечения. Надёжность технической структуры АСУ ТП обеспечивается за счёт резервирования подсистем и сети передачи данных.

В последнее время все большее распространение получил термин ПТК (программно-технический комплекс) АСУ ТП. ПТК представляет собой совокупность средств измерительной и вычислительной техники, программного обеспечения для выполнения функций АСУ ТП. Термин ПТК является более широким понятием, чем УВК и ИВК. Поэтому дальнейшее изложение (если специально не оговорено) будет вестись с использованием термина ПТК, имея в виду ПТК, реализующий функции АСУ ТП.

При использовании ПТК в управлении исключается большое число показывающих и самопишущих приборов. ПТК собирает и обрабатывает информацию о состоянии оборудования, выдает оператору результаты в виде предупредительной и аварийной сигнализации. Оператор имеет возможность вызывать на экран дисплея значения интересующих его параметров. При таком способе управления можно не устанавливать громоздкий щит управления и освободить оператора от непрерывного наблюдения за многими технологическими параметрами.

Первые применявшиеся в управлении технологическими процессами ЭВМ имели малую производительность и низкую надежность, неразвитое программное обеспечение. Но уже тогда они могли выполнять следующие функции.

1. *Периодический опрос контролируемых параметров, датчики которых подключены к входам машины.* Помимо поочередного опроса параметров, возможен более частый опрос тех из них, значения сигналов которых постоянно или временно необходимы для наблюдения или использования в расчетах.

2. *Аварийная сигнализация.* Машина постоянно сравнивает величины опрашиваемых входных параметров с заданными программой уставками, т.е. допустимыми аварийными пределами. Если параметр выходит за эти пределы, то включается световой или звуковой сигналы, а также фиксируются во внешней памяти время отклонения, название или номер данного параметра и его значение. В программе может быть предусмотрено автоматическое изменение уставок в зависимости от режимов работы технологического оборудования или других факторов.

3. *Автоматическая регистрация отклонений.* Машина автоматически регистрирует в цифровой форме и печатает значения определенного количества параметров по выбору оператора. При этом могут печататься как абсолютные величины параметров, так и значения их отклонений.

4. *Периодическая автоматическая регистрация.* Машина через заданные интервалы времени (30 минут, 1 час, 8 часов и т.д.) автоматически регистрирует все основные параметры технологического процесса, а также рассчитываемые величины технико-экономических показателей работы оборудования. В программе вычислительной машины могут быть также предусмотрены суточные, месячные и другие циклы автоматической регистрации, во время которых отпечатываются средние и просуммированные значения ряда параметров за отчетные периоды.

5. *Расчет технико-экономических показателей.* Вычислительные машины рассчитывают технико-экономические показатели работы объекта. Результаты этих расчетов могут быть ис-

пользованы в оперативных целях для оптимизации работы оборудования (за интервалы времени 10, 15, 30 минут и более), а также для учета (за смену, сутки, месяц и т.п.).

После появления ЭВМ третьего поколения (конец 60-х, начало 70-х годов прошлого века) значительно расширились возможности их применения в АСУ ТП. Переход к интегральной технологии позволил резко улучшить такие характеристики управляющих ЭВМ, как: надежность, быстродействие, габариты, потребляемая мощность, простота обслуживания и т.д. Программное обеспечение приобрело четкую иерархическую структуру. Появились многозадачные операционные системы (ОС). ОС включает планировщик задач, подсистему обработки внешних прерываний, драйверы внешних устройств, службу времени и др. Прикладные программы выполняются в виде задач, которые логически разделены между собой. Появились специализированные и универсальные языки программирования высокого уровня. Все это позволило поручить ЭВМ не только разнообразные информационные и вычислительные функции, но и функции управления, включая непосредственное цифровое регулирование технологических параметров.

Эволюцию ПТК можно проследить на примере создания средств вычислительной техники для управления технологическими процессами в Северодонецком НПО «Импульс». В 1963 г. были разработаны образцы первых серийных управляющих вычислительных машин - УМ-1 и МППИ-1; в 1972 г. завершилась разработка мини-ЭВМ народно-хозяйственного назначения, выполненной на микроэлектронной элементной базе - М-6000; с 1975 г. дальнейшее развитие средств управляющей вычислительной техники НПО «Импульс» велось по международной программе СМ ЭВМ. Были разработаны и освоены в серийном производстве комплексы СМ-1, СМ-2 (затем модернизированный вариант - СМ-2М), СМ1210, СМ1634, ТВСО-1. Наибольшее распространение из них получили СМ-2/СМ-2М и ТВСО-1. Они эксплуатируются в составе АСУ ТП на подавляющем большинстве атомных и крупных тепловых электростанций, а также на таких предприятиях, как "Азовсталь", "Криворожсталь", Криворожский ГОК, Киевский механический завод им. Антонова, Северодонецкое ПО "Азот", на космодроме "Байконур", в Центре управления полетами, в Центре подготовки космонавтов, на Московском и Санкт-Петербургском метрополитене и многих других объектах. На базе вычислительных комплексов НПО «ИМПУЛЬС» создана общесоюзная система резервирования и продажи авиабилетов «СИРЕНА». К системам массового обслуживания, построенным на средствах НПО «Импульс», относятся региональные центры АСУ «Олимпиада-80» в Москве и Таллинне, система автоматизации скорой помощи, разработанная Киевским институтом Горсистемотехника. С 1992 года в НПО «Импульс» серийно выпускаются средства Микропроцессорной Системы Контроля и Управления - МСКУ М [6, 7].

Значительное место работам НПО "Импульс" по созданию управляющей вычислительной техники отведено в книге Б.Н. Малиновского "Очерки по истории компьютерной науки и техники в Украине" (издана в 1998г. в Киевском издательстве "Феникс"). Там приведены сведения, начинающиеся от первых специализированных машин типа "Автодиспетчер" и "Автооператор", создания технических средств систем СОУ-1, АСВТ, СМ ЭВМ, ПС и заканчивающиеся современными разработками - микропроцессорной системой контроля и управления МСКУ М.

МСКУ М ориентирована на создание многоуровневых децентрализованных ПТК в различных отраслях промышленности: энергетике (АСУ ТП и ИВС энергоблоков для ТЭС и АЭС, АСУ ТП открытых распределительных устройств, системы внутриреакторного контроля энергоблоков АЭС, системы сбора и передачи телеметрической информации); в нефтехимии и химии (АСУ ТП нагревательных печей для предприятий нефтеоргсинтеза, автоматизированные системы управления газовыми нагнетателями, АСУ ТП производства пропилена); в газовой промышленности (САУ газоперекачивающими агрегатами, АСУ ТП компрессорных станций и цехов); в металлургической промышленности (АСУ ТП прокатных станов металлургических заводов, АСУ ТП доменных печей); в пищевой промышленности (АСУ ТП сахарных заводов). На базе МСКУ М в 1992-2002 г.г. разработаны и введены в эксплуатацию АСУ ТП на Кольской и Курской АЭС, система внутриреакторного контроля на Запорожской АЭС, АСУ ТП энергоблоков Запорожской, Змиевской, Углегорской ТЭС (Украина), Киришской ТЭС (Россия), ИВС

реакторных установок на ПО «Маяк», модернизирована АСУ ТП на Балаковской АЭС, АСУ ТП ряда металлургических производств, нефтегазового комплекса и других объектов [8 - 12].

Основными составляющими МСКУ М являются микропроцессорные комплексы контроля и управления (МСКУ), рабочие станции, локальные вычислительные сети, средства локальной автоматики. Для организации нижнего уровня ПТК предназначены МСКУ, осуществляющие прием и необходимую обработку измерительной информации от объекта, управление технологическим оборудованием, регулирование параметров технологического процесса, защиту оборудования и блокировки. Каждый МСКУ представляет собой программируемый промышленный контроллер с набором блоков связи с объектом и коммуникационными средствами.

Для организации верхнего уровня ПТК предназначены промышленные рабочие станции ПС 5110. На их базе komponуются рабочие места операторов-технологов, высокопроизводительные вычислители для решения оптимизационных задач и экспертных систем, информационные серверы, инженерные станции и станции другого назначения. Для высокоскоростного обмена информацией между компонентами системы служат промышленные локальные вычислительные сети (ЛВС) с протоколами МАПС (модульная асинхронная перестраиваемая сеть - стандарт НПО «Импульс») и Ethernet. Обеспечена также работа по интерфейсам МВС, RS-485, RS-232C, ИРПС. К средствам локальной автоматики относятся исполнительные автоматы, средства управления силовым оборудованием.

Компоненты МСКУ М обеспечивают реализацию всего диапазона управления - от одно-контурного регулирования до адаптивного управления сложнейшими технологическими процессами. МСКУ М является открытой системой за счет использования стандартных интерфейсов и программных модулей. Надежность ПТК на базе МСКУ М обеспечивается комплексом технологических, аппаратных, системных и программных решений: тщательной конструктивной проработкой системы и ее компонентов; выбором надлежащей элементной базы, входным контролем комплектующих и их выбраковкой; всесторонними испытаниями блоков элементов, узлов, устройств, программно-технических комплексов; возможностью резервирования (дублирование, троирование) на уровне узлов, модулей, подсистем и шин передачи данных; резервированными фидерами электропитания, аккумуляторной поддержкой, использованием устройств бесперебойного питания; встроенными аппаратно-программными средствами диагностики, позволяющими обнаружить неисправный блок, блокировать его работу, а после его замены обеспечить автоматическое включение в рабочий режим; контролем выдаваемых на управляемый объект сигналов; безударным переходом на ручное управление исполнительными механизмами при отказе управляющей системы; длительной непрерывной приработкой комплексов при граничных значениях параметров окружающей среды, что предотвращает практически все неисправности в начале эксплуатации.

ОБЩАЯ ХАРАКТЕРИСТИКА ПТК АСУ ТП

1.1. Основные понятия и определения

Современный ПТК представляет собой совокупность средств измерительной и вычислительной техники, программного обеспечения, средств для создания и заполнения машинной информационной базы, достаточных для выполнения функций АСУ ТП.

В состав одной АСУ ТП могут входить несколько ПТК, каждый из которых функционирует автономно, но имеет средства взаимодействия с другими. Кроме ПТК в состав АСУ ТП входят измерительные преобразователи, исполнительные механизмы, щиты управления, обслуживающий персонал (рис. 1.1). Могут входить также различные автономные устройства (например, локальные регуляторы).



Рис. 1.1. Обобщенная структурная схема АСУ ТП.

В ПТК могут быть реализованы практически все функции управляющей системы: регистрация и обработка параметров технологического процесса, визуализация процесса, регулирование, защиты и блокировки, сигнализации, вычислительные операции и экспертные системы. Информация о ТООУ вводится в ПТК от средств измерений в виде сигналов тока, напряжения, время-импульсных сигналов, частоты, дискретных сигналов. Технические средства, предназначенные для формирования входных сигналов, называются *датчиками*. Входная информация содержит значения технологических переменных (температур, давлений, расходов и т.п.). На основе принятой информации вырабатываются решения по управлению объектом. В последние годы интенсивно развиваются «интеллектуальные датчики», осуществляющие частичную пред-

варительную обработку информации и передающую ее в ПТК по последовательным линиям связи.

Управляющие сигналы передаются на ТОУ. Преобразование результатов решения задачи управления в непосредственные воздействия на объект управления осуществляют исполнительные устройства. Обычно они неотделимы от объекта управления, их не относят к составу ПТК, но включают в состав управляющих систем АСУ. Они являются потребителями выходной информации ПТК. Обработку входных данных и выработку управляющего воздействия на объект осуществляют средства переработки информации и вычислительные устройства.

Сложность задач управления и необходимость выполнять управление в условиях неполной информации требует включения в контур управления *человека - оператора* или группы операторов. Оператор работает параллельно с вычислительным устройством и должен иметь возможность вмешиваться в процесс управления. Необходимая ему информация для принятия решения должна быть представлена в удобной для восприятия форме. Оператору нужны также средства для ввода управляющих воздействий в систему, отображения информации и ручного управления. Отображению подлежат не только входные данные, но и информация о результатах, полученных средствами переработки информации.

На основе применения микропроцессорной техники для управления ТП, стала возможной реализация иерархической децентрализованной распределённой структуры ПТК. Она предполагает создание территориально-распределённых по технологическим участкам локальных подсистем. Каждая локальная подсистема может быть реализована на базе серийно выпускаемых промышленных контроллеров и ЭВМ или на основе специализированной, специально спроектированной для этой задачи микропроцессорной системы.

Все подсистемы объединяются в единую систему с помощью общей сети передачи данных и сетевого программного обеспечения. Такой подход дает возможность в значительной мере повысить надежность технической структуры АСУ ТП за счет резервирования подсистем и сети передачи данных.

Возможность приближения локальных подсистем к технологическому объекту и персоналу позволяет уменьшить длину физических цепей связи с датчиками, исполнительными механизмами, средствами ручного ввода и отображения информации и т.д. Вместе с использованием малопроводных локальных сетей (с последовательной передачей данных) это способствует существенной экономии кабельной продукции. Территориальное рассредоточение позволяет наилучшим образом совместить техническую структуру систем промышленной автоматизации со сложившимися на производстве технологической и организационной структурами. При этом в каждой зоне создаются автоматизированные (человеко-машинные) или автоматические системы, взаимосвязанные и координируемые со стороны вышестоящего уровня управления.

Все множество ПТК, образующих АСУ ТП достаточно сложного объекта, условно делятся на два уровня иерархии (нижний и верхний). На нижнем иерархическом уровне выполняются задачи централизованного контроля и непосредственного цифрового управления локальным технологическим процессом. При этом реализуются информационные функции ПТК, целью которых является сбор информации с датчиков, фильтрация от помех, определение достоверных значений величин на основе показаний датчиков, преобразование информации в физические величины, обнаружение отклонения технологических параметров от установленных регламентных значений, контроль и прогнозирование предаварийных ситуаций. На нижнем уровне выполняются управляющие функции, обеспечивающие реализацию режима работы ТОУ путем формирования управляющих воздействий на его исполнительные механизмы с помощью регуляторов непосредственного цифрового управления (НЦУ). На нижнем уровне могут выполняться также расчёты неизмеряемых параметров по математическим моделям и технико-экономических показателей, создание и оперативное ведение базы данных локального технологического процесса.

На верхнем уровне реализуются информационные функции для объекта в целом, а также управляющие функции, обеспечивающие оптимизацию отдельных технологических участков и объектов, настройку и адаптацию параметров регуляторов НЦУ к изменениям динамики объек-

та управления. На верхнем уровне выполняются расчёты неизмеряемых параметров по математическим моделям и технико-экономических показателей, создание и оперативное ведение базы данных всего технологического процесса. На этом уровне могут также решаться задачи по организации, синхронизации и координации взаимодействия технологических операций и оборудования. К организационным задачам относится комплекс операций по планированию технологического процесса, обеспечивающий эффективную загрузку оборудования и выполнение плановой задачи с максимальной скоростью. Задача синхронизации обеспечивает совпадение времени завершения и начала операций последовательного ТП в соответствии с требуемыми режимами работы производственной линии. Координационная задача обеспечивает целесообразное параллельное функционирование нескольких ТП в целях максимального производственного результата.

Подсистемы работают в режиме реального времени. Работа в режиме реального времени (или в реальном масштабе времени) означает, что события обрабатываются подсистемой настолько быстро и гарантировано предсказуемо, насколько это возможно, без промежуточной записи соответствующей информации в файл для ее более поздней обработки в пакетном режиме. Допустимое время обработки событий может составлять и минуты, и доли миллисекунд в зависимости от динамики технологического процесса. Интервал времени между некоторым событием в объекте и откликом системы реального времени на это событие называется *временем ответа*. Для организации комплекса задач реального времени в АСУ ТП необходимо реализовать взаимосвязь задач по управлению и информации, интенсивный обмен информацией с технологическим объектом, интенсивный обмен информацией с оперативным персоналом, функционирование подсистем в режиме реального времени. Для выполнения указанных требований в ПТК необходимо организовать следующие функциональные подсистемы.

- ввода и первичной обработки информации;
- передачи данных и обмена информацией в пределах ПТК;
- дистанционного управления приводами исполнительных механизмов;
- автоматического регулирования, автоматического дискретного управления (функционально-группового управления, защит и блокировок);
- информационно-вычислительную;
- представления информации и общения оператора с ПТК;
- инструментальную - для создания, контроля, отладки прикладных программ и выполнения технической диагностики;
- формирования алгоритмов сервисных функций.

Для объединения этих подсистем используется общая база данных реального времени. В общем случае не существует полностью формализованных методов синтеза ПТК по требуемым функциям, динамическим, надежностным и другим характеристикам ПТК. На практике выбор конкретной функционально-логической и технической структуры ПТК зависит от многих объективных (типа объекта, характеристик программно-технических средств) и субъективных (опыта, знаний проектировщика) факторов.

1.2. Функции ПТК

1.2.1. Информационные функции

Традиционно к информационным функциям относят [10]: сбор и первичную обработку измерительной информации; отображение информации на мониторах рабочих станций; регистрацию информации в нормальных режимах работы оборудования; идентификацию аварийных ситуаций и регистрацию информации в аварийных ситуациях; расчет и анализ технико-экономических показателей (ТЭП); расчет неизмеряемых параметров.

1.2.1.1 Сбор и первичная обработка аналоговой измерительной информации

Ввод аналоговых сигналов производится через специальные устройства связи с объектом (УСО). Они могут быть реализованы в виде отдельных устройств, либо в виде блоков элементов, устанавливаемых в общий каркас с процессором. УСО для ввода аналоговых сигналов обеспечивают прием множества разнообразных сигналов, формируемых датчиками и первичными преобразователями. Функция ввода аналоговых сигналов выполняется периодически. Для этого предусматриваются различные циклы в зависимости от динамики объекта. Обычно их величины находятся в пределах от единиц миллисекунд до единиц секунд. Вся информация собирается на нижнем уровне ПТК и подвергается первичной обработке, к которой относится: аналого-цифровое преобразование (если датчики или первичные преобразователи не осуществляют его сами), фильтрация, линеаризация сигналов датчиков с нелинейными характеристиками, компенсация температуры холодных спаев термоэлектрических преобразователей, масштабирование, контроль достоверности измерительной информации, контроль отклонения сигналов от уставок. Дополнительная погрешность, вносимая современными ПТК при обработке аналоговых сигналов, как правило, не превышает 0,1 % - 0,15 %. Некоторые аспекты первичной обработки измерительной информации приведены в приложении В.

Аналого-цифровое преобразование в ПТК выполняется, как правило, с разрешающей способностью 12-16 двоичных разрядов. Существенное внимание при этом уделяется подавлению помех промышленной частоты. Сегодня трудно найти ПТК, где подавление сигнала с частотой 50 ± 1 Hz было бы менее 60 dB (1000 раз), а в лучших образцах эта величина достигает 90 dB (10^5 раз). Каналы ввода аналоговых сигналов для уменьшения влияния разнообразных помех и токов утечки силового оборудования, существующих на больших промышленных объектах, стремятся делать с гальванической развязкой, как между различными каналами ввода, так и между каналами и вычислителем ПТК. Величина испытательного напряжения гальванической развязки обычно составляет 1-3 kV.

Контроль достоверности измерительной информации осуществляется, как правило, аппаратно в УСО и в вычислителе ПТК. В УСО выполняется диагностическая самопроверка функционирования, проверка исправности линий связи с датчиком, проверка на соответствие сигнала датчика диапазону измерения. УСО сообщает об этом вычислителю. При фатальных неисправностях УСО вычислитель теряет с ним связь. При обнаружении неисправности программа вычислителя должна воспринимать информацию от такого УСО как недостоверную. Недостоверность фиксируется индивидуально по каждому каналу, квалифицируется как событие и регистрируется функцией регистрации событий.

Для увеличения надежности сбора информации может применяться резервирование как УСО, так и датчиков (для некоторых важнейших параметров). Резервирование может осуществляться троированием или дублированием. В первом случае при любом единичном отказе датчика или УСО достоверность информации обеспечивается программно - выбором совпадающих (близких) сигналов по принципу мажорирования «2 из 3». Во втором случае, в особенности, когда отказ не является полным (обрывом или закороткой), а связан с ухудшением метрологических характеристик датчика или УСО, для определения какой из двух дублирующих сигналов недостоверен, должны применяться косвенные методы. Например, пусть ПТК, регулирующий температуру в помещении, оснащен датчиками температуры теплоносителя и дублированными датчиками расхода теплоносителя. По температуре в помещении и теплоносителя можно достаточно грубо оценить величину его расхода. Если показания датчиков расхода различны, то достоверно то, которое ближе к расчетному. В общем случае формализация метода выбора достоверного сигнала по косвенным признакам представляется невозможной ввиду многообразия и специфичности объектов. Использование этого метода относится к искусству разработчика АСУ ТП и является весьма дорогостоящим с точки зрения затрат на разработку. Необходимо отметить, что для тиражируемых АСУ ТП достоверность показаний всех, в том числе и нерезервированных датчиков (измерительных каналов), определяется по косвенным

признакам. Естественно, что определение достоверности как путем мажорирования, так и по косвенным признакам выполняется программно.

Контроль отклонения сигналов от уставок выполняется для достоверных сигналов циклически с циклом ввода аналоговых сигналов. Обычно используют четыре разновидности уставок: верхняя и нижняя границы технологического параметра; верхняя и нижняя границы аварийного параметра.

Иногда применяется контроль скорости изменения параметра и, соответственно, уставка максимальной (минимальной) скорости. Переход контролируемого параметра за какую-либо из установленных границ, как правило, инициирует одну из ветвей алгоритма управления технологическим процессом. Выходы за уставки и возвращение к норме фиксируются индивидуально для каждого параметра с указанием времени наступления события.

Для того чтобы избежать потока событий выхода за уставку и возвращения к норме в ситуации, когда реальное значение параметра очень близко к уставке, программы контроля реализуют функцию «зоны нечувствительности» или гистерезиса, которая сводится к тому, что принятие решения о превышении уставки происходит тогда, когда параметр превышает значение уставки «плюс» Δ (для верхней уставки), а решения о возвращении к норме, когда параметр меньше уставки «минус» Δ , где Δ – ширина зоны нечувствительности.

1.2.1.2. Сбор и обработка дискретных сигналов

Источниками дискретных сигналов являются разнообразные датчики положения («открыто-закрыто» задвижек, клапанов, дверей и т.п.), датчики фиксации событий в технологическом процессе (очередная деталь поступила на конвейер и т. п.), датчики достижения граничных значений (давления, уровня и т. п.), органы ручного управления (кнопки, переключатели и т.п.), сигналы, характеризующие режимы работы или информирующие о наступлении определенного события, формируемые отдельными устройствами автоматизации и многие другие.

Физически дискретные сигналы бывают двух типов:

- **потенциальные**, когда на вход УСО ПТК поступает сигнал напряжения (в промышленности широко используется напряжение 24 V постоянного тока, хотя могут использоваться и другие уровни напряжения, вплоть до 220 V переменного тока);

- **«сухой контакт»**, когда на вход УСО ПТК поступает значение сопротивления либо достаточно малое (контакт замкнут), обусловленное, в основном, сопротивлением линии связи, либо достаточно большое из-за конечного сопротивления изоляции линии связи. В качестве «сухого контакта» могут использоваться как механические контакты (концевые выключатели, реле), так и разнообразные электронные ключи (транзистор с открытым коллектором, твердотельные реле и т. п.).

Прием дискретных сигналов осуществляется специализированными УСО с гальванической развязкой входных цепей. Для потенциальных сигналов это реализуется достаточно просто, например, посредством оптронов, а для сигналов типа «сухой контакт» в УСО требуется наличие источника питания для каждого гальванически развязанного входного канала. Величина испытательного напряжения гальванической развязки в УСО ввода дискретных сигналов такая же, как и в УСО ввода аналоговых сигналов. Если в качестве «сухого контакта» используется механический контакт, то в силу упругих свойств механической системы контакта, а также из-за не идеальности (шероховатости) поверхности контакта при его замыкании-размыкании возникает явление «дребезга», которое сводится к многократным замыканиям и размыканиям, продолжающимся в течение 5-30 ms. В УСО и в программах ввода дискретных сигналов должны быть приняты меры, исключающие повторный ввод «дребезжащего» сигнала.

Дискретные сигналы разделяются на две группы – пассивные и инициативные. Сбор и обработка сигналов каждой группы производится по разным алгоритмам. Сбор и обработка пассивных сигналов выполняется периодически с величиной цикла, зависящей от конкретной АСУ ТП (обычно, от десятка миллисекунд до единиц секунды). Введенные в текущем цикле состояния дискретных сигналов сопоставляются с их значениями, введенными ранее. Изменение со-

стояния дискретного сигнала трактуется как событие, осуществляется привязка к текущему времени и производится запись в оперативную базу данных. Сбор и обработка инициативных сигналов отличаются от пассивных тем, что требуется немедленная реакция на их возникновение. Это может обуславливаться необходимостью точной фиксации времени наступления инициативного события, либо необходимостью принятия немедленного управленческого решения в АСУ ТП. Обработка инициативных сигналов может выполняться посредством прерывания вычислителя ПТК и немедленного запуска соответствующей обрабатывающей программы. Вместе с тем, известно множество рекомендаций, нашедших отражение даже в Государственных стандартах, к особо важным системам (например, обеспечивающим ядерную безопасность атомных электростанций), которые требуют минимизации использования механизма прерываний с целью повышения надежности программного обеспечения за счет большей предсказуемости (детерминированности) его функционирования. Поэтому в АСУ ТП особо ответственных объектов, а также благодаря резкому скачку вычислительной мощности процессоров ПТК, ввод инициативных сигналов выполняется как псевдоинициативный посредством специализированной процедуры с малым циклом ($\leq 1 \text{ ms}$) по алгоритмам ввода пассивных сигналов. Естественно, что в этом случае количество инициативных сигналов не должно превышать 10-20 % от общего количества дискретных сигналов в системе, иначе требования к скоростным характеристикам вычислителя ПТК нижнего уровня (если АСУ ТП физически иерархическая) могут стать чрезмерными.

Контроль достоверности дискретных сигналов выполняется с циклом их ввода и осуществляется аппаратно и программно. Аппаратно проверяется отсутствие обрыва линии связи и работоспособность УСО. Программно контролируется совокупный сигнал двух и более датчиков на отсутствие невозможных положений. Кроме того, исходя из особенностей технологического процесса конкретного объекта, может быть априори определена максимально возможная частота переключения для каждого дискретного сигнала (обычно несколько раз в секунду или реже). Если реальная интенсивность переключений превосходит эту величину, то можно предположить неработоспособность датчика или линии связи и считать такой сигнал недостоверным по причине нестабильности. Недостоверность квалифицируется как событие и регистрируется функцией регистрации событий. Повышение достоверности ввода дискретных сигналов может выполняться посредством резервирования по аналогии с вводом аналоговых сигналов.

1.2.1.3 Сбор и обработка кодированных сигналов

Уменьшение стоимости и рост функциональных возможностей микроэлектроники позволили поставить на серийное производство достаточно дешевые «интеллектуальные» датчики, которые способны выполнять некоторые операции первичной обработки, возлагавшиеся ранее на вычислитель ПТК (фильтрация, линеаризация, компенсация температуры холодных спаев термопар, довольно сложные операции по вычислению расхода газообразных сред и т.п.). Такие датчики могут быть групповыми, то есть измерять несколько, в том числе, разнородных параметров (например, для определения расхода газа необходимо измерить абсолютное давление, перепад давлений на сужающем устройстве, температуру газа). Информация от таких датчиков передается по двухпроводным линиям связи последовательным кодом. Датчики размещаются, как правило, непосредственно на объекте. Протяженность линии связи от них до ПТК может быть от сотен метров до нескольких километров, и они могут располагаться в непосредственной близости от источников промышленных помех. Поэтому для таких условий большое внимание уделяется повышению помехоустойчивости цифровых линий связи. Наибольшее распространение получили три интерфейса (не считая волоконно-оптических линий связи): токовая петля, HART (фирма Fisher-Rosemount) и AS-I (фирма Siemens). Скорость передачи информации по этим интерфейсам не велика и ограничивается 1 Kbit при расстояниях до 3 km и практически полной помехоустойчивости.

В составе ПТК нередко используются групповые преобразователи, осуществляющие прием и первичную обработку десятков-сотен сигналов от объекта. Обычно они располагаются в

специальных помещениях относительно недалеко (до десятка и сотен метров) от вычислителя ПТК. Основные интерфейсы, используемые для связи с ними – RS-485, CAN. Скорости передачи информации составляют от десятка до сотен *Kbod*.

В АСУ ТП сложных объектов входят АСУ ТП или САУ, поставляемые комплектно с основным оборудованием. Так, в состав теплового или атомного энергоблока входит паровая турбина со своей АСУ турбины (АСУТ), разработанной заводом-изготовителем турбины. Естественно, АСУТ должна быть связана с АСУ ТП энергоблока. Здесь используются интерфейсы RS-485 или CAN на предельных скоростях (до 2 *Mbod*) или оптический Ethernet (10/100 *Mbit/s*). Поэтому вычислитель современного ПТК нередко оснащается аппаратурой, обеспечивающей прием информации по последовательным линиям связи. Эту роль выполняют специализированные (предназначенные, как правило, для работы с одним из интерфейсов) связные модули, подключаемые к вычислителю ПТК либо на правах УСО, либо посредством высокоскоростных межблочных интерфейсов (ISA, Compact PSI, VME и т.п.).

Контроль достоверности передачи сообщений по последовательным линиям связи осуществляется с использованием избыточного кодирования информации (контроль по паритету, циклический контроль, формирование контрольных сумм и т.п.). Указанные способы обеспечивают достаточно высокую вероятность обнаружения сообщения с ошибками. Для устранения ошибок практикуется повторная передача сообщения, так как практически все протоколы обмена сообщениями по последовательным линиям связи предусматривают передачу сообщения-квитанции от получателя устройству-передатчику главного сообщения о наличии или отсутствии ошибок в принятом сообщении. При физическом нарушении линии связи в устройстве-передатчике срабатывает временной контроль (тайм-аут) на прием сообщения-квитанции. В случае неудачи повторной передачи может быть использована резервная линия связи.

1.2.1.4 Архивация

Эта функция предназначена для накопления и последующего представления оперативно-му и другому персоналу данных об истории протекания ТП, работе автоматики и действиях оператора. Архив заполняется при возникновении изменений состояний АСУ ТП. Обычно архивируется следующая информация:

- о событиях (функция регистрации событий);
- об аварийных ситуациях (функция регистрации аварийных ситуаций);
- о результатах расчета технико-экономических показателей (ТЭП);
- об изменениях во времени заданного набора параметров с целью выдачи оператору графиков изменения;
- о заданном наборе параметров для формирования часовых, сменных, суточных и других типов ведомостей;
- об изменениях состояний автоматических устройств с указанием источника управляющих воздействий (протокол состояния автоматики);
- о работе защит технологического оборудования от аварий;
- о работе технических и программных средств ПТК (протокол работы системы);
- о появлении и исчезновении недостоверной информации;
- другой информации в соответствии с техническим заданием на АСУ ТП конкретного объекта.

Информация из архива представляется в виде таблиц, протоколов, графиков и других форм, как на мониторах, так и в виде твердых копий. В нормальных режимах работы предусматриваются следующие виды регистрации:

- регистрация значений важных аналоговых параметров;
- регистрация событий по отклонениям аналоговых параметров, дискретным и формируемым сигналам;
- регистрация аналоговой информации для суточной ведомости;

- регистрация значений аналоговых параметров по запросу оператора;
- регистрация текущих состояний дискретных объектов;
- регистрация текущих значений аналоговых параметров.

Регистрация важных параметров обеспечивает непрерывную фиксацию в архивах значений параметров по списку постоянного состава. Для хранения зафиксированной информации предусматриваются сменный и долговременный архивы. Сменный архив используется для хранения сформированных значений параметров - средних за 60 s (при отсутствии отклонений) или наихудших (при наибольшем отклонении), а также суммарных длительностей отклонений, рассчитанных почасово по каждой границе контроля для каждого параметра. В долговременный архив заносятся часовые значения параметров, сформированные аналогично минутным (средние или наихудшие за час), и суммарные длительности отклонений, рассчитанные по-сменно. По данному виду регистрации предусматривается получение следующих протоколов:

- минутных значений параметров за задаваемый интервал времени с указанием признаков отклонений;
- "журнала нарушений" за задаваемую смену с указанием признака наибольшего отклонения за каждый час данной смены только по тем параметрам, по которым были зафиксированы нарушения;
- протоколов суммарной длительности отклонений по каждой из границ контроля, которые задаются интервалом времени (час, смена, сутки, месяц).

Регистрация текущих событий обеспечивает фиксацию и накопление в архиве текущих событий. При этом фиксируются следующие типы событий:

- по отклонениям непосредственно измеренных и рассчитанных аналоговых параметров от технологических уставок;
- изменения состояния оборудования (задвиги, механизмы и пр.);
- срабатывания защит;
- изменения всех дискретных сигналов;
- изменения состояния средств программно-технического комплекса.

Регистрация аналоговых параметров для ведения суточной ведомости обеспечивает фиксацию с периодом в 1-2 часа значений параметров по заданному списку. Зафиксированная информация сохраняется в архиве не менее суток.

Регистрация значений аналоговых параметров по запросу оператора обеспечивает фиксацию значений параметров по оперативно формируемым спискам с задаваемыми периодами.

Регистрация текущих состояний дискретных объектов обеспечивает фиксацию их состояний на момент запроса без предварительной архивации.

Регистрация текущих значений аналоговых параметров обеспечивает фиксацию значений параметров на момент запроса без предварительной архивации.

Распечатка протоколов всех видов регистрации осуществляется по запросу оператора в специально разработанных бланках печати. Формы бланков подразделяются на два основных вида: с разверткой по столбцам, т.е. с представлением в столбце процесса изменения каждого параметра и с разверткой по строкам, т.е. с представлением информации об одном параметре в строке. При распечатке протоколов регистрации возможен вывод информации в заранее подготовленные или оперативно формируемые бланки печати. Предусмотрена выборочная печать по задаваемому списку параметров или по признаку, например, по идентификатору или любой его части. Существует возможность предварительного просмотра протоколов регистрации на экране рабочей станции.

Регистрация аварийных ситуаций обеспечивает в хронологической последовательности фиксацию срабатываний технологических защит, событий и значений параметров объекта управления в аварийных ситуациях. Кроме того, обеспечивается проведение анализа аварийных ситуаций с установлением их первопричины на основе достаточно полной и достоверной информации в предаварийный и послеаварийный периоды. Регистрация аварийных ситуаций включает в себя следующие задачи: идентификацию аварии; фиксацию и архивацию аварийной информации; вывод на печать информации из аварийных архивов. При идентификации аварии

формируются сигналы начала и конца аварии, а также выводится сигнал аварии на все видеogramмы для сигнализации об аварийной ситуации. Этот сигнал формируется путем программной логической обработки сигналов, характеризующих аварию. Конец аварии определяется вводом оператором директивы "конец аварийной регистрации" или по истечении заданного интервала времени. В аварийной ситуации фиксируется следующая информация: последовательность срабатывания защит; состояние всех дискретных сигналов на момент начала аварии; значения важных аналоговых параметров за предаварийный, аварийный и послеаварийный периоды; события по аналоговым и дискретным сигналам за предаварийный, аварийный и послеаварийный периоды. Регистрация значений важных аналоговых параметров осуществляется по жесткому списку.

Информация по всем видам аварийной регистрации распечатывается в соответствующих бланках печати аналогично печати нормального режима. Пример протокола регистрации последовательности срабатывания защит приведен в табл.1.1. Для анализа зафиксированной информации предоставляются средства просмотра архивов на экране одной из рабочих станций. При этом обеспечивается вывод аналоговых параметров в виде графиков, совмещение их по времени с соответствующими им дискретными событиями и получение твердых копий за нужные интервалы времени (рис. 1.2).

Таблица 1.1

Протокол регистрации аварийного режима

.ПОСЛЕДОВАТЕЛЬНОСТЬ СРАБАТЫВАНИЯ ЗАЩИТ					
Начальное время - 26/02/2000 08:51:44					
Конечное время - 26/02/2000 09:39:47					
ЗМГРЭС		БЛОК 9	ДАТА	26/02/2000	ПРОТОКОЛ: АРЗ
Лист					
Нп/п	№ защ	Дата	Время	Шифр	Наименование
Начало блочной аварии 08:51:44					
1	042	26/02/2000	08:51:44:950	D102034	Понижение Т вт. пара за корп. Б
2	041	26/02/2000	08:51:53:524	D102033	Понижение Р до ВЗ, корпус Б
3	012	26/02/2000	08:52:05:754	D102010	Останов корпуса А
4	015	26/02/2000	08:53:12:454	D102013	Откл. сист. водородного охл. ген.
5	016	26/02/2000	08:53:25:650	D102014	Погасание факела корпуса А
6	014	26/02/2000	08:56:35:028	D102012	Снижение L в демпферном баке
7	014	26/02/2000	09:01:16:387	D102012	Снижение L в демпферном баке
8	017	26/02/2000	09:01:30:879	D102015	Вывод ключа защит корпуса А
Конец блочной аварии 09:39:47					
Конец документа					
Дата и время печати: 27/02/2000 10:01:33					

1.2.1.5 Расчет и анализ ТЭП

Расчет ТЭП предназначен для представления оперативному персоналу данных об эффективности работы технологического оборудования, что позволяет проводить коррекцию режимов и прогнозировать сроки вывода на ремонт оборудования; составления отчетных документов об экономичности объекта; проведения анализа работы оперативного персонала и автоматических систем; анализа причин ухудшения экономичности в межремонтный период для принятия решений о своевременном выводе оборудования в ремонт. В процессе расчета ТЭП предусматривается получение интегральных показателей с нарастающим итогом с начала месяца до момента запроса. Для оценки состояния оборудования предусматривается хранение пока-

зателей в архиве. Результаты расчетов выводятся по требованию оператора на экраны мониторов и принтеры в виде таблиц и графиков.

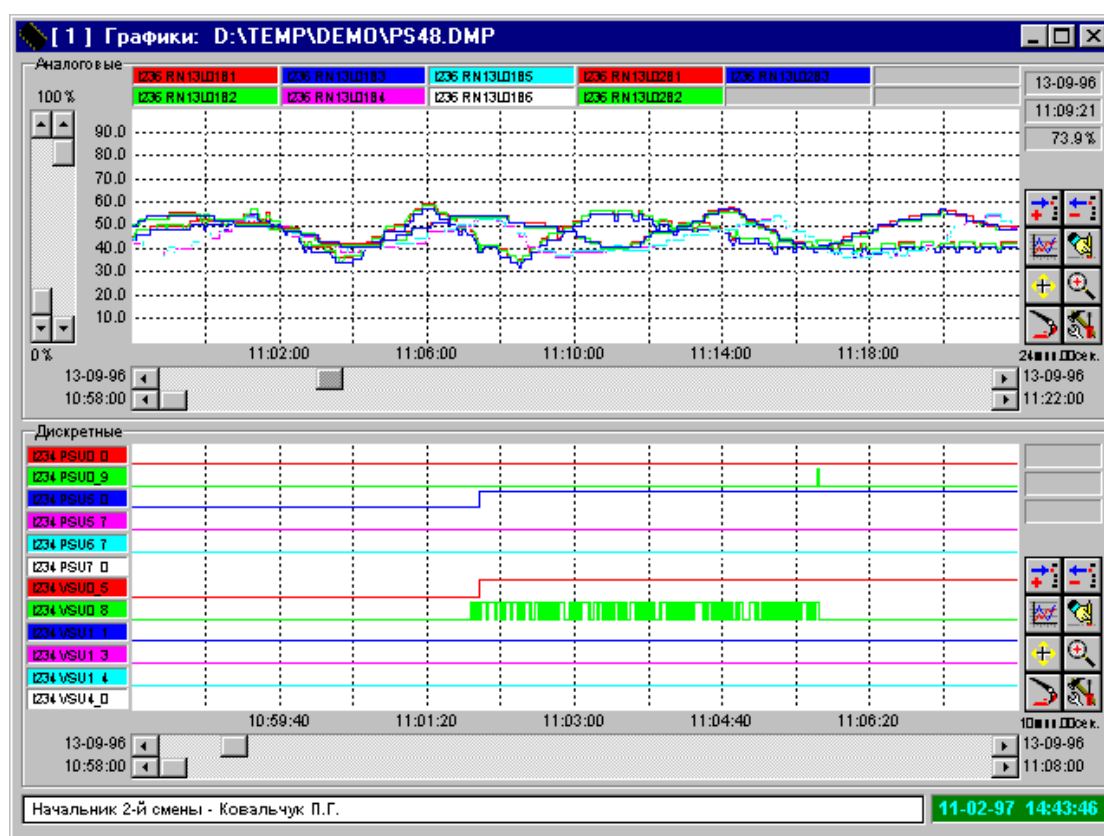


Рис. 1.2. Представление архивных данных в графическом виде.

1.2.1.6 Реализация информационных функций

Реализация информационных функций рассмотрена на примере ПТК для АСУ ТП тепловой электростанции (разработчик прикладного ПО – ХИКА). Вызов информации на дисплей операторских станций осуществляется оператором с помощью функциональных клавиатур и графических манипуляторов через системы меню. Информация на дисплеях отображается в следующем виде:

- *видеограмм*, представляющих собой фрагменты мнемосхем или таблицы;
- *аварийных и предупредительных сигналов* индивидуальной и групповой сигнализации;
- *графиков изменения технологических параметров*, показывающих тенденцию их развития, включая предысторию;
- *линейных* (гистограмм) и «зонтичных» диаграмм для технологически взаимосвязанных параметров с указанием заданных и предельных значений;
- *сообщений* о важнейших событиях (экран событий);
- *справочной информации* о параметрах.

Операторские станции могут использоваться также для дистанционного управления исполнительными механизмами. Ввод команд управления осуществляется со специальных клавиатур операторских станций, а контроль исполнения команд обеспечивается стандартными средствами представления информации на видеограммах.

Представление информации на видеограммах является вызываемым и заключается в вызове на экран библиотечного фрагмента. Мнемосхема представляет собой упрощенное графическое изображение технологической схемы, на которое выводятся значения аналоговых параметров, состояния объектов управления (механизмов, двигателей, задвижек, регулирующих органов), данные о воздействии автоматики и оператора на объекты управления и другая инфор-

мация, необходимая оператору для ведения технологического процесса. Мнемосхема удобна тем, что на ней наглядно представлена взаимосвязь между технологическими элементами, указаны точки измерений и значения параметров, наименования элементов и узлов, приведены различные вспомогательные надписи. Для обеспечения более детального анализа мнемосхемы дополняются таблицами, графиками или гистограммами.

Содержание фрагментов задается при проектировании системы разработчиками технологического объекта. Информация должна представляться по принципу «от общего к частному». Основной объем информации, позволяющий оценить ситуацию в целом, содержится на общих фрагментах мнемосхем. В случаях отклонений от нормальных режимов эксплуатации внимание оператора должно быть привлечено к этому отклонению. Пакет рабочих видеограмм обеспечивает оператора информацией по всем режимам работы оборудования. В состав пакета входят видеограммы различного назначения: нормального режима, пусковых (остановочных) режимов, аварийных ситуаций, технико-экономических показателей, диагностики технологического оборудования, контуров автоматического регулирования, контроля работы функционально-группового управления, диагностики средств программно-технического комплекса и содержащие дополнительную информацию (справки, инструкции и др.). Возможно формирование нескольких пакетов, рассчитанных на конкретного пользователя (оператора котла, оператора турбины, начальника смены блока и т.д.). Разработка пакетов видеограмм выполняется для каждого конкретного объекта и зависит от объема информации, введенной в систему. Для того, чтобы оператор мог качественно оценить работу технологического оборудования, разрабатываются видеограммы различных уровней:

- 1 - отображающие отдельный механизм,
- 2 - отображающие технологическую группу,
- 3 - отображающие технологические системы (видеограммы по котлу, турбине и т.д.),
- 4 - отображающие технологический процесс блока в целом.

Для обобщенного контроля за технологическим процессом создаются видеограммы, охватывающие объект в целом. Они представляются в виде таблиц, либо укрупненных мнемосхем, на которых изображаются только контуры основного оборудования (например, котел, турбина, генератор) и выводятся основные параметры блока.

Примеры фрагментов мнемосхем, используемых в составе информационно-вычислительной системы тепловой электростанции, показаны на рис. 1.3 - 1.5.

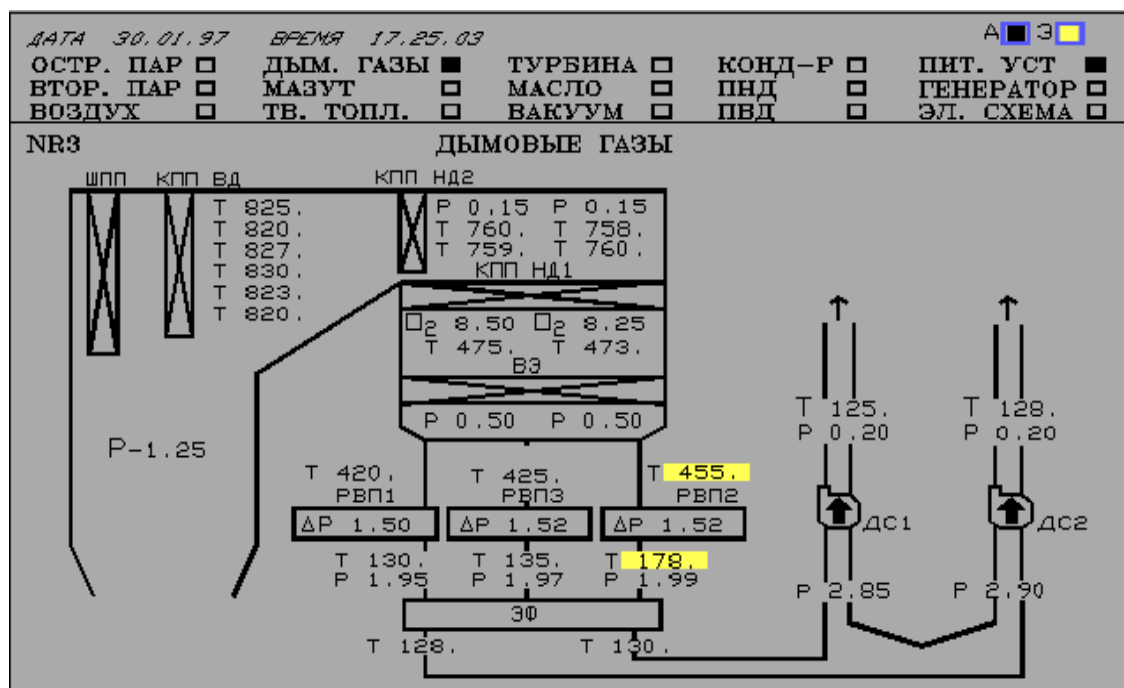


Рис. 1.3. Фрагмент мнемосхемы по технологической системе.

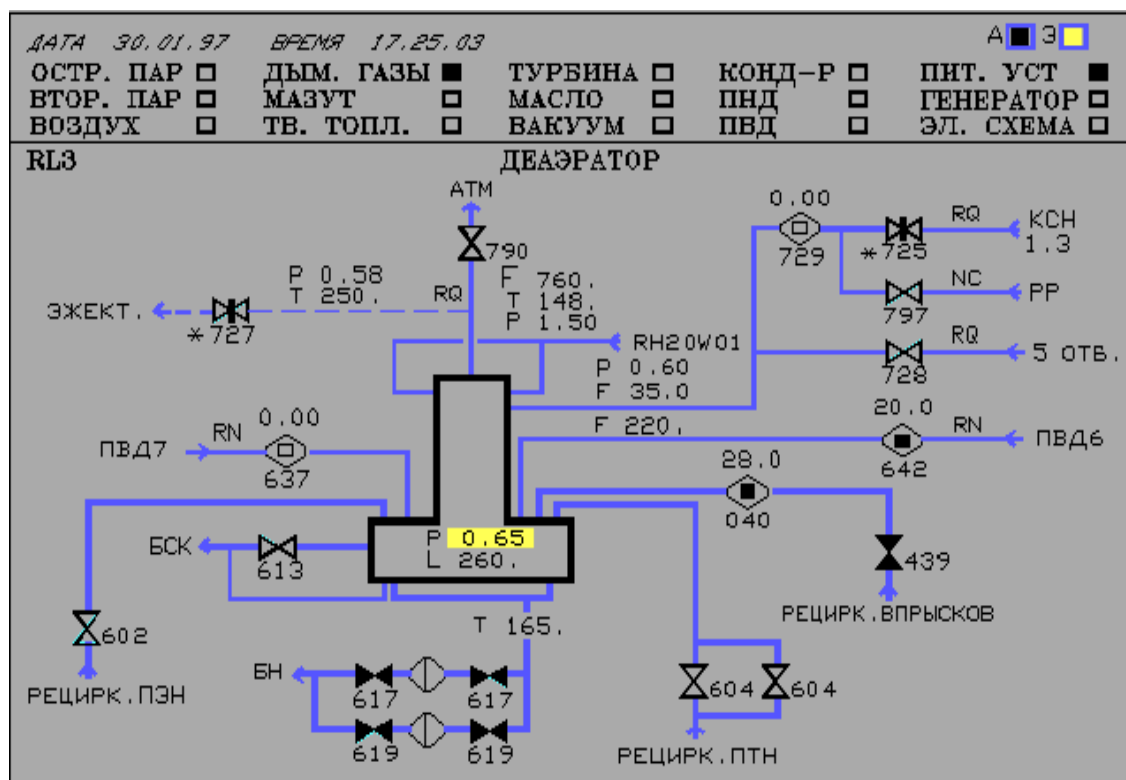


Рис. 1.4. Фрагмент мнемосхемы по технологической группе.

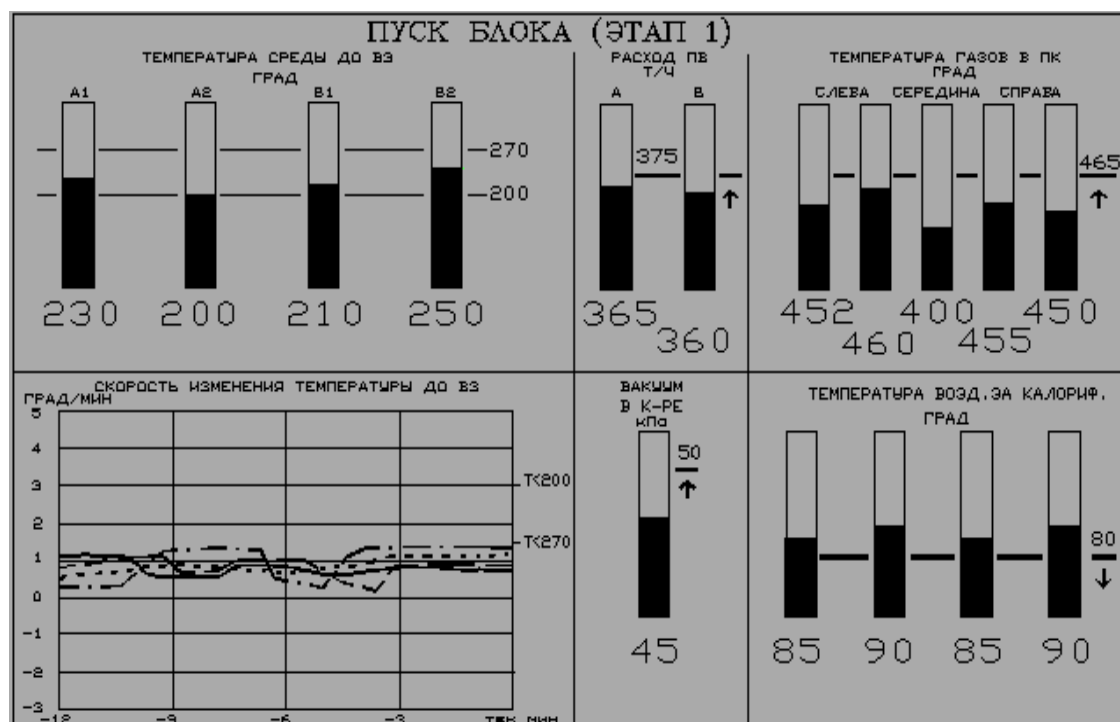


Рис. 1.5. Комбинированная видеодиаграмма пускового режима.

Индивидуальная сигнализация информирует об отклонениях аналоговых параметров и отображается на дисплеях изменением цвета символов значений параметров. Для каждого вида нарушений устанавливается свой цвет сигнализации. Нормальные значения параметров отображаются *зеленым* цветом, отклонившиеся за предельные границы - *желтым*, за аварийные - *красным* и за возможные значения (контроль достоверности) - *фиолетовым*. Значения откло-

няющихся от нормы параметров в отличие от формальных значений отображаются на цветных мигающих подложках. Аварийные ситуации различного рода, связанные со срабатыванием защит, а также локальные отказы оборудования и т.д. тоже охватываются индивидуальной сигнализацией.

Групповая сигнализация предназначена для привлечения внимания оператора к нарушениям на видеограммах, которые в данный момент не вызваны на отображение. Групповая сигнализация предоставляется операторам в виде сигнализации по группам параметров на клавиатурах операторских станций и сигнализации по группам параметров на видеограммах. Групповая сигнализация на функциональных клавиатурах заключается в подсветке клавиш вызова тех видеограмм, на которых индицируются отклонившиеся от нормы параметры. Подсветка сопровождается миганием и подачей звукового сигнала. После квитации звуковой сигнал исчезает, а подсветка переходит на ровное свечение. Каждое новое нарушение вновь вызывает мигание и подачу звукового сигнала. Групповой сигнал снимается после исчезновения всех нарушений в данной группе. Для быстрого поиска нарушений на всех видеограммах выделяется поле групповой сигнализации, на которое выводятся обобщенные сигналы по основным технологическим системам.

Для представления динамики изменения параметров во времени используются *графики*, обеспечивающие наглядное восприятие оперативным персоналом тенденции изменения параметров, включая предысторию. В виде графиков может быть представлен как один, так и группа параметров. Состав группы параметров может быть определен предварительно либо формироваться оперативно. Выбор группы осуществляется через меню. При отображении кривая каждого параметра индицируется своим цветом. При вызове графиков задаются параметры настройки - количество временных отсчетов и интервал между ними. Графики, как правило, представляются в относительной шкале с возможностью вывода абсолютной шкалы одного из параметров. Обеспечивается оперативное изменение масштаба шкалы одного из параметров, а также получение справок по параметрам. Существует возможность совмещения графиков параметров с видеограммами различного вида. Отображение параметров в виде графиков приведено на рис. 1.6.

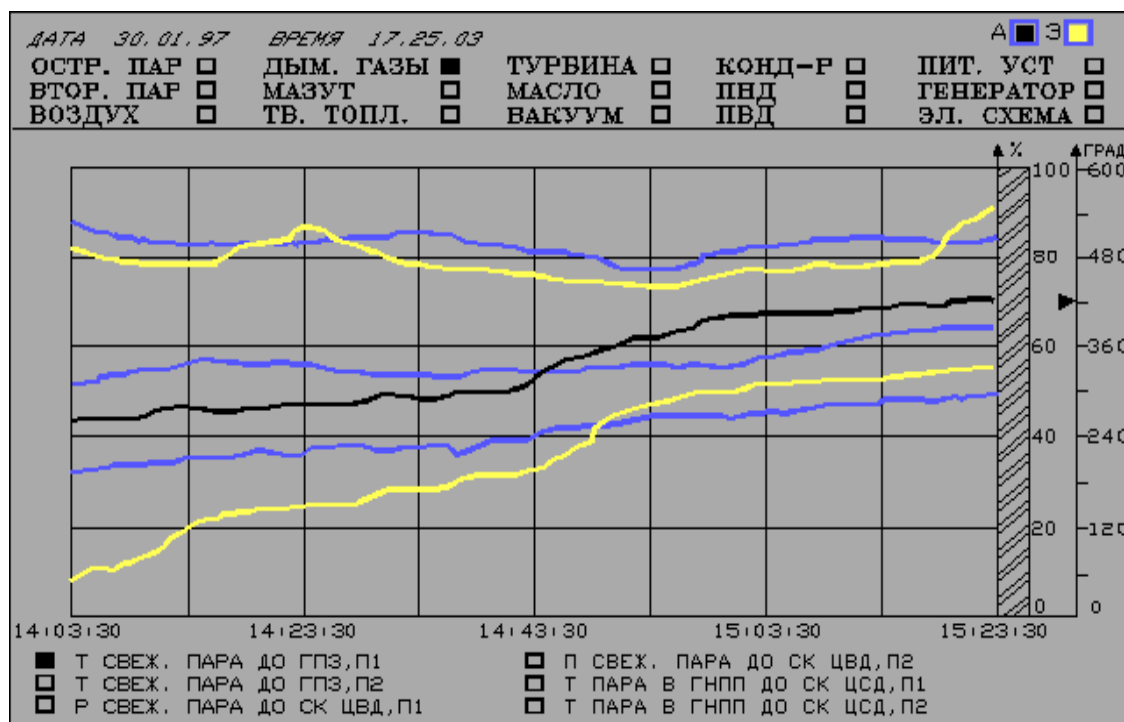


Рис. 1.6. Отображение параметров в виде графиков.

Одной из наиболее образных и наглядных форм представления взаимосвязанных аналоговых параметров являются *гистограммы*. Значения параметров на гистограммах представляются штриховыми площадками, длина которых изменяется пропорционально значениям параметров. Под каждой гистограммой располагается цветная линейка, указывающая границы контроля (уставки) и шкалу параметра. Гистограммы дополняются также текущими значениями параметров в цифровом виде и справочной информацией по каждому параметру. На одном экране одновременно могут отображаться значения не более 8-10 параметров. Аналогично графикам, группы параметров могут быть подготовлены заранее либо формироваться оперативно. Выбор группы осуществляется при помощи меню.

Предусматривается также отображение гистограмм совместно с мнемосхемой или графиками на видеограммах. В этом случае выполняются как вертикальные, так и горизонтальные гистограммы. Отображение значений параметров в виде гистограмм приведено на рис. 1.7. Оператор может получить информацию в форме *справки* по любому аналоговому и дискретному параметру, введенному в систему. Получение справочной информации по параметру обеспечивается маркерной операцией или специальной директивой. Справка содержит полную информацию о параметре, включая его текущее его значение.

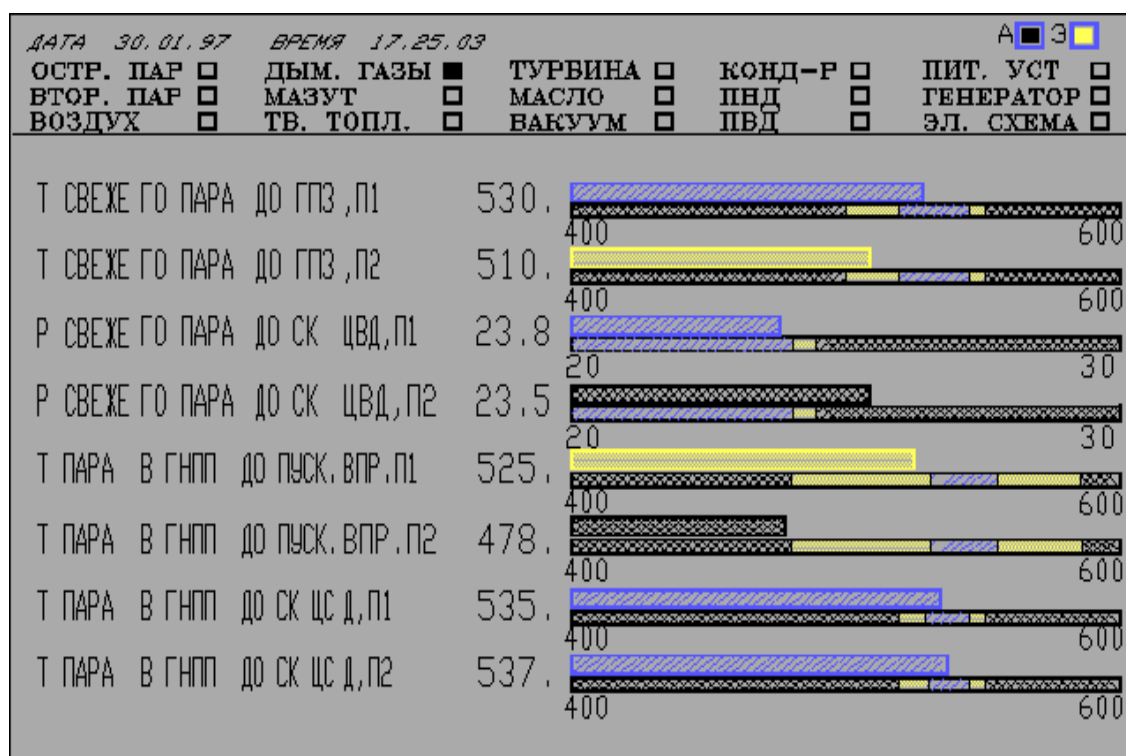


Рис. 1.7. Отображение значений параметров в виде гистограмм «Экран событий».

"Экран событий" позволяет оператору просматривать события по ограниченному количеству важнейших параметров на экране операторской станции. Сообщения появляются на экране в хронологической последовательности с указанием времени события. Сообщения по мере их появления движутся роллингом снизу вверх. Кроме этого обеспечивается ретроспективный просмотр (листание) событий ушедших с экрана. Сообщения о событиях выводятся на экран на подложках соответствующего цвета в зависимости от их важности (аварийные – красного, предупредительные – желтого цвета). При квитировании подложка исчезает, и сообщение индицируется в инверсном виде. Новые сообщения вновь индицируются на подложках. Появление каждого нового события или группы событий сигнализируется оператору мигающим специальным символом, выводимым на все видеограммы. Данная функция может быть вызвана оператором на любой из экранов операторских станций или отображаться постоянно на специально выделенном для этой цели экране.

1.2.2 Управляющие функции ПТК

К *управляющим функциям* относятся: дистанционное управление; автоматическое регулирование; автоматическое логическое управление; технологические блокировки; технологические защиты и защитные блокировки.

Дистанционное управление предназначено для реализации команд оператора-технолога по управлению оборудованием. Оно охватывает все исполнительные устройства, а также элементы автоматических устройств, предусматривающие включение (отключение) алгоритмов управления различного уровня и контуров автоматического регулирования. Примерный перечень объектов управления и воздействий на них приведен в табл. 1.2.

Таблица 1.2

Перечень объектов управления

Объект управления	Воздействие
Запорная арматура	Открыть-закрыть
Регулирующие клапаны	Больше - меньше
Запорные органы с соленоидным приводом	Открыть-закрыть
Электроприводы механизмов	Включить-отключить
Регуляторы	Автомат-дистанция
Задатчики регуляторов	Больше - меньше

Объект управления выбирается с помощью избирательной системы управления, которая реализуется в виде полей номеронабирателей на функциональных клавиатурах или через экран дисплея. В последнем случае объект отмечается на соответствующей видеограмме (фрагменте мнемосхемы) с помощью графического манипулятора или через сенсорный экран. Команды управления воздействуют только на выбранный объект. Выдача команд осуществляется нажатием специально промаркированных клавиш на функциональных клавиатурах или посредством воздействия на соответствующие элементы экранных меню. Выполнение воздействия на объекты управления производится в соответствии с алгоритмами управления исполнительными механизмами данного типа с учетом приоритетов воздействия от защит и блокировок от программ логического и местного управления. Последнее предполагает наличие органов ручного управления непосредственно на ТОО или местном щите управления.

Автоматическое регулирование может осуществляться как по стандартным законам регулирования (П, ПИ, ПИД) с необходимыми преобразованиями входной и выходной информации [13], которые обычно поддерживаются средствами автоматизации АСУ ТП, так и по алгоритмам произвольной сложности, разработанными при создании конкретной АСУ ТП. Для повышения качества регулирования (здесь и далее – по стандартным законам) в ПТК обеспечивают: многоконтурные схемы регулирования, динамические связи между контурами регулирования для компенсации связей через объект регулирования, автоподстройка, безударность при переходе с ручного на автоматическое управление и наоборот.

В рамках автоматического регулирования должны выполняться дискретные операции, задаваемые программными модулями из состава прикладных задач АСУ ТП, обеспечивающие свойства всережимности (изменение структуры регуляторов, параметров их настройки, и т.п.). В каждом контуре регулирования должны быть предусмотрены операции, выполняемые человеком, непосредственно взаимодействующим с ПТК через дисплеи и клавиатуры, такие, как: контроль регулируемого параметра, задания, рассогласования и положения регулирующего органа; возможность изменения сигнала задания, а также структуры регуляторов: контроль и изменение режима управления (автоматическое или ручное).

Автоматическое логическое управление, как и автоматическое регулирование, обычно поддерживается стандартными библиотечными программами и языковыми средствами. Вместе с тем, разработчикам произвольных алгоритмов регулирования и логического управления доступны супервизорные функции операционной системы вычислителя ПТК по организации многозадачного режима работы и функции синхронизации задач, как между собой, так и с внешни-

ми событиями. Программирование задач управления с произвольными алгоритмами осуществляется на универсальных языках (например, СИ). Стандартные средства поддержки логического управления позволяют решать следующие задачи: управление, реализуемое пошаговыми алгоритмами; управление, реализуемое последовательными логическими зависимостями (блокировки); комбинацию первых двух задач.

Управление, реализуемое последовательными логическими зависимостями, осуществляет анализ соответствия ряда исходных условий их заданным значениям и принимает решение об изменении состояния управляемого оборудования. Управление, реализуемое пошаговыми алгоритмами, предполагает выполнение последовательности (если необходимо - с ветвлениями) шагов. Причем каждый последующий шаг может выполняться, если образованы соответствующие условия. Пошаговое логическое управление предусматривает: контроль состояния оборудования; переключение запорной арматуры, механизмов и других исполнительных устройств в соответствии с алгоритмом; включение и отключение контуров регулирования, изменение структуры регулятора или параметров регулирования; формирование информации о ходе выполнения и завершения алгоритма, а также причинах приостанова выполнения. Алгоритмы шагово-логического управления представляют последовательность элементарных операций, которые необходимо выполнить для решения какой-либо технологической задачи. Выполнение команды и наличие разрешающих условий должно контролироваться по времени. Если в течение контрольного времени не выполнены разрешающие условия или не будет выполнена команда, дальнейшая отработка алгоритма должна прекращаться с выдачей сигнала о причинах его приостанова.

Инструментальные средства *проектирования и наладки алгоритмов управления* предназначены для автоматизированного выполнения следующих операций:

- проектирование алгоритмов управления (компоновки функциональных схем на экране монитора, документирования алгоритмов);
- преобразование алгоритмов управления в программы для контроллеров (генерации загрузочных модулей);
- оперативная коррекция настроек алгоритмов управления в контроллерах и базе данных;
- наладка алгоритмов, в т.ч. и с использованием моделей объекта в контроллерах;
- отображение информации о ходе процессов управления и состоянии объекта управления на мониторе при наладке алгоритмов.

В основу технологического языка проектирования положено графическое представление алгоритмов управления в виде взаимосвязанного набора типовых алгоритмических модулей (библиотеки функциональных схем) и табличное представление параметров настройки модулей. В библиотеку типовых модулей управления включены наиболее часто встречающиеся в схемах управления алгоритмы обработки входной информации и формирования управляющих воздействий. Библиотеку составляют как специализированные укрупненные модули (регулятор, шаг, блок управления клапаном и др.) так и универсальные элементарные модули (логическое И, ИЛИ и др.). Все процедуры выполняются на основе системы меню с использованием «оконного» интерфейса. По большинству позиций меню проектировщику предлагается выбор из списка допустимых параметров и настроек. Такой подход исключает внесение синтаксических ошибок и минимизирует количество семантических. Для иллюстрации на рис. 1.8 приведен вид экрана при настройке алгоритма регулятора питания и фрагменты меню.

Блокировки технологического оборудования предназначены для запрета выполнения какой-либо технологической операции при определенных условиях. Например, дверь кабины лифта не открывается при его движении между этажами, газовая колонка не включается при отсутствии давления воды, генератор электростанции нельзя подключать к сети электропередачи пока частота вырабатываемой электроэнергии не станет равной сетевой.

Алгоритмы блокировок представляют логические зависимости между входными условиями и состоянием подчиненных исполнительных устройств, которые не требуют для своего действия команд оператора и используют только внутреннюю информацию объекта. Особую группу алгоритмов блокировок образуют алгоритмы *аварийного включения резерва - АВР*. Они

обеспечивают подключение резервного устройства при аварийном отключении работающего или при недопустимом отклонении параметра, характеризующего исправную работу механизма. Характерным для этой группы блокировок является наличие внутренней памяти, фиксирующей факт первого включения рабочего устройства.

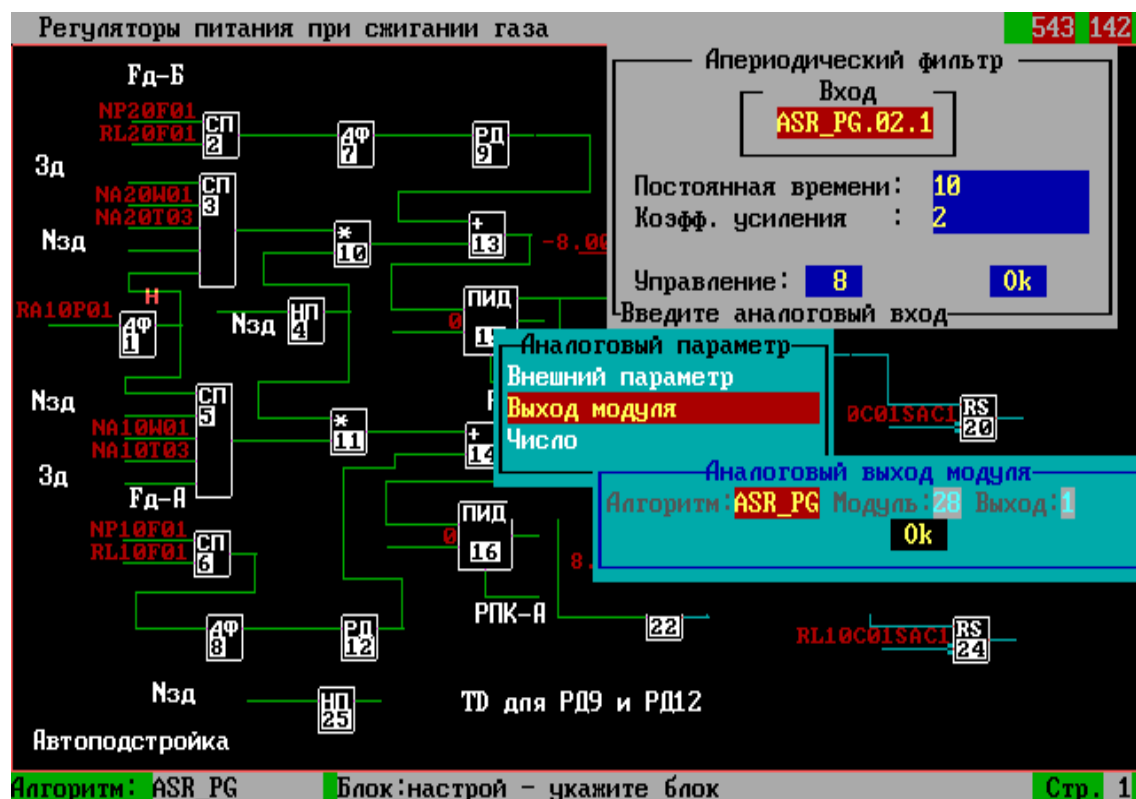


Рис. 1.8. Вид экрана при настройке алгоритма регулятора питания.

Технологические защиты предназначены для формирования управляющих воздействий на исполнительные органы с целью предотвращения повреждения оборудования, защиты персонала и локализации последствий аварий. Задачи, реализующие функции технологических защит, выполняются по техническим требованиям и алгоритмам, разработанным технологическими организациями с учетом требований нормативно-технологических документов. Требования к надежности автоматических защит, включая программное обеспечение, с формулировками отказов, ложных действий и значениями показателей надежности содержатся в техническом задании на конкретную систему.

Защиты имеют наивысший приоритет по отношению к другим дискретным действиям. Например, исключается возможность оперативного отключения защит оператором. Действие защит - одностороннее: ввод оборудования в работу после отключения его защитой должны производиться оперативным персоналом после устранения причин, вызвавших ее срабатывание. Кроме того, защиты имеют иерархическую структуру. При одновременном наступлении условий для действия нескольких защит приоритетом обладает защита высшей степени иерархии. Работоспособность и готовность технологических защит должны автоматически непрерывно контролироваться. Изменение уставок и других параметров защит производятся персоналом с санкционированием доступа и обязательным протоколированием.

Задачи, реализующие функции технологических защит, выполняют обработку следующей входной информации:

- сигналов от каждого канала защиты (типа «сухой контакт», от аналого-дискретных преобразователей, аналоговых компараторов);
- сигналов ввода в действие и отмены защит;
- команд на разрешение опробования защит и команды опробования.

Задачи формируют такие выходные команды: действия защит на оборудование, запрета отработки других каких-либо алгоритмов и включения аварийной светозвуковой сигнализации срабатывания защит. В зависимости от сложности объекта защиты являются одно или многошаговыми процессами. В последнем случае они реализуются по принципам пошагового логического управления.

К *сервисным* (вспомогательным) функциям относятся: метрологический контроль, аттестация, тестирование и самодиагностика устройств ПТК. Перечень и алгоритмы сервисных функций приводятся в техническом задании на конкретный объект.

1.3 Виды обеспечения ПТК

Создание и внедрение АСУ ТП, связано с реализацией различных видов обеспечения, которые находятся в тесной взаимосвязи. В соответствии со стандартом выделяют ряд следующих основных видов обеспечения АСУ ТП:

- математическое обеспечение;
- техническое обеспечение;
- программное обеспечение;
- лингвистическое обеспечение;
- информационное обеспечение;
- метрологическое обеспечение.

Математическое обеспечение - это совокупность математических методов, моделей и алгоритмов обработки информации, используемой при создании АСУ ТП. Остальные виды обеспечения реализуются в составе ПТК.

Техническое обеспечение - это комплекс технических средств (КТС), применяемых для функционирования АСУ. КТС предназначен для выполнения функций АСУ ТП, обеспечения создания АСУ различного масштаба, расширения системы или изменения ее структуры в процессе эксплуатации.

Программное обеспечение (ПО) разделяется на *базовое* и *прикладное*. В базовое (или общее) ПО входят: операционные системы, системы программирования, библиотеки программных модулей, реализующих типовые функции обработки информации; сервисные и контрольно-диагностические программы.

В соответствии со стандартами определены следующие термины.

Операционная система (ОС) - система программ, предназначенная для автоматизированного управления работой вычислительной системы и предоставления пользователю определенного набора услуг.

Система программирования - система автоматизации программирования, образуемая языком программирования, компиляторами или интерпретаторами программ, соответствующей документацией, а также вспомогательными средствами для подготовки программ к выполнению.

Пакет прикладных программ - это система прикладных программ, предназначенная для решения задач определенного класса. Программное обеспечение, используемое в АСУ ТП, должно отвечать следующим принципам: модульность построения; иерархичность организации; открытость (возможность расширения и модификации); гибкость (возможность внесения изменений и перенастройки); надежность (соответствие заданному алгоритму, отсутствие ложных действий), защита от разрушения и несанкционированного доступа; живучесть (выполнение возложенных функций в полном или частичном объеме при сбоях и отказах, восстановление после сбоев); унификация решений, простота и наглядность.

Прикладное программное обеспечение охватывает все функции контроля, управления и обработки информации в АСУ ТП. Оно предусматривает возможность загрузки прикладного ПО через интерфейсные каналы в память контроллеров ПТК. Прикладное ПО позволяет выполнять изменение и корректировку непосредственно на объекте без привлечения разработчиков и профессиональных программистов.

Лингвистическое обеспечение представляет собой совокупность средств и правил, используемых при общении пользователей и эксплуатационного персонала с комплексом средств ПТК при его разработке, монтаже и эксплуатации.

Информационное обеспечение - это совокупность решений по объемам, размещению и формам организации информации, циркулирующей в АСУ при ее функционировании.

Метрологическое обеспечение - совокупность технических средств, требований, положений, правил, норм и методик, направленных на обслуживание ПТК как средства измерения.

1.3.1 Лингвистическое обеспечение

Лингвистическое обеспечение рассчитано на пользователя (специалиста), не владеющего универсальными языками программирования или описания алгоритмов. Лингвистическое обеспечение оператора-технолога представляет собой систему видеogramм и текстовых сообщений, снабженных необходимыми «меню», «подсказками» и «помощью». Кодирование информации должно производиться единообразно. Лингвистическое обеспечение разработчиков, наладчиков и обслуживающего персонала содержит:

- средства, способы, пакеты программ генерации структуры комплекса технических средств и программного обеспечения;
- средства заполнения баз данных;
- системы описания и программирования характерных задач контроля и управления - первичной обработки информации, автоматического регулирования, автоматического логического управления, действия защит, дистанционного управления;
- способы формирования видеogramм, отчетов, протоколов, ведомостей и архивов;
- способы включения в систему типовых информационных задач;
- средства проведения тестирования, диагностирования и других регламентных работ.

Системы описания типовых информационных задач включают подсистемы генерации видеogramм, отчетов и ведомостей. Подсистема генерации видеogramм содержит редактор изображений, средства организации библиотек изображений, библиотеки типовых изображений, средства описания «анимации» изображений, «меню», способов формирования, условий вывода и задания текста технологических сообщений различного класса. Подсистема генерации отчетов содержит средства формирования форматов отчетов, средства описания данных, включаемых в отчет, условий формирования (вывода) отчета, средства задания операций над данными. Подсистема генерации архивов содержит средства описания архивов (название, глубина хранения, условия уничтожения и т.п.), информации, вводимой в архив, и условий ее записи. Лингвистическое обеспечение реализуется в составе прикладного программного обеспечения при помощи SCADA-пакетов.

1.3.2 Информационное обеспечение

Основные принципы организации информационного обеспечения включают создание единой информационной базы (ИБ), разработку типовой схемы обмена данными между системой и оперативным персоналом, внесение в нее изменений и выдачу данных, разработку единой общесистемной схемы хранения и обеспечения решаемых задач исходными данными, обеспечение возможности поэтапного и непрерывного наращивания емкости информационной базы, обеспечение одноразовости и независимости ввода данных от времени решения и количества решаемых задач.

Исходя из этих принципов, основными задачами информационного обеспечения являются: определение форм информационного представления объектов и процессов; определение структуры и состава информации и ее увязка с решаемыми задачами; формирование нормативного словаря для обозначения и описания объектов и их свойств.

В настоящее время, как правило, ИБ формируется в виде совокупности баз данных. «База данных АСУ» - это множество используемых при функционировании АСУ данных, организо-

ванное по определенным правилам, предусматривающим общие правила хранения, описания и манипулирования данными, и независимая от прикладных программ.

Способы хранения и передачи информации должны предусматривать ее помехоустойчивое кодирование и защиту от разрушения и несанкционированного доступа. Для кодирования технологического оборудования, технических средств ПТК, алгоритмов и программ должна использоваться единая система кодирования. Информационное обеспечение реализуется в составе базового и прикладного программного обеспечения.

1.3.2.1 Архитектура информационной базы

Эффективность функционирования АСУ во многом зависит от архитектуры информационной базы. В настоящее время перспективной является архитектура клиент-сервер. Она предполагает наличие компьютерной сети и распределенной базы данных, включающей общую (ОБД) и локальные базы данных (ЛБД). ОБД размещается на компьютере-сервере, ЛБД размещаются на компьютерах оперативного персонала, являющихся клиентами ОБД.

Сервером определенного ресурса в компьютерной сети называется компьютер (программа), управляющий этим ресурсом, клиентом. Компьютер (программа) использует этот ресурс. В качестве ресурса компьютерной сети могут выступать, к примеру, базы данных, файловые системы, службы печати. Тип сервера определяется видом ресурса, которым он управляет. Если управляемым ресурсом является база данных, то соответствующий сервер называется сервером базы данных. Достоинством организации информационной базы по архитектуре клиент-сервер является сочетание централизованного хранения, обслуживания и коллективного доступа к общей информации с индивидуальной работой над персональной информацией.

Общая база данных создается, поддерживается и функционирует под управлением сервера БД, например, Microsoft SQL Server или Oracle Server. Для создания и управления функционированием ЛБД и приложений, работающих с ними, используются СУБД такие, например, как Access и Visual FoxPro фирмы Microsoft, Paradox фирмы Borland. В зависимости от размеров организации и особенностей решаемых задач информационная система может иметь одну из следующих конфигураций:

- компьютер-сервер, содержащий общую и персональные базы;
- компьютер-сервер и персональные компьютеры с ЛБД;
- несколько компьютеров-серверов и персональных компьютеров с ЛБД.

Использование архитектуры клиент-сервер дает возможность постепенного наращивания информационной базы, во-первых, по мере развития объекта управления; во-вторых, по мере развития самой информационной базы. Разделение данных на ОБД и ЛБД позволяет уменьшить сложность проектирования БД по сравнению с централизованным вариантом, а значит, снизить вероятность ошибок при проектировании и его стоимость.

Важнейшим достоинством применения баз данных в АСУ ТП является обеспечение независимости данных от прикладных программ. Это позволяет не обременять пользователей проблемами представления данных на физическом уровне: размещение данных в памяти, методов доступа к ним и т.д. Такая независимость достигается поддерживаемым СУБД многоуровневым представлением данных на логическом (пользовательском) и физическом уровнях. Благодаря СУБД и наличию логического уровня представления данных обеспечивается отделение концептуальной (понятийной) модели БД от ее физического представления в памяти ЭВМ.

Модели данных это - хранимые в базе данные, которые имеют определенную логическую структуру, поддерживаемую СУБД. Известны следующие модели данных: иерархическая, сетевая, реляционная и объектно-ориентированная.

Реляционная модель данных получила название от английского термина **relation** - отношение. При соблюдении определенных условий отношение представляется в виде двумерной таблицы, привычной для человека. Большинство современных БД для персональных ЭВМ являются реляционными. Достоинством реляционной модели данных являются ее простота, удобство реализации на ЭВМ, наличие теоретического обоснования и возможность формирования

гибкой схемы БД, допускающей настройку при формировании запросов. Она используется, в основном, в БД среднего размера. При увеличении в базе данных числа таблиц заметно падает скорость работы с этой базой.

В АСУ ТП для управления динамическими процессами применяют жестко организованные иерархические модели баз данных. Объектно-ориентированные БД объединяют в себе две модели данных: реляционную и сетевую и используются для создания крупных БД со сложными структурами данных.

Система управления базами данных (СУБД) представляет собой совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД. СУБД, как правило, разделяют по используемой модели данных (как и базы данных) на следующие типы: иерархические, сетевые, реляционные и объектно-ориентированные. По характеру использования СУБД делят на персональные и многопользовательские.

Многопользовательские СУБД включают в себя сервер БД и клиентскую часть, работают в неоднородной вычислительной среде (допускаются разные типы ЭВМ и различные операционные системы). Поэтому на базе СУБД можно создать информационную систему, функционирующую по технологии клиент-сервер. Универсальность многопользовательских СУБД отражается соответственно на высокой цене и требуемых компьютерных ресурсах для их поддержки.

Персональные СУБД обеспечивают возможность создания персональных БД и недорогих, работающих с ними, приложений, а также приложений, работающих с сервером БД.

Управляющим компонентом многих СУБД является ядро, выполняющее следующие функции: управление данными во внешней памяти, управление буферами оперативной памяти (рабочими областями, в которые осуществляется подкачка данных из базы для повышения скорости работы), управление транзакциями.

Транзакция - это последовательность операций над БД, которая рассматривается СУБД как единое целое. При выполнении транзакция может быть либо успешно завершена, и СУБД зафиксирует произведенные изменения во внешней памяти, либо (например, при сбое в аппаратной части ПК) ни одно из изменений не отразится в БД. Понятие транзакции необходимо для поддержания логической целостности БД.

Язык современной СУБД включает подмножества команд, относящихся к следующим специализированным языкам:

- язык описания данных - высокоуровневый непроцедурный язык декларативного типа, предназначенный для описания логической структуры данных;
- язык манипулирования данными - командный язык СУБД, обеспечивающий выполнение основных операций по работе с данными - ввод, модификацию и выборку данных по запросам;
- структурированный язык запросов SQL (Structured Query Language) - обеспечивает манипулирование данными и определение схемы реляционной локальной БД - является стандартным средством доступа к серверу БД.

Для обработки команд пользователя или оператора программ в СУБД используются интерпретаторы команд (операторов) и компиляторы.

Обеспечение целостности БД является необходимым условием успешного его функционирования. Целостность БД это - свойство, означающее, что база данных содержит полную и непротиворечивую информацию, необходимую и достаточную для корректного функционирования приложений. Для обеспечения целостности БД накладывают ограничения целостности в виде некоторых условий, которым должны удовлетворять данные, хранимые в базе. Примером таких условий может служить ограничение диапазонов возможных значений атрибутов объектов, сведения о которых хранятся в БД, или отсутствие повторяющихся записей в таблицах реляционных БД.

Обеспечение безопасности достигается в СУБД шифрованием прикладных программ, данных, защиты паролем, поддержкой уровней доступа к базе данных, к отдельной таблице. Расширение возможностей пользователя СУБД достигается за счет подключения систем по-

строения графиков и диаграмм, а также подключения модулей, написанных на языках СИ или Ассемблера. Поддержка функционирования в сети обеспечивается:

- средствами управления доступом пользователей к совместно используемым данным, т.е. средствами блокировки файлов (таблиц), записей, полей, которые в разной степени реализованы в разных СУБД;

- средствами механизма транзакций, обеспечивающими целостность БД при функционировании в сети.

1.3.3 Метрологическое обеспечение

Метрологическое обеспечение это - совокупность технических средств, требований, положений, правил, норм и методик, направленных на обслуживание ПТК как средства измерения. Оно должно охватывать все стадии создания ПТК и эксплуатации системы на объекте. Метрологическое обеспечение распространяется на измерительно-информационные каналы и алгоритмы контроля технологического процесса и оборудования ТОУ, включая расчетные алгоритмы. Организационно-технические мероприятия по метрологическому обеспечению должны предусматривать: определение обобщенных метрологических характеристик измерительного канала по метрологическим характеристикам средств обработки информации и средств измерений; проведение аттестации измерительных и расчетных каналов путем первичной и периодической поверки (калибровки); оснащение устройств ПТК аппаратными средствами метрологического контроля, а обслуживающей ПТК лаборатории - высокоточной аппаратурой для проведения поверок и градуировок; проведение метрологической экспертизы проектной документации; проверку соответствия фактической реализации проектным решениям.

Измерительные каналы проходят метрологическую аттестацию как при вводе системы в эксплуатацию, так и в процессе эксплуатации. Алгоритмы и программы расчетов, производимых в системе, должны быть аттестованы. Технические средства и подсистемы, характеристики которых влияют на точность, проходят государственные испытания или метрологическую аттестацию. В основном, это касается АСУ ТП, осуществляющих коммерческий учет (например, энергоносителей) или влияющих на безопасность (например, АСУ ТП энергоблока атомной электростанции).

Материалы, касающиеся метрологии, изложены во многих работах. Одной из заметных работ, в которой на основе системно-ориентированного подхода изложены теория и практика измерений, является [14]. Далее будут рассмотрены только некоторые аспекты, существенные для технических измерений.

Основным способом получения информации о ТОУ является измерение. Измерение – это получение информации с помощью измерительных систем в форме результата измерения, отражающего характеристику о состоянии объекта измерений. ПТК можно рассматривать как одну из разновидностей измерительных систем. Основной информацией, используемой в ПТК, является метрическая информация, то есть количественная информация о величине, амплитуде или интенсивности определенной характеристики объекта.

При выполнении измерений различают [15] лабораторные и технические измерения, статические и динамические, а также прямые и косвенные. Под лабораторными понимаются измерения, при выполнении которых производится учет точности измерений, а под техническими - измерения, для которых наперед принимается определенная точность, достаточная для данной практической цели [16]. Технические измерения являются основной операцией, ядром процессов получения измерительной информации в АСУ ТП, при испытаниях и контроле качества продукции и т.д. Методики технических измерений базируются на метрологических нормативных документах, регламентирующих обязательные требования и правила планирования, организации и проведения технических измерений.

Статический режим измерений это процесс измерения, при котором можно пренебречь изменением измеряемых величин во времени. На практике он может встречаться только при целенаправленном оценивании постоянных погрешностей, когда измеряемую величину специ-

ально поддерживают неизменной с целью определения отличия постоянного выходного сигнала средства измерений от того «номинального» значения, которое соответствует постоянному входному сигналу. При практических измерениях измеряемая величина медленно или быстро, но всегда изменяется. Например, в АСУ ТП типичным является непрерывное (медленное или быстрое) изменение той величины, которая служит критерием управления. Известно, что при некоторых свойствах как измеряемой величины (или процесса, информативным параметром которого она является), так и средств измерений погрешность измерений будет разной при одних и тех же значениях измеряемой величины (и других условиях измерений), но при разных скоростях ее изменений.

Погрешности, вызываемые влиянием скоростей изменения измеряемой величины, называются динамическими погрешностями. Известны две общие причины влияния скорости изменения измеряемой величины на результаты измерений, т. е. на погрешности измерений. Первая - это инерционные (или динамические) свойства средств измерений. Это те их свойства, вследствие которых уравнения, связывающие между собой выходной и входной сигналы средств измерений, являются не алгебраическими, а дифференциальными. Ясно, что вопрос о том, нужно ли при данной скорости изменений измеряемой величины учитывать динамическую погрешность, должен решаться на основе не только этой скорости, но и динамических свойств средств измерений, а также наименьшего уровня динамической погрешности, при котором ее необходимо учитывать. Таким образом, признаком, по которому измерение должно быть отнесено к статическим или динамическим, является динамическая погрешность измерений при данной области скоростей (или частот) изменений измеряемой величины и при данных динамических свойствах средств измерений. Если при данной скорости изменений измеряемой величины и при применяемых средствах измерений динамическая погрешность настолько мала, что в данной измерительной задаче ее можно не учитывать, то такие измерения можно отнести к статическим. В противном случае - к динамическим. Это признак условный, не определяющий жестко скорость изменения измеряемой величины, при которой измерение должно быть признано статическим или динамическим, но для практических измерений (особенно технических) весьма удобный.

Прямое измерение – измерение, при котором значение измеряемой величины определяется непосредственно по показаниям измерительного прибора.

Косвенное измерение - измерение, при котором значение измеряемой величины, представляющей собой известную функцию других величин, определяется путем расчета значения данной функции по результатам прямых измерений величин – аргументов функции.

Методику выполнения технических измерений (МТИ) удобно определить, как обобщенное понятие, включающее в себя процедуру использования соединенных между собой и с объектом измерений технических средств (в том числе, средств измерений) определенных типов, направленную на получение конечного для данной МТИ результата - результата измерений, отражающего определенные свойства объекта измерений. Методика выполнения технических измерений должна обеспечивать получение результатов измерений, погрешность каждого из которых находится в определенных пределах для определенной группы объектов измерений при определенных условиях. МТИ, предназначенная для получения результатов измерений, представляет собой следующее:

- совокупность средств измерений (СИ), других технических средств, материалов и деталей определенных типов, программных средств; схему соединения с объектом измерений и средств измерений между собой, других технических средств, материалов и деталей; процедуру проведения измерений, то есть все операции, которые необходимо провести при измерениях;
- формулы (или алгоритмы) расчета результатов измерений (при косвенных измерениях);
- формулы (алгоритмы) расчета характеристик погрешностей, которые можно приписать любым получаемым результатам измерений в известных условиях.

При заданной группе объектов измерений, измеряемых величин и их значений в заданном диапазоне, заданном диапазоне скоростей (частот) изменений измеряемых величин или частотном спектре процесса (информативным параметром которого является измеряемая величина),

заданных внешних условиях МТИ должна обеспечивать получение результатов измерений с погрешностями, характеристики которых не выходят за заданные допустимые пределы. Можно сказать, что МТИ - это своеобразный «технологический процесс» измерений.

Принципиальным является то, что определение МТИ относится не к «документу» или «установке», а к понятию. Конечно, одного этого понятия недостаточно для практики технических измерений. Но оно важно, как принципиальная основа технических измерений.

Для технических измерений наиболее важны классификации погрешностей по двум направлениям: *методические* и *инструментальные*; *систематические* и *случайные*. Характерной особенностью методических погрешностей является то, что они обусловлены свойствами только методик измерений и не зависят от свойств применяемых средств измерений. Наоборот, характерной особенностью инструментальных погрешностей является то, что они обусловлены свойствами применяемых средств измерений. Можно сказать, что методические погрешности измерений определяют потенциальные возможности методик измерений. Методические погрешности можно представить как полные погрешности измерений при применении «идеальных» средств измерений. Под «идеальным» можно понимать средство измерений, удовлетворяющее трем условиям:

- действительная функция преобразования в реальных условиях применения средства измерений точно совпадает с его номинальной функцией преобразования, т. е. погрешность средства измерений в реальных условиях применения равна нулю;

- при взаимодействии с объектом средство измерений на него не воздействует, то есть не изменяет его состояния;

- средство измерений обладает бесконечной пространственной разрешающей способностью по координатам, по которым изменения измеряемой величины должны различаться.

К инструментальным погрешностям, можно отнести следующие основные три группы погрешностей. К первой группе относится погрешность средств измерений в реальных условиях их применения. Ко второй группе относится погрешность, обусловленная взаимодействием средств измерений с объектом измерений. За исключением редких случаев (таких, например, как измерение длины с помощью измерительной линейки), при измерениях происходит обмен энергией между объектом измерений и средством измерений. Тривиальным примером подобного обмена энергией может служить измерение вольтметром выходного электрического напряжения генератора. В зависимости от свойств выходной цепи генератора и входной цепи вольтметра последний поглощает большую или меньшую часть энергии от генератора. В результате, выходное напряжение генератора будет отличаться от того, какое имело место на выходе генератора до подключения к нему вольтметра. Это может вызвать некоторую погрешность измерений, если задачей измерений предусматривается определение выходного напряжения самого генератора. Другим примером может служить измерение температуры достаточно тонкой металлической пластины с помощью термопары. Контакт термопары с пластиной приводит к некоторому уменьшению температуры в месте контакта. Уменьшение температуры зависит от свойств как пластины, так и термопары. Для измерений температуры пластины может применяться и термометр сопротивления, рассеивающий, как известно, некоторую энергию. В зависимости от свойств пластины и термометра сопротивления, температура пластины в месте контакта может как понизиться, так и повыситься. В обоих случаях взаимодействие средства измерений с объектом измерений приводит к некоторому изменению измеряемой величины относительно того ее значения, которое требуется определить путем измерений. Возникает соответствующая погрешность измерений, зависящая от свойств как объекта измерений, так и средства измерений. В различных измерениях взаимодействие средств измерений с объектами измерений может быть и более сложным. Нужно отметить, что соответствующие погрешности измерений достаточно хорошо изучены лишь для электрических измерений при линейных свойствах выходных цепей объектов измерений и входных цепей средств измерений. Что касается измерений электрических величин в нелинейных цепях и измерений неэлектрических величин, то задача исследований погрешностей, обусловленных взаимодействием объектов измерений с их средствами, еще изучена недостаточно. Погрешности измерений, обусловленные взаимодей-

ствием средств измерений с объектом измерений, зависят от их свойств, в том числе, и от свойств средств измерений. Поэтому они относятся к инструментальным погрешностям измерений. К третьей группе относятся погрешности, зависящие от некоторого специфического свойства, которое предназначено для таких средств измерений. Это свойство может быть обусловлено тем, что измерение величины строго в точке пространства, как правило, не может быть осуществлено из-за ограниченной пространственной разрешающей способности средства измерений. Практически измеряются величины, усредненные на некоторых малых интервалах длины, на малых площадках, объемах, покрывающих те дискретные точки пространства, в которых требуется определить значения измеряемой величины. Это вызывает соответствующую составляющую погрешности измерений, зависящую от свойств средств измерений. Таким образом, к инструментальным погрешностям измерений можно отнести следующие:

- погрешность средств измерений в реальных условиях их применения;
- погрешность, обусловленную взаимодействием средства измерений с объектом измерений;
- погрешность, обусловленную конечной пространственной разрешающей способностью средств измерений.

При применении цифровых средств измерений иногда нужно учитывать инструментальную погрешность измерений, обусловленную некоторым несоответствием результата измерения значению измеряемой величины в тот момент времени, которому приписывается этот результат.

Систематические погрешности обусловлены несовершенством измерительных приборов, методов измерений, мер, а также влиянием окружающих условий, оцениваемых с помощью измерительных средств или же расчетным путем, и влиянием операторов, производящих измерения. Они являются, как правило, преобладающими, представляют собой погрешности, которые в принципе можно оценить, исключить или уменьшить. Неисключенные составляющие систематической погрешности увеличивают случайную погрешность, что приводит к расширению доверительных границ погрешности результатов измерения.

Случайные погрешности являются следствием влияния на объект измерений случайных факторов. Такие условия существуют при проведении измерений определенной величины на одном и том же объекте одним и тем же оператором и с использованием одного и того же средства и метода измерения. Разброс измеренного значения обусловлен изменяющимися во времени и пространстве причинами, вызывающими появление погрешности. Поэтому случайные погрешности можно оценить, но невозможно исключить.

Погрешность средства измерений можно представить как состоящую из следующих трех составляющих:

- основной погрешности;
- дополнительной погрешности, обусловленной отклонением влияющих величин от их нормальных значений, оговоренных техническими условиями на средство измерения;
- динамической погрешности, обусловленной инерционностью средства измерений.

Из этих трех составляющих только основная погрешность присуща собственно средству измерений. Две других обусловлены как свойствами средства измерений, так и некоторыми условиями, не связанными с их свойствами. Следовательно, для отражения качества самого средства измерений нельзя пользоваться ни дополнительной, ни динамической погрешностями средства измерений, так как они обуславливаются не только свойствами средств измерений, но и факторами, которые связаны с ними. Поэтому для отражения свойств самого средства измерений надо использовать некоторые характеристики средств измерений, от которых зависят (но не полностью ими определяются) дополнительные и динамические погрешности средств измерений [17].

В инструментальную погрешность измерений, входит не только погрешность средства измерений. Имеются еще составляющие инструментальной погрешности измерений: первая, обусловленная взаимодействием средства измерений с объектом измерений, а также взаимодействием между средствами измерений, включенными последовательно в измерительных си-

стемах и вторая - обусловленная конечной пространственной разрешающей способностью средства измерений.

Модель инструментальной погрешности измерений может быть представлена в следующем виде:

$$\Delta_{in} = \Delta_{MI} * \Delta_{int} * \Delta_{\tau} \quad (1.1)$$

где Δ_{in} - инструментальная погрешность измерений; Δ_{MI} - погрешность средства измерений; Δ_{int} - погрешность, обусловленная взаимодействием средства измерений с объектом измерений; Δ_{τ} - погрешность, обусловленная конечной пространственной разрешающей способностью средства измерений; * - символический знак объединения.

Модель (1.1) может быть представлена следующим образом:

$$\Delta_{in} = \Delta_0(t) * \sum \Delta_{cj} * \Delta_{dyn} * \Delta_{int} * \Delta_{\tau}, \quad (1.2)$$

где $\Delta_0(t)$ - основная погрешность средства измерений; $\sum \Delta_{cj}$ - сумма их дополнительных погрешностей; Δ_{dyn} - динамическая погрешность.

Характеристики составляющих модели (1.2), которые отражают свойства средств измерений, могут быть приняты за их метрологические характеристики (МХ).

При выборе модели основной погрешности средства измерений предполагается, что основная погрешность отдельного экземпляра средств измерений в одной точке диапазона измерений, в соответствии со своими источниками, относится к таким нестационарным случайным процессам $\Delta_0(t)$, которые в общем случае могут быть представлены объединением систематической погрешности $\Delta_{os}(t)$, стационарного центрированного эргодического процесса $\overset{0}{\Delta}_0(t)$, случайных центрированных величин $\overset{0}{\Delta}_{0q}$, $\overset{0}{\Delta}_{0H}$ и случайной центрированной величины $\overset{0}{\Delta}_{dig}$ [17].

$$\Delta_0(t) = \Delta_{os}(t) * \overset{0}{\Delta}_0(t) * \overset{0}{\Delta}_{0q} * \overset{0}{\Delta}_{0H} * \overset{0}{\Delta}_{dig} \quad (1.3)$$

Составляющая $\Delta_{os}(t)$ обусловлена неточностью градуировки и регулировки средства измерения, а также медленными временными изменениями (типа временного старения) свойств материалов и элементов, из которых оно состоит. Составляющая $\overset{0}{\Delta}_0(t)$ – стационарный центрированный эргодический случайный процесс, который представляет собой временные изменения погрешности средства измерения, группирующиеся вокруг систематической погрешности. Эти изменения вызываются случайными временными изменениями свойств материалов и элементов средства измерения, не имеющими определенной направленной тенденции, в том числе внутренними процессами в электронных и полупроводниковых элементах. Составляющая погрешности $\overset{0}{\Delta}_{0q}$ - случайная величина, параметры которой не изменяются во времени и которая, в частности, отражает свойства «белого шума»; $\overset{0}{\Delta}_{0H}$ - случайная составляющая погрешности средств измерений в нормальных условиях, обусловленная гистерезисом, а также люфтом и другими подобными явлениями. Составляющая $\overset{0}{\Delta}_{dig}$ существует только у аналого-цифровых преобразователей (АЦП) и цифровых измерительных приборов (ЦИП). Она обусловлена квантованием измеряемой величины по уровню и полностью определяется номинальной ступенью квантования. Данная составляющая погрешности АЦП и ЦИП не связана с их «инструментальными» свойствами и поэтому не нуждается в экспериментальном оценивании при испытаниях и поверке АЦП и ЦИП.

В систематическую погрешность средства измерений включается погрешность дрейфа, так как выделить на практике постоянную, «чисто» систематическую погрешность средства измерений на фоне дрейфа обычно не представляется возможным. Условность систематической погрешности $\Delta_{os}(t)$ приводит к ее отличию от традиционной систематической погрешности

средств измерений. Обычно под систематической погрешностью понимают известную постоянную или изменяющуюся по известному закону детерминированную величину. Ее оценивают как среднее арифметическое значение определенного количества реализации погрешности. При этом, если систематическая погрешность является функцией времени, то реализации погрешности, усреднением которых оценивают значение систематической погрешности в некоторый момент времени t , должны измеряться в течение такого малого интервала времени Δt , чтобы изменением систематической погрешности за этот интервал времени можно было бы пренебречь.

В основной погрешности, представленной моделью (1.3), вся нестационарность основной погрешности, как случайной функции, и математические ожидания случайных величин отражены систематической погрешностью $\Delta_{0s}(t)$. Остальные составляющие модели (1.3) рассматриваются как стационарный случайный центрированный процесс и центрированные случайные величины.

Практическое использование систематической погрешности как детерминированной величины при технических измерениях невозможно. Учитывать эту погрешность можно только как некоторую случайную величину. На совокупности средств измерений данного типа «систематическая» погрешность является типичной случайной величиной, генеральная совокупность которой определяется совокупностью средств измерений данного типа. Однако, основное, что интересует при технических измерениях, - это методика расчета инструментальной погрешности измерений по известным из нормативно-технических документов МХ средств измерений. Этот расчет можно осуществить, если считать «систематическую» составляющую основной погрешности средства измерений случайной величиной; ее реализации могут находиться в некоторых пределах, которые возможно оценить.

При расчете характеристик инструментальной погрешности технических измерений характеристики составляющей Δ_{0s} в (1.3), как случайной величины, должны объединяться с характеристиками остальных составляющих модели (1.3), как случайного процесса и случайных величин. Другого способа учитывать «систематическую» составляющую основной погрешности средства измерений при расчете характеристик инструментальной погрешности технических измерений нет. Представление основной погрешности средства измерений в виде модели (1.3) позволяет установить относительно простые характеристики свойств средств измерений.

Для средств измерений, частотный диапазон которых охватывает нулевую частоту, наиболее часто вместо «нормального» частотного спектра принимают частоту, равную нулю. Это означает, что основная погрешность определена при измерении величины, не изменяющейся во времени. В таких случаях основная погрешность определена в статическом режиме, то есть является статической. Если средство измерений предназначено для измерений величин, представляющих собой какой-либо параметр или функционал процесса, изменяющегося во времени, то часто в качестве «нормального» - частотного спектра принимают какую-либо определенную частоту гармонического сигнала. В подобных случаях целесообразно называть режим, в котором определена основная погрешность средства измерений, квазистатическим. Тогда считается, что основная погрешность - квазистатическая.

Любое отличие частотного спектра входного сигнала от принятого «нормального» вызывает соответствующую составляющую погрешности средства измерений, и ее следует отнести к той группе, которая выше была названа динамической погрешностью средства измерений. Выделение динамических погрешностей практически целесообразно лишь тогда, когда изменение частотного спектра входного сигнала средства измерений относительно «нормального» спектра вызывает заметное (существенное в конкретных задачах измерений) изменение погрешности средства измерений. Это означает, что для одного и того же средства измерений иногда (при каком-либо частотном спектре входного сигнала) нужно учитывать динамическую погрешность, а иногда (при другом частотном спектре) - это не требуется. Надо ли рассматривать погрешность средства измерений как статическую (квазистатическую) или как динамическую, зависит не только от частотного спектра входного сигнала. Это зависит от соотношения между отличием частотного спектра от «нормального» спектра и инерционностью средства измерений. Именно это соотношение определяет уровень динамической погрешности по отношению к

статической (квазистатической) погрешности. Если для конкретной задачи измерений получаемая разность между ними может не учитываться, то погрешность средства измерений целесообразно признать статической. Инерционность средства измерений (при данном частотном спектре входного сигнала) можно в этом случае также не учитывать. Это существенно упростит анализ инструментальной погрешности измерений при разработке методики выполнения измерения. Если уровень динамической погрешности при данном частотном спектре входного сигнала существенно (для данной задачи измерений) отличается от уровня статической (квазистатической) погрешности, то надо учитывать инерционность средства измерений. В этом заключается условность понятия «динамическая погрешность» средств измерений.

Для таких средств измерений, в которых процессы описываются линейными дифференциальными уравнениями с постоянными коэффициентами (для «линейных» средств измерений), погрешность средства измерений приближенно может быть представлена как аддитивная сумма взаимно независимых статической и динамической погрешностей. Под статической погрешностью здесь понимается погрешность средства измерений при «нормальном» частотном спектре входного сигнала. Под динамической погрешностью - разность между погрешностями средства измерений при реальном и «нормальном» частотных спектрах входного сигнала. Такое представление статических и динамических погрешностей оказывается весьма удобным как для нормирования МХ средств измерений (раздельно можно нормировать характеристики статической основной погрешности и динамические характеристики средств измерений), так и для их применения в метрологической практике [17].

Для объединения составляющих погрешностей средств измерений следует пользоваться первыми двумя моментами их распределений: математическим ожиданием и дисперсией, а также автокорреляционной функцией погрешности средств измерений. Если конечный результат определяется обработкой выходных сигналов не одного и того же, а разных экземпляров средств измерений (независимо от того, одинакового ли они типа или разных), нужно пользоваться не автокорреляционной, а взаимнокорреляционной функцией погрешностей средств измерений. Но взаимнокорреляционная функция не является характеристикой одного средства измерений; она является совместной характеристикой, по крайней мере, двух средств измерений.

Основная погрешность, будучи статической (или квазистатической) погрешностью, сама представляет собой динамический случайный процесс, обладающий соответствующими свойствами и характеристиками.

Метрологические характеристики средств измерений – это характеристики, отражающие их свойства, влияющие на них и позволяющие определять (рассчитывать) результаты и характеристики инструментальных погрешностей измерений. Такое определение соответствует тому, что приведено в [17]. МХ средств измерений, предназначенные для определения результатов измерений (без введения поправок, что обычно при технических измерениях) – это номинальные характеристики: номинальная функция преобразования измерительного преобразователя; номинальная цена деления шкалы показывающего измерительного прибора; вид выходного кода, число разрядов кода, номинальная цена единицы наименьшего разряда кода АЦП и ЦИП.

Инструментальная погрешность измерений в реальных условиях измерений отражается совокупность моделей (1.2) и (1.3). Характеристики составляющих этих моделей, отражающие свойства средств измерений, являются их МХ. Пользуясь ими, можно рассчитывать характеристики инструментальных погрешностей измерений в реальных условиях измерений (располагая информацией об этих условиях).

Первый член $\Delta_0(t)$ модели (1.2), раскрытый моделью (1.3), определяет соответствующую инструментальную погрешность измерений, обусловленную только свойствами самого средства измерений. Остальные четыре составляющие модели (1.2) определяют составляющие инструментальной погрешности измерений, обусловленные как свойствами средства измерений, так и условиями его применения: составляющая $\sum \Delta_{c_j}$ обусловлена чувствительностью средства измерений к влияющим величинам (свойство средства измерений), характером измерений и значениями влияющих величин (не зависят от свойств средства измерений); составляющая

Δ_{dyn} обусловлена свойством инерционности средства измерений (свойство средства измерений) и частотным спектром входного сигнала средства измерений (не зависит от свойств средства измерений); составляющая Δ_{int} обусловлена свойством «входной цепи» средства измерений и свойством «выходной цепи» объекта измерений (не зависит от свойств средств измерений); составляющая Δ_r зависит от пространственной разрешающей способности средства измерений (свойство средства измерений) и от характера поля, параметры которого измеряются (не зависит от свойств средства измерений). Эта составляющая инструментальной погрешности измерений присуща только средствам измерений, предназначенным для измерений параметров полей.

Зависимость некоторых составляющих инструментальной погрешности измерений не только от свойств средств измерений, но и от других факторов, не позволяет характеризовать свойства средств измерений только самими составляющими инструментальной погрешности измерений. Одно и то же средство измерений может применяться в разнообразных условиях. Поэтому его свойства следует отражать так, чтобы его характеристики (МХ средства измерений) могли быть использованы для расчета инструментальных погрешностей измерений в любых реальных условиях применения данного средства измерений.

В качестве МХ средств измерений, предназначенных для использования при расчетах инструментальных погрешностей измерений в реальных условиях измерений, должны быть приняты такие характеристики составляющих модели (1.2), которые отражают только соответствующие свойства самого средства измерений и не зависят от других факторов. Однако знание соответствующих МХ средств измерений необходимо, но недостаточно для расчета инструментальных погрешностей измерений в реальных условиях измерений. Нужно знать и учитывать также характеристики условий применения средств измерений – влияющих величин, частотного спектра входного сигнала средства измерений, «выходных» свойств объекта измерений, поля (для средств измерений, предназначенных для измерений параметров полей).

В метрологической практике наиболее распространен учет воздействия влияющих величин на систематическую составляющую погрешности средств измерений. Воздействием влияющих величин на другие МХ обычно пренебрегают из-за того, что соответствующие изменения других МХ, как правило, достаточно малы по сравнению с самими МХ. Кроме того, учет функций влияния не только на систематическую составляющую погрешности средства измерений без заметной пользы обычно существенно усложнил бы как нормирование и оценивание функций влияния, так и расчет характеристик инструментальных погрешностей измерений. Изменения систематической погрешности средств измерений, вызванные воздействием изменений влияющих величин, называются дополнительными погрешностями. Характеристиками средств измерений, отражающими дополнительные погрешности, служат несколько (в зависимости от числа существенно воздействующих влияющих величин) функций влияния на систематическую погрешность. Поэтому в уравнении (1.2) показана сумма нескольких дополнительных погрешностей, и поэтому следует учитывать, в общем случае, несколько функций влияния.

Любая МХ, в общем случае, может быть функцией измеряемой величины в том смысле, что может изменяться (принимать разные значения) в диапазоне измерений средств измерений. Поэтому часто МХ оцениваются не в одной, а в нескольких точках диапазона измерений, в зависимости от вида функций МХ в диапазоне измерений данного средства измерений.

Основной операцией определения метрологической исправности и, следовательно, допустимости применения средств технических измерений служит контроль МХ. Средство измерений должно признаваться метрологически исправным, если все его нормированные МХ (НМХ) во всех точках диапазона измерений удовлетворяют нормам, регламентированным в технических условиях на средства измерений данного типа. Однако на практике невозможно каждый раз контролировать все его НМХ во всех точках диапазона измерений, когда требуется установить допустимость применения тех или иных средств измерений.

Все НМХ средств измерений (исключая номинальные характеристики, такие как номинальная функция преобразования, номинальная цена деления шкалы и т.п., которые, не контро-

лируются) можно разделить на две группы. НМХ первой группы контролируются у каждого экземпляра средств измерений как при выпуске из производства, так и периодически в процессе эксплуатации. НМХ второй группы контролируются периодически на заводе-изготовителе у некоторой выборки средств измерений данного типа, при контрольных испытаниях. К первой группе относятся, главным образом, характеристики основной погрешности, контролируемые при первичной и периодических поверках.

Поверка – это установление пригодности средства измерения к применению на основании экспериментально определенных метрологических характеристик и контроля их соответствия установленным требованиям. Ко второй группе относятся такие НМХ, как функция влияния или другие нормированные характеристики чувствительности средств измерений к влияющим величинам, динамические характеристики и др.

Характеристики основной погрешности разных средств измерений одного и того же типа могут значительно различаться и существенно изменяться во времени. Другие МХ более стабильны как на множестве средств измерений одного и того же типа, так и во времени. Кроме того, на большинство МХ второй группы оказывают влияние те же элементы средств измерений, которые влияют и на основную погрешность.

Тот факт, что контроль основной погрешности представляет собой массовую операцию, которой периодически подвергаются все средства измерений, а контроль других МХ – это операция, которой подвергается малое количество средств измерений на заводе-изготовителе, причем редко (только при контрольных испытаниях) позволяет использовать разные подходы к определению качества, достоверности контроля этих двух групп МХ.

В качестве критериев достоверности контроля (т.е. соответствия результатов контроля его цели) характеристик основной погрешности используются те же показатели, которыми пользуются в теории контроля изделий – вероятностными характеристиками ошибок контроля. Такой подход в данном случае вполне естественен, т. к. средства измерений – это тоже изделия. При контроле основной погрешности результат контроля признается положительным, и средство измерений – исправным, если основная погрешность соответствует заданной норме во всех точках диапазона измерений, вернее, во всем диапазоне измерений. Если хотя бы в одной точке диапазона измерений основная погрешность не удовлетворяет норме, средство измерений признается дефектным, по результатам контроля.

Метрологическое обеспечение МСКУ. Каналы приема и формирования аналоговых сигналов МСКУ являются метрологическими. Описание блоков, входящих в эти каналы, приведено в приложении Д «Характеристики блоков МСКУ 2». В приложении В рассмотрены некоторые вопросы первичной обработки значений сигналов: методы фильтрации, линеаризации измеренных значений сигналов термопар и термопреобразователей сопротивления, масштабирования измерений.

Метрологические характеристики имеют следующие каналы связи с объектом: приема аналоговых сигналов напряжения среднего и низкого уровней, тока и сопротивления; приема сигналов переменного тока, приема частотных и время - импульсных сигналов; формирования аналоговых сигналов тока и напряжения; формирования время-импульсных сигналов.

Для уменьшения систематической погрешности результаты измерения подвергаются *цифровой коррекции*. Суть коррекции заключается в вычислении поправочных коэффициентов, определяющих смещение нуля и угла наклона характеристики преобразования. Результаты преобразования корректируются программой калибровки каналов ввода аналоговых сигналов. В процессе работы МСКУ программа калибровки каналов с объектом периодически подключает на входы этих каналов эталонные сигналы. В качестве источника эталонных сигналов используются имитаторы сигналов, которые по программному запросу выдают фиксированные значения сигналов, находящиеся в любой точке диапазона изменения сигнала. По результатам измерения эталонных сигналов производится вычисление поправочных коэффициентов и коррекция результатов измерения сигналов реальных датчиков.

Проверка метрологических характеристик отдельных каналов ввода-вывода аналоговых сигналов проводится в соответствии с графиком поверок измерительных приборов и устройств предприятия.

В составе программного обеспечения МСКУ есть программы для автоматического контроля и оценки метрологических характеристик. В качестве нормируемых характеристик погрешностей измерительных каналов МСКУ установлены:

- пределы допускаемой основной приведенной погрешности;
- пределы допускаемой систематической составляющей основной погрешности;
- пределы допускаемого среднего квадратичного отклонения случайной составляющей основной погрешности;
- пределы допускаемой приведенной погрешности в интервале изменения температуры и относительной влажности при рабочих условиях эксплуатации.

Периодичность поверок установлена не реже одного раза в год. При проверке МСКУ по контрольной задаче проверяемые каналы связи с объектом должны быть отключены от объекта. Для реализации проверки используется коммутационная панель, при помощи которой обеспечивается подключение внешних источников-имитаторов дискретных сигналов постоянного и переменного токов.

В качестве примера рассмотрим методику определения систематической составляющей основной погрешности каналов приема сигналов напряжения среднего уровня.

Пример. Определить систематическую составляющую основной погрешности каналов приема сигналов напряжения среднего уровня для МСКУ.

Решение. Принимаем, что данная методика определения систематической составляющей Δ_s исключает влияние на нее случайной составляющей погрешности. С этой целью в окрестности исследуемой точки (рис. 1.9) x_i находится такая точка x'_i , в которой случайная составляющая погрешности распределена симметрично и x'_i является медианой распределения [18].

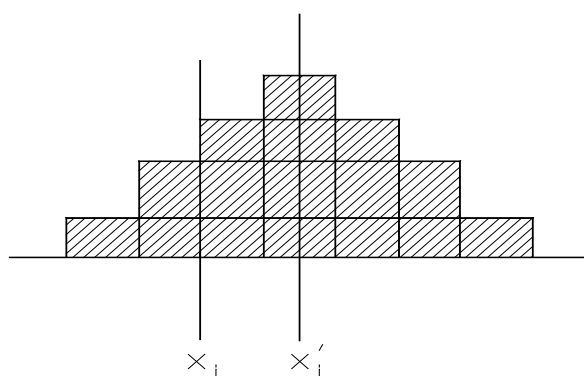


Рис. 1.9- Диаграмма распределения.

Пусть в точке x'_i результат измерения, приведенный к входу, равен x''_i . Тогда систематическая погрешность Δ_i в точке равна:

$$\Delta_i = x''_i - x'_i \quad (1.4)$$

Сигнал на входе в канал x_i устанавливается вручную. Схема измерений систематической погрешности на каналах приема сигналов напряжения среднего уровня показана на рис. 1.10.

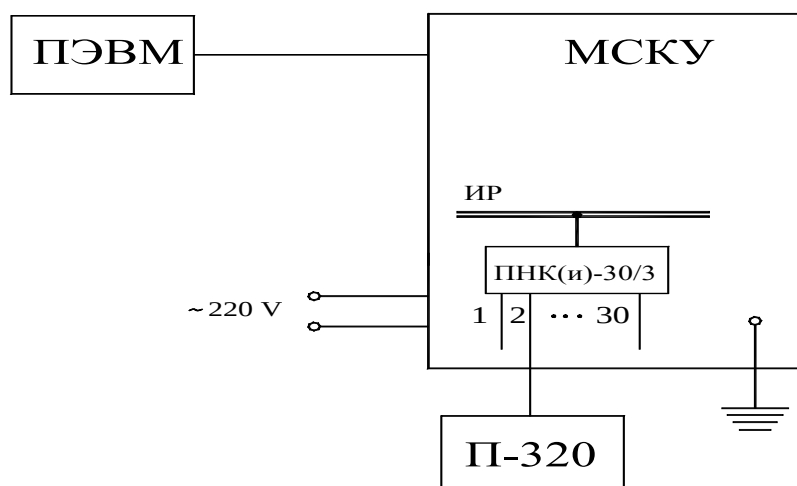


Рис. 1.10. Схема измерения систематической погрешности.

На рис. 1.10 обозначено: П-320 – программируемый калибратор, который устанавливает значение напряжения на входе ИК; ПНК(И)-30/3 – преобразователь напряжения; ИР – внутренний интерфейс МКСУ.

В точке x_i выполняется n измерений установленного на входе в канал сигнала. Результаты измерений – принятые коды АЦП – преобразования упорядочиваются согласно их величине. Общее количество интервалов, в которые может попасть конкретный результат измерения, ограничено $K_i \pm 20$ квантов (рис. 1.11).

На рис. 1.11 K_i – код, соответствующий сигналу x_i :

$$\kappa_i = 32000 \cdot \frac{U_{\text{вх}}(mV)}{U_{\text{max}}(mV)} \quad (1.5)$$

где $U_{\text{вх}}$ – напряжение на входе в канал; U_{max} – максимальное значение диапазона напряжений входного сигнала.

В случае если сумма попаданий кодов K_i в интервал $[K_i - 20; K_i - 1]$ не равна сумме попаданий кодов K_i в интервал $[K_i + 1; K_i + 20]$ т.е.:

$$\sum_{i=20}^{i-1} Cr_i \neq \sum_{i+1}^{i+20} Cr_i \quad (1.6)$$

то входной сигнал x_i смещается в точку x'_i в сторону большей суммы. В новой точке x'_i опять выполняется n измерений входного сигнала. Процесс поиска точки x'_i заканчивается, если

$$\sum_{i=20}^{i-1} Cr_i = \sum_{i+1}^{i+20} Cr_i \quad (1.7)$$

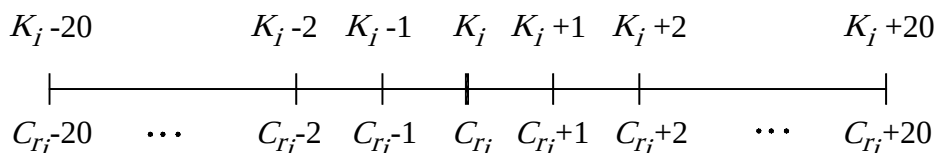


Рис. 1.11. Распределение результатов измерений.

В качестве систематической погрешности Δ_s в точке x_i принимаем разность между сигналом образцового источника x_i и полученным значением x'_i

$$\Delta_s = x_i - x'_i \quad (1.8)$$

Количество измеряемых точек x_i в диапазоне изменения входного сигнала $i \approx 20$. Погрешность (основная приведенная): $\Delta_{op} = \pm 0,1\%$.

1.4 Создание ПТК

Весь процесс создания ПТК делится на ряд стадий, установленных стандартами. Предусмотрены две предпроектные стадии (*Технико-экономическое обоснование (ТЭО)* и *Техническое задание*) и две проектные (*Технический проект* и *Рабочий проект*), которые допускается объединять в одну стадию - *Технорабочий проект*, а также стадии *Ввод в действие (внедрение)* и *Анализ функционирования*. Каждая стадия подразделяется на этапы.

Предпроектные стадии. *Технико-экономическое обоснование (ТЭО)* при создании ПТК является основной целью работ, выполняемых на этой стадии. Разработка ТЭО проводится Заказчиком системы совместно с генеральной проектной организацией и разработчиком ПТК. На

этом этапе определяются главные источники ожидаемой эффективности создаваемого ПТК. Проводится изучение и анализ действующего технологического объекта и существующего ПТК, определяются существующие недостатки управления и устанавливаются причины этих недостатков. Основные результаты работ этого этапа оформляются в виде законченного документа, в котором изложены исходные технические требования к ПТК и приведен предварительный расчет ожидаемых показателей технико-экономической эффективности от внедрения.

Разработка технического задания (ТЗ). Основная цель данной стадии заключается в создании документа, в соответствии с которым проводится разработка и осуществляется проверка ПТК при его передаче в промышленную эксплуатацию. Ответственность за разработку ТЗ несет исполнитель - разработчик ПТК. ТЗ на создание ПТК, должно содержать следующие основные разделы: назначение ПТК; требования к видам обеспечения ПТК; состав и содержание работ по созданию ПТК; порядок сдачи ПТК в эксплуатацию.

Проектные стадии. Целью *технического проекта* является обоснование и разработка основных технических решений по создаваемому ПТК и определение его сметной стоимости. Исходными документами для разработки техпроекта являются:

- утвержденное ТЗ на создание ПТК;
- ТЭО системы;
- исходные данные Заказчика о технологическом объекте.

На стадии «Технический проект» проводятся необходимые исследовательские и проектные работы по следующим основным этапам:

1. *Системотехнический синтез ПТК.* На этом этапе прорабатываются основные решения по системе в целом (структурная и функциональная схемы ПТК, описания функциональных задач и др.). На этом же этапе составляются задания на работы по проектированию ПТК.

2. *Аппаратурно-технический синтез.* Посвящен определению характеристик информационно-измерительных каналов, уточнению структуры комплекса технических средств и функциональных схем автоматизации и т.п.

3. *Техническое проектирование* всех видов обеспечения ПТК. Осуществляется разработка всех алгоритмов функционирования ПТК и завершается выпуском задания на программирование.

Документация технического проекта разделяется на *общесистемную, информационного и технического обеспечения.*

Рабочий проект. Целью работ является выработка рабочих решений по создаваемой системе с выпуском проектно-сметной документации, документации на техническое и программное обеспечения, необходимой для изготовления, наладки и эксплуатации системы. Основанием для начала работ служит утвержденный технический проект ПТК. Результатом рабочего проекта являются выпуск документации.

Стадии реализации. *Вводом в действие* является физическая реализация системы и передача ее в промышленную эксплуатацию. Основные этапы работ: изготовление КТС; наладка и испытание ПТК; внедрение и опытная эксплуатация; приемо-сдаточные испытания.

Анализ функционирования. Цель работ заключается в получение объективных данных о качестве ПТК, текущем состоянии и реальном эффекте от использования ПТК на основании опыта ее промышленной эксплуатации. Основные этапы работ: предварительное обследование состояния ПТК; экспериментально-статистические исследования; анализ полученных результатов.

Эксплуатация ПТК включает обеспечение функционирования и дальнейшую модернизацию ПТК.

Вопросы

1. Дайте определение ПТК.
2. Какие функции может выполнять ПТК?
3. В каком виде информация об объекте вводится в ПТК?

4. Для чего в ПТК нужны средства отображения информации?
5. Какое место занимает ПТК в АСУ ТП?
6. Каковы достоинства распределённой структуры ПТК?
7. Какие функции выполняются на нижнем (верхнем) уровне АСУ ТП?
8. Какие функциональные подсистемы должны входить в ПТК?
9. Какие информационные функции выполняет ПТК?
10. Что входит в первичную обработку информации?
11. Как производится контроль достоверности входных сигналов?
12. Для чего необходима гальваническая развязка каналов ввода/вывода?
13. Какие вы знаете виды уставок, которые обычно используются в ПТК АСУ ТП?
14. Что является источниками дискретных сигналов?
15. Какие существуют типы дискретных сигналов?
16. Каким образом может быть реализован ввод дискретных сигналов в ПТК?
17. В чем состоит различие в обработке пассивных и инициативных сигналов?
18. Как выполняется сбор и обработка кодированных сигналов?
19. В каком виде отображается информация на операторской станции?
20. Что такое мнемосхема?
21. Что входит в пакет видеограмм?
22. Какие существуют виды сигнализаций?
23. Что используется для представления динамики изменения параметров?
24. Какая технологическая информация подлежит архивации?
25. Что является целью расчета технико-экономических показателей оборудования?
26. Что относится к управляющим функциям?
27. Для чего предназначено дистанционное управление?
28. Как реализуется дистанционное управление?
29. Какие виды регулирования могут быть реализованы в ПТК?
30. Для чего предназначено автоматическое логическое управление?
31. Для чего предназначены инструментальные средства проектирования и наладки алгоритмов управления?
32. На чем основаны алгоритмы блокировок?
33. Для чего предназначены технологические защиты?
34. В чем состоят особенности задач, реализующих функции технологических защит?
35. Что входит в состав сервисных функций ПТК?
36. Какие существуют виды обеспечения ПТК?
37. В чем состоит понятие «лингвистическое обеспечение ПТК» ?
38. Какие средства включает лингвистическое обеспечение?
39. Что входит в понятие «информационное обеспечение ПТК» ?
40. В чем заключаются основные принципы организации информационного обеспечения?
41. Что является задачами информационного обеспечения?
42. Что входит в понятие «метрологического обеспечения ПТК» ?
43. В чем заключаются организационно-технические мероприятия по метрологическому обеспечению?

ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПТК

Техническое обеспечение - это комплекс технических средств (КТС), применяемых для построения АСУ ТП. В состав КТС ПТК для АСУ ТП входят: технические средства нижнего уровня – контроллеры, устройства связи с объектом; средства передачи информации; технические средства верхнего уровня - операторское оборудование, информационно-вычислительный комплекс, инженерные станции; сервисные средства для эксплуатации АСУ ТП.

2.1 Контроллеры

Функции нижнего уровня АСУ ТП выполняют контроллеры. Количество и состав контроллеров для реализации конкретной АСУ ТП выбираются исходя из оптимального сочетания выполнения требований быстродействия, надежности и стоимости. Контроллеры различаются между собой по степени быстродействия, объему оперативной памяти, количеству каналов ввода-вывода, интерфейсу, надежностным характеристикам, реализуемому математическому обеспечению.

Контроллеры должны иметь модульную структуру с целью максимального облегчения наладки, обслуживания и обучения персонала. Из отечественных контроллеров наиболее известны контроллеры семейства МСКУ М и Ремиконт (Ломиконт).

2.1.1 Программируемые логические контроллеры

Первый программируемый контроллер (ПЛК), был разработан по требованию компании General Motors, пожелавшей иметь устройство, которое обладало бы гибкостью компьютера и могли бы программировать и обслуживать заводские инженеры и техники. В прошлом столетии средства автоматики технологических процессов строились в основном на пневматических, электромеханических и электронных устройствах (реле, командоаппаратах) и созданных на этой базе ПИД-регуляторах. Реле были самые разнообразные, в том числе, поляризованные-двухстабильные, которые имели две катушки управления для включения и выключения реле, соответственно, и реле с фиксированными (регулируемыми) временами включения/выключения. В основу электромеханических командоаппаратов был положен вращающийся со стабильной угловой скоростью барабан с установленными на нем в определенных местах штырями, которые вызывали включение неподвижных по отношению к барабану контактов, когда штырь (благодаря вращению барабана) оказывался под одним из контактов. Схемы, построенные на конденсаторах, могли быть с обратными связями, то есть вращение барабана приостанавливалось до наступления ожидаемого события, образуя, таким образом, шагово-логический процесс управления. Автоматика на базе электромеханических устройств применяется достаточно широко и сегодня. Примером могут служить многочисленные бытовые стиральные машины.

В 60-х годах прошлого столетия начали появляться электронные реле, счетчики, таймеры, используемые для построения схем автоматики, реализующих шагово-логические процессы управления. Схемы автоматики строились в виде физически связанных разнородных обособленных устройств (реле, счетчики, таймеры, командоаппараты, регуляторы и т.п.). У такого подхода было, как минимум, два недостатка: во-первых, аппаратура автоматики получалась довольно громоздкой, так как, по существу, каждой логической функции соответствовало отдельное физическое устройство (счетчик, таймер, схема логического преобразования и т.п.) и, во-вторых, любое изменение алгоритма управления объектом требовало физического изменения аппаратуры (схемы взаимосвязей между аппаратными элементами) автоматики, то есть отсут-

ствовала гибкость. Выход был в применении программируемых устройств. Существовавшие в те годы большие ЭВМ располагались на площадях в десятки квадратных метров, потребляли до 100 kW мощности и, самое главное, отказывали по несколько раз в сутки. Поэтому никогда не возникала мысль использовать большую ЭВМ в контуре управления технологическим процессом. Правда, к концу 60-х годов появились мини-ЭВМ фирм HP (HP21MX) и DEC (PDP-11), которые послужили прообразами для отечественной мини-ЭВМ М-6000 и последующего семейства малых ЭВМ (СМ-1, СМ-2, СМ-3, СМ-4). На базе мини-ЭВМ были построены первые примеры контуров непосредственного цифрового управления (НЦУ) технологическими процессами и была доказана возможность реализации на средствах вычислительной техники любых законов дискретного (шагово-логические процессы и, как их подмножество, защиты и блокировки) и непрерывного (регулирование) управления процессами. Однако мини-ЭВМ задумывались и реализовывались как «персональные компьютеры». По своим габаритам, а главное надежностным показателям и условиям эксплуатации они были ориентированы для работы скорее в «тепличных» условиях вычислительных центров, а не промышленном производстве. Поэтому мини-ЭВМ применялись в АСУ ТП очень широко, но только для информационных функций. Собственно, термин «автоматизированная» система родился именно тогда. Это система, которой не доверяется непосредственное управление объектом, но в ее контур включен человек-оператор, которого она информирует о состоянии объекта и которому доверяется принимать и реализовывать управленческие решения.

Делались попытки создания специализированных промышленных контроллеров, построенных по принципу «малых ЭВМ». Однако из-за низкой степени интеграции элементной базы и примитивности дешевого периферийного оборудования они получались хоть и достаточно надежными, но предельно неудобными в эксплуатации. Так, например, известны малые ЭВМ для народного хозяйства, в которые программа заносилась посредством прошивки магнитного запоминающего устройства. То есть, для каждого слова команды физически прокладывался (прошивался) через магнитные тороидальные сердечники отдельный провод. Команд в программе управления объектом могло быть до тысяч. Модернизировать такую программу силами специалистов по автоматизации промышленных объектов не представлялось возможным. К тому же, как было показано выше, на промышленных объектах они работали со схемами автоматики, составленными из отдельных функциональных узлов (приборов). Для перехода к использованию гибких устройств автоматики на базе ЭВМ, программируемых на Ассемблере и даже универсальных языках программирования, образовался труднопреодолимый психологический барьер.

В конце 60-х в начале 70-х годов того же столетия произошли два события, которые открыли путь средствам вычислительной техники в управление технологическими процессами: появился первый функционально-полный микропроцессор Intel 8080 с набором периферийных микросхем, достаточных для построения надежного малогабаритного и достаточно производительного вычислителя, в котором аппаратные элементы автоматики как счетчики, таймеры, командоаппараты, регуляторы и т.п. могли быть представлены информационными образами в памяти. Появился язык технологического программирования, в основу которого были положены алгоблоки в виде понятных каждому инженеру-автоматчику регистров, триггеров, счетчиков, таймеров, коммутаторов, регуляторов, каналов ввода-вывода, модулей логических преобразований. Взаимосвязи алгоблоков в программе были очень похожи на структурные схемы узлов автоматики в проектной документации объектов управления. Более того, прошло всего несколько лет, и проектировщики стали изображать схемы автоматики в терминах алгоблоков. Вскоре на базе микропроцессоров и языка технологического программирования появились программируемые контроллеры, занимавшие меньше места, чем реле, счетчики, таймеры и прочие средства контроля и имевшие гораздо большую гибкость с точки зрения возможностей перепрограммирования.

Каждая фирма, производящая программируемые контроллеры, считала своим долгом внести лепту в развитие языка технологического программирования. К началу восьмидесятых годов их образовалось множество, причем сильно отличающихся друг от друга. Международной

Электротехнической Комиссией (МЭК) был выпущен стандарт IEC 1131 на базовые элементы языка технологического программирования (см. главу 3). Первые ПЛК в силу малой производительности используемых процессоров, оснащались примитивными средствами связи с оперативным персоналом (цифровой, очень редко алфавитно-цифровой индикатор, несколько кнопок управления), что для малого количества каналов ввода-вывода и объема решаемой задачи управления объектом было вполне достаточно.

С ростом производительности микропроцессоров объем функций управления, поручаемых ПЛК, стал расширяться. Потребовалось ввести возможности архивирования данных и обеспечить связь с оперативным персоналом посредством мнемосхем и графической информации. Технически это поддерживалось матричными индикаторами, приспособленными для жестких промышленных условий эксплуатации, которые позднее стали оснащаться сенсорными экранами. В ПЛК увеличилось количество каналов ввода-вывода, соответственно он смог управлять многими контурами с различными циклами управления. Расширилась номенклатура интеллектуального периферийного (в том числе и связи с объектом) оборудования. Это потребовало организации многозадачного режима работы и файловой системы в программном обеспечении.

Для того чтобы ПЛК удовлетворял указанным возможностям, необходимо было превратить узкоспециализированный на работу с каналами связи процессор ПЛК в универсальный. Таким требованиям удовлетворяла IBM PC, однако особенности ее конструктивной реализации, предназначенной для конкретных условий эксплуатации (например, максимальная рабочая температура $\leq 35^{\circ}\text{C}$, влажность воздуха $< 80\%$, вибрация $< 0,1\text{ g}$, а в промышленности максимальная температура эксплуатации – 50°C , влажность воздуха – 95% , вибрация – 1 g), не позволяли использовать ее в промышленности. В решении этой проблемы большую роль сыграл бурный рост переносных компьютеров, который потребовал от микроэлектроники создания элементной базы с низким тепловыделением и виброустойчивых дисковых накопителей, энергонезависимой полупроводниковой памяти. Здесь необходимо упомянуть, что в 2000 г. объем реализации IBM PC компьютеров превышал 20 млрд. долларов США, а объем реализации ПЛК не достигал 1 млрд. долларов. Поэтому разработка микроэлектронных компонентов специально для ПЛК экономически не эффективна. Кроме того, с момента появления первых IBM PC делались попытки включить их в контур управления объектом. Однако ни MS DOS, ни тем более Windows для этих целей не подходили. Были разработаны операционные системы реального времени для платформы IBM PC (см. главу 3). Так появились ПЛК с архитектурой IBM PC.

Производимые сегодня ПЛК делятся на два основных типа: неизменяемые и модульные. Неизменяемые ПЛК поставляются как самостоятельные устройства с процессором, источником питания и фиксированным количеством каналов дискретного и аналогового ввода/вывода. Неизменяемые контроллеры могут иметь отдельные взаимосвязанные компоненты для расширения, обладают меньшими размерами, они дешевле и проще в установке. Модульные же контроллеры обладают большей гибкостью, имеют широкие возможности по вводу/выводу, объемам процессорной памяти, входным напряжениям, типу и количеству каналов связи и т.д. Первоначально ПЛК использовались в управляющих системах с дискретным вводом/выводом. С течением времени эти контроллеры приобрели новые возможности, что позволило их использовать для систем управления непрерывными технологическими процессами.

Развитие программируемого управления привело к появлению микроконтроллеров (микро-ПЛК), которые могли применяться и в тех системах, традиционно строящихся на основе других методов. Микроконтроллеры – это, как правило, неизменяемые контроллеры с 10-32 каналами ввода/вывода на один блок, способные заменить даже самые маленькие релейные панели. Кроме того, благодаря их дешевизне они могут использоваться даже в небольших устройствах (к примеру, в автономных регуляторах).

Современные ПЛК обрабатывают дискретные и аналоговые сигналы, поступающие от датчиков, конечных выключателей, пультов оператора и других устройств ввода в соответствии с заданными алгоритмами. Формируют дискретные и аналоговые выходные сигналы для исполнительных механизмов, таких, например, как соленоиды, приводы клапанов, задвижки и др.

Во многих случаях технологический процесс не нуждается в большой вычислительной мощности системы управления. Ядром аппаратуры управления вполне может быть процессор, поддерживающий адресацию к памяти размером до 1024 КВ, что достаточно для размещения более 20000 строк управляющей программы и хранения большого объема данных. ПЛК используются в системах, где необходимо обеспечить надежное управление в реальном времени. В настоящее время существуют сотни моделей ПЛК. Среди них и "сверхлегкие" - узкоспециализированные, миниатюрные контроллеры, и "сверхтяжелые" модели - универсальные, на базе 32-разрядных процессоров, с числом каналов ввода/вывода 1024 и более. Среди иностранных наиболее распространены ПЛК фирм Allen-Bredley, Siemens, ABB, Schneider, а среди отечественных - Ломи-конт, Ремиконт, Эмикон, Ш-711 и др. Рассмотрим особенности наиболее представительных ПЛК.

ПЛК Modicon фирмы Schneider (рис. 2.1.) является младшей моделью в ряду контроллеров, выпускаемых компанией, [19]. Простейший тип Nano имеет 10 дискретных входов-выходов: 6 входов на 24 V постоянного тока и 4 релейных или транзисторных выхода, 0,5 А. Габариты этого контроллера - 105x85x60 mm, питание от сети переменного тока (110-240 V) или от источника постоянного тока 24 V. Несмотря на свои малые размеры, этот ПЛК обладает хорошими техническими характеристиками по быстродействию, достаточной пользовательской памятью для программ и данных, а также "дружественными" средствами программирования, отладки и настройки. Так, время сканирования одной тысячи булевских команд программы



пользователя (с учетом накладных расходов на организацию сканирования) составляет 1,40 ms. Контроллер имеет встроенный счетчик (10 kHz) и импульсный выход, к нему можно подключать интеллектуальный пульт оператора, а также включать его в сеть вместе с другими контроллерами для организации обменов данными.

Рис.2.1. Внешний вид контроллера Nano.

Программирование и отладка программ могут быть выполнены как с помощью ручного терминала-программатора (в этом случае языком программирования является язык "Instruction List" - IL) так и с помощью IBM-совместимого ПК (в этом случае пользователь может выбрать любой из двух языков - язык IL или графический язык релейно-контактных схем "Ladder Diagrams" - LD).

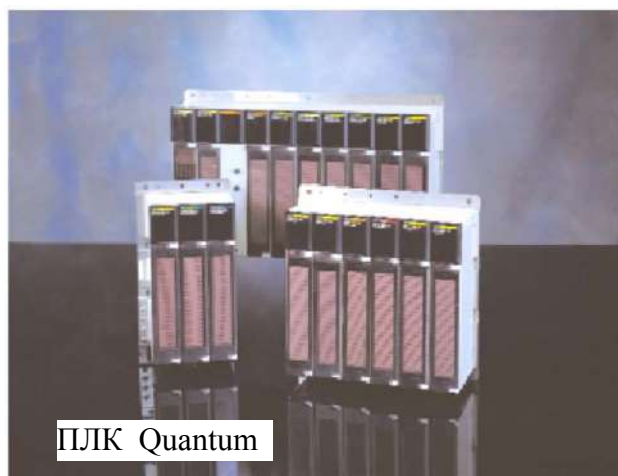


Рис.2.2. Внешний вид контроллера Quantum.

Контроллер Modicon TSX Quantum (рис. 2.2) является одним из самых мощных контроллеров, выпускаемых для целей автоматизации. Имеется свыше 40 типов моделей ввода-вывода и модулей специального назначения (быстрого счета, коммуникации, горячего резервирования и др.), которые можно устанавливать на монтажных панелях серии Quantum. Кроме того, имеется около 60 типов модулей предыдущего поколения (так называемой серии 800), которыми можно управлять со стороны процессора

Quantum, используя систему удаленного ввода-вывода.

Старшая модель контроллера имеет ОЗУ 2 МВ и обеспечивает сканирование одной тысячи булевских команд программы пользователя за 0,1 ms. Локальная система ввода-вывода кон-

троллера может обеспечивать работу с 448 дискретными входами-выходами, а система удаленного ввода-вывода контроллера позволяет увеличить потенциально это число до 64 тысяч.

Конструкция программируемых контроллеров Modicon TSX Quantum позволяет экономить пространство в щите. Обладая глубиной всего в 4 дюйма (включая экран), для этих контроллеров не требуются большие щиты, они размещаются в стандартном 6-дюймовом электрическом шкафу, что позволяет экономить до 50 % стоимости обычных панелей управления. Они могут легко устанавливаться на задних панелях шкафов класса NEMA или в 19-дюймовых стойках. Несмотря на малые размеры, контроллеры Quantum поддерживают высокий уровень производительности и надежности. Возможность подключения к сетям предприятия и полевым шинам реализована для восьми типов сетей от Ethernet до INTERBUS-S. Контроллер Quantum совместим с более старыми сериями модулей ввода/вывода использующими сеть удаленного ввода/вывода Modicon S908.

Инструментальные средства разработки приложений являются общими для контроллеров. Пакет программ ConSert устанавливается под управлением MS Windows и позволяет работать на всех пяти языках по стандарту IEC 1131. Отличительной чертой этого пакета является наличие программы, моделирующей контроллер со всеми его модулями (эмулятор аппаратных средств). Это позволяет выполнять большую часть работы по отладке приложения в автономном режиме, без подключения к контроллеру.

Программируемые логические контроллеры семейства SIMATIC (внешний вид модулей контроллеров SIMATIC показан на рис. 2.3). Они способны работать в реальном масштабе времени и могут быть использованы как для построения узлов локальной автоматики, так и распределенных систем с организацией обмена данными по PPI- или MPI-интерфейсу, сети PROFIBUS DP или AS-I [20]. Семейство SIMATIC S7-200 объединяет в своем составе 4 типа центральных процессоров, отличающихся объёмом памяти, количеством встроенных входов/выходов, набором встроенных функций, возможностями расширения системы, имеет широкий спектр модулей ввода/вывода дискретных и аналоговых сигналов и два коммуникационных модуля, обеспечивающих возможность подключения к AS-интерфейсу и сети PROFIBUS-DP (только ведомое устройство).

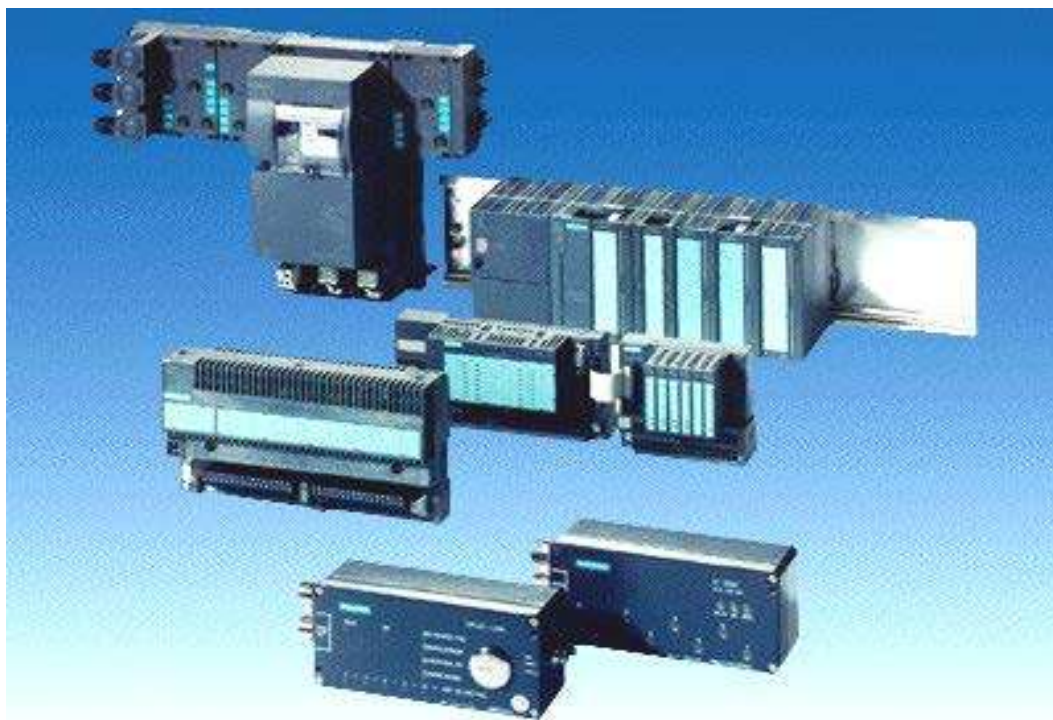


Рис. 2.3. Внешний вид модулей контроллеров SIMATIC.

Отличительные особенности заключаются в том, что время выполнения тысячи логических инструкций не более $0,37\text{ ms}$, имеет скоростные счетчики внешних событий, быстродействующие входы внешних прерываний и возможность наращивания количества обслуживаемых входов/выходов (за исключением CPU 221). Кроме того, имеются импульсные выходы, потенциометры аналогового задания параметров, встроенные часы реального времени, мощный набор инструкций языка программирования, последовательный PPI-интерфейс, который может быть использован в качестве свободно программируемого интерфейса, функции ведущего устройства AS-интерфейса, обеспечиваемые коммуникационным модулем CP 243-2, функции ведомого устройства PROFIBUS-DP, обеспечиваемые коммуникационным модулем EM 277, дружественная оболочка программирования STEP 7 Micro/Win, трехуровневая парольная защита программ пользователя, возможность работы с устройствами человеко-машинного интерфейса. Встроенный коммуникационный порт модулей ЦП с интерфейсом RS-485 может функционировать в трех режимах:

- сканирующий PPI-интерфейс для программирования контроллера и подключения программатора, компьютера (PC/PPI-кабель), текстового дисплея TD 200 или панели оператора, связи с другим S7-200. Скорость передачи данных до $187,5\text{ Kb/s}$;

- MPI-интерфейс (ведомое устройство) для обмена данными с S7-300, S7-400, панелями оператора, текстовыми дисплеями, кнопочными панелями. Связь между S7 200 невозможна. Скорость передачи данных до $187,5\text{ Kb/s}$;

- скоростной свободно программируемый интерфейс, обеспечивающий реализацию необходимых протоколов связи (ASCII, ModBus и т.д.). Скорость передачи данных может регулироваться и достигать $38,4\text{ Kb/s}$.

С помощью коммуникационного модуля CP 277 все модули ЦП, за исключением CPU 221, могут подключаться к сети PROFIBUS-DP, выполняя при этом только функции ведомого устройства. Модули ЦП серии S7-200, за исключением CPU 221, способны выполнять функции ведущего устройства AS-интерфейса. Подключение к AS-интерфейсу производится с помощью коммуникационных модулей CP 243-2. К каждому контроллеру может быть подключено до 31 ведомого устройства AS-интерфейса, позволяющего обслуживать до 248 дискретных сигналов.

Программирование ПЛК SIMATIC S7-200 осуществляется с помощью пакета STEP 7 Micro/Win (V3.0 и выше), исполняемого под управлением ОС Windows. В пакете реализована поддержка языков LAD (релейно-контактные схемы), STL (список инструкций) и FBD (функциональных блочных диаграмм), соответствующих DIN EN 61131-3. Связь компьютера с программируемым ЦП осуществляется через PC/PPI-кабель. STEP 7 Micro/Win V3.1 и выше позволяет выполнять все операции по программированию контроллеров SIMATIC S7-200, их конфигурированию и параметрированию, а также решать вопросы конфигурирования и программирования PPI-сетей, устройств человеко-машинного интерфейса (TD 200 и TP 070), систем регулирования, обеспечивает поддержку USS-протокола.

Контроллеры SIMATIC S7-200 имеют модульную конструкцию. Модули расширения ввода/вывода и коммуникационные процессоры имеют тот же дизайн, что и центральные процессоры. Модули ввода-вывода имеют несколько исполнений, позволяющих обслуживать входы/выходы с различными параметрами электрических сигналов. Монтаж контроллеров может осуществляться на 35 mm профильную DIN-шину или на плоскую поверхность с креплением винтами. Степень защиты IP20.

Операторский интерфейс (рис. 2.4). Текстовый дисплей TD 200 может быть использован со всеми контроллерами семейства SIMATIC S7-200. Дисплей подключается к контроллеру соединительным кабелем через PPI-интерфейс. По этому же кабелю осуществляется питание дисплея. К одному контроллеру может быть подключено несколько дисплеев TD 200. Рабочий диапазон температур от 0 до $+60\text{ }^{\circ}\text{C}$. Степень защиты IP65.

Дисплей позволяет отображать до 80 текстовых сообщений и до 4 переменных (2 строки по 20 символов). Сообщения могут формироваться на немецком, английском, французском, испанском, итальянском и русском языках. При необходимости информация может представляться в виде столбиковых диаграмм, сопровождаться обработкой чисел с плавающей запятой и

блоков данных для нескольких дисплеев TD 200. Дисплей позволяет использовать встроенное меню и парольную защиту программ пользователя. Функции клавиш задаются программно. Программирование дисплея TD 200 производится с помощью пакета STEP 7 Micro/Win или STEP 7 Micro/DOS. Технические характеристики ПЛК SIMATIC S7-200 приведены в табл. 2.1.

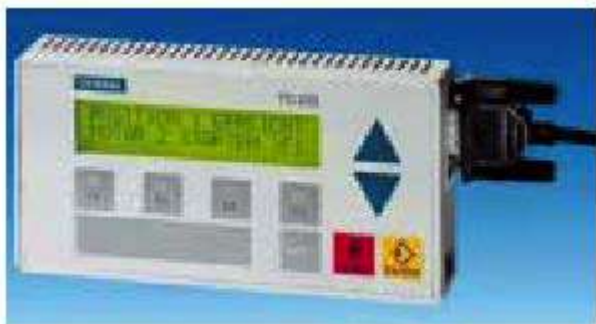


Рис.2.4. Внешний вид текстового дисплея TD 200 оператора SIMATIC TP 070.

Таблица 2.1

Технические характеристики ПЛК SIMATIC S7-200

Параметр	CPU 221	CPU 222	CPU 224	CPU 226
Объем памяти программ (EEPROM)	2К слов	2К слов	4К слов	4К слов
Объем памяти данных	1К слов	1К слов	2,5К слов	2,5К слов
Буферизация данных, ч	50	50	190	190
Интегрированные входы/выходы	6 DI/4 DO	8 DI/6 DO	14 DI/10 DO	24 DI/16 DO
Быстрые счетчики	4	4	6	6
Количество модулей расширения ввода/вывода	—	2	7	7
Максимальное количество входов/выходов (с помощью модулей расширения)	—	24 DI/22 DO 6 AI/4 AO	94 DI/ 74DO 16 AI/10 AO	128 DI/ 112 DO 28 AI/7 AO
Коммуникационные интерфейсы	1×RS-485	1×RS-485	1×RS-485	2×RS-485
Размеры (Ш×В×Г), мм	90×80×62	90×80×62	120,5×80×62	196×80×62
Часы реального времени	Да (модуль)	Да (модуль)	Да	Да

Рис.2.5. Внешний вид сенсорной панели оператора SIMATIC TP 070.



Условия эксплуатации: температура окружающей среды от 0 до 55 °С; относительная влажность от 5 до 95 %; атмосферное давление до 2500 kPa.

Сенсорная панель оператора SIMATIC TP 070 (рис.2.5) предназначена для работы с контроллерами SIMATIC S7-200 (за исключением CPU 212). Она способна поддерживать функции мониторинга и оперативного управления небольшими машинами и системами с использованием до 20 изображений, сенсорной клавиатуры, масштабируемых шрифтов высотой до 2,4 см, других сервисных возможностей, предоставляемых опе-

рационной системой Windows CE. Имеет монохромный (4 оттенка голубого цвета) STN-дисплей с диагональю 5,7" и разрешением 320x240 точек. Подключение к контроллеру осуществляется через MPI-интерфейс. Конфигурирование панели выполняется с помощью пакета STEP 7 Micro/Win V3.1 (Pro). Рабочий диапазон температур от 0 до 50 °С. Степень защиты IP65.

Контроллеры GE Fanuc 90-70 (рис. 2.6) предназначены для решения сложных задач автоматического управления, требующих как высокого быстродействия контроллера, так и боль-

ших ресурсов процессора. Обеспечивают тройное резервирование, улучшенную пакетную обработку и др. [21].

На основе ПЛК GE Fanuc 90-70 возможно построение АСУ ТП для большинства технологических процессов. Эти ПЛК имеют в своем составе модули управления движением, высокоскоростные счетчики, модули программируемого сопроцессора. Если базовой корзины недостаточно для размещения требуемого количества модулей, возможно подключение нескольких корзин расширения.



Рис. 2.6. Контроллеры GE Fanuc Series 90-70.

Модули 90-70 устанавливаются в 5 или 9 слотовую корзину. Корзина крепится на монтажную панель или непосредственно в 19" стойку. GE Fanuc Series 90-70 Для замены модуля требуется открутить 2 винта. Системная шина контроллеров Series 90-70 выполнена в стандарте VME, контроллер поддерживает VME-устройства третьих фирм. Допускается использование блоков питания на 24 V, 48 V (постоянного тока) и 120/240 V (переменного тока).

Контроллеры Series 90-70 содержат в своем составе блоки прерывания и ПИД-регуляторов для различных требований к модулям ввода/вывода. Например, аналоговые модули Series 90-70 обеспечивают высокое разрешение, масштабирование входного сигнала в инженерные единицы, предупреждение о верхней и нижней границе, автоматическое размещение в регистрах и программное конфигурирование уровней сигналов. GE Fanuc предлагает широкий выбор релейных модулей, модулей постоянного и переменного тока в вариантах высокой плотности и стандартной, а также различные варианты модулей по напряжению и току.

Модуль программируемого сопроцессора - однослотовый модуль может быть запрограммирован для подключения операторского интерфейса, обработки в реальном режиме времени, хранения и передачи данных. Программируется с помощью MegaBasic и СИ. Многозадачность позволяет использовать Mega Basic или СИ при применении модуля программируемого сопроцессора в качестве коммуникационного интерфейса. Три светодиода на передней панели модуля указывают о его состоянии. Высокоскоростной счетчик обеспечивает прямую обработку импульсного сигнала до 200 kHz. Тип А - 4 независимо программируемых однонаправленных 16 битных счетчиков. Каждый имеет 3 входа: 1 - предустановки, 2 - счетный вход, 3 - строб. Тип В - 2 идентичных программируемых 32 битных счетчиков. Каждый счетчик может быть независимо сконфигурирован на работу с инкрементом и декрементом. Тип С имеет 1 дифференциальный двунаправленный счетчик с входом, запрещающим счет. Счетчик типа С также обеспечивает начальный цикл для инициализации аккумулятора для определения начальной позиции. Входной маркер используется для указания начального положения. Тип D имеет 4 двунаправленных 32 битных счетчика, которые могут быть сконфигурированы для счета с инкрементом и декрементом. Каждый счетчик включает входной маркер, который может быть использован для установки значения его аккумулятора, определяющего начальное состояние счетчика. Тип Е имеет 2 идентичных 16 битных счетчика со стробированием и входами предварительной загрузки, обеспечивающие счет с инкрементом и декрементом. Каждый имеет запрещение счета и строба. Power Mate I (Модуль управления движением) обеспечивает возможность управления движением, интегрирован с ПЛК для широкого диапазона приложений, начиная с простого по-

зиционирования и измерения скорости до комплексного многоосевого управления позиционированием. Модуль поддерживает до 4 осей. Около 400 настроек движения может быть сохранено в модуле.

Для создания прикладных программ и конфигурирования системы GE Fanuc предлагает программное обеспечение Logicmaster и VersaPro. Имеются средства разработки программных модулей на "C", "State Logic" и "SFC". Система имеет открытую архитектуру и поддерживает следующие сети: Profibus-DP(Master&Slave), Interbus-S(Slave), World FIP(Master&Slave), Lon-Works(Master), DeviceNet(Master), SDS(Master), CAN Open(Master), Modbus RTU(Master&Slave).

Логический микропроцессорный контроллер «Ломиконт» предназначен для решения задач управления в энергетической, химической, металлургической, строительной и других отраслях промышленности [22]. Контроллер принимает дискретные, аналоговые и импульсные входные сигналы; формирует управляющие дискретные, аналоговые и импульсные выходные сигналы; реализует логические операции; производит отсчет времени; выполняет арифметические операции над аналоговыми и целочисленными величинами. Он выполняет дополнительные более сложные операции такие, как: регулирование, фильтрация, интегрирование, кусочно-линейная интерполяция по времени и по параметру и ряду других операций.

Программное обеспечение, хранимое в постоянной памяти контроллера, позволяет работать с ним проектировщикам систем управления, наладчикам и операторам-технологам, от которых не требуется знаний по программированию и вычислительной технике. Специальный простой технологический язык основан на традиционных технологических понятиях ("вход", "выход", "включить", "выключить" и т.п.). Общение пользователя с контроллером осуществляется с пульта контроллера снабженного экраном и клавиатурой. Логика управления конкретным объектом, заданная пользователем с пульта на технологическом языке, называется программой пользователя (ПрП). Максимальный объем ПрП соответствует приблизительно 4000 элементарных логических операций. Предусмотрены средства для быстрой и наглядной отладки, изменения и записи ПрП в энергонезависимую постоянную память. Контроллер имеет выход на печатающее устройство (для документирования процессов), на стандартный алфавитно-цифровой дисплей (для наблюдения за ходом процесса), на ЭВМ верхнего уровня. Информация, выводимая на печатающее устройство и дисплей, может содержать текущие значения технологических параметров и текстовые технологические сообщения произвольного характера, задаваемые пользователем в ПрП, с указанием текущего времени в часах, минутах и секундах.

Существуют четыре модели контроллера Ломиконт. Модели Л-120 и Л-110 - "малый" и "большой" отличаются числом входов/выходов. Модели Л-122 и Л-112 в дублированном исполнении обеспечивают особо высокую надежность работы, достигаемую сочетанием самодиагностики с резервированием.

Контроллер Ломиконт содержит встроенную, непрерывно функционирующую программно-аппаратную систему контроля, которая обнаруживает большинство возможных неисправностей. Конструктивно контроллер выполнен в виде блочного каркаса, в котором располагаются модули с шагом 20 мм. Каркас вмещает 23 модуля. Контроллер является проектно-компонуемым изделием. Содержит базовый и проектно-компонуемый комплекты аппаратуры. Число модулей в каркасе и число каркасов определяются необходимым числом входов-выходов. Кроме блочного каркаса в комплект входят: источник питания, блок вентилятора, батарея сухих элементов, пульт управления, кожух или шкаф. Внешний вид контроллера Ломиконт показан на рис. 2.7.

Основные характеристики Ломиконт-110: до 512 дискретных входов, до 256 дискретных выходов, до 128 аналоговых входов (0-5; 0-20; 4-20 мА и 0-10 В), до 64 аналоговых выходов (0-5; 0-20; 4-20 мА и 0-10 В), до 8 импульсных входов, до 8 импульсных выходов, до 72 таймеров, до 128 счетчиков, коммутирующая способность выходных контактов 48 В при токе 0,2 А, входные и выходные аналоговые сигналы 0-5; 0-20; 4-20 мА и 0-10 В, предел допускаемой основной погрешности 0,5 %, питание от однофазной сети переменного тока 220 В, 50 Гц.

Рис. 2.7. Внешний вид контроллера Ломиконт.

Контроллеры с архитектурой IBM PC занимают все более значительное место в области автоматизации технологических процессов, вытесняя всевозможные промышленные контроллеры нижнего уровня АСУ ТП [23, 24]. Этому способствует несколько факторов, а именно: наличие огромного запаса программного обеспечения, включая и операционные системы реального времени, дешевые аппаратные средства, огромное сосредоточение финансовых и интеллектуальных ресурсов на рынке IBM PC совместимых изделий.

Транснациональный концерн SIEMENS, следуя принципам создания открытых систем, выпустил новое семейство контроллеров SIMATIC WinAC с архитектурой IBM PC [25]. Основное достоинство PC их доступность, т.е. возможность применения при создании АСУ ТП самого современного оборудования очень широкого выбора, поскольку его производят сотни фирм. Для построения систем автоматизации можно применять продукцию различных фирм, отслеживая соответствие стандартам.

К другим достоинствам PC-контроллеров относятся: обширная номенклатура блоков ввода/вывода; наличие стандартизованных интерфейсов (ISA, Compact PCI и др.), промышленных протоколов взаимодействия между контроллерами (Profibus, Modbus, CAN и др.), сетевых средств (Ethernet, Arcnet, Token Ring и др.), сетевых протоколов (TCP/IP, IPX/SPX, Netbios и др.); разнообразные инструментальные средства для разработки программного обеспечения.

Особенностью систем автоматизации нижнего уровня является способность их встраивания непосредственно в промышленное или бортовое оборудование. Подобные системы наряду с хорошими характеристиками вибростойкости, ударопрочности, рабочих диапазонов температур обладают малыми габаритами, низкой потребляемой мощностью и другими специфическими характеристиками.

Открытость - это процесс стандартизации аппаратных и программных архитектур, направленный на достижение аппаратно-программной совместимости и переносимости продуктов большого числа независимых поставщиков. Открытость стандарта, как новая концепция развития рынка средств автоматизации означает отсутствие патентов или авторских прав на спецификацию стандарта и его расширений, отсутствие лицензионной платы за использование стандарта. Открытая система должна быть гибкой и обеспечивать работу 8/16/32/64 - разрядных процессоров и любых устройств ввода-вывода, что позволяет при необходимости совершенствовать аппаратуру без особых затрат и неудобств, связанных с переходом к новой архитектуре при новых технологических изменениях.

Переход подавляющего числа фирм на платформу IBM PC- совместимых контроллеров можно проследить на примере трансформации Ломиконт 110 в IBM PC- совместимый контроллер **Ломиконт ТМ** [28]. Новый контроллер «Ломиконт ТМ» имеет в своем составе новый базовый модуль PRC-TM, который построен на основе одноплатного PC-компьютера. Предназначен для контроля состояния технологических процессов, логического управления, многоконтурного регулирования и других задач, требующих большой информационной и вычислительной мощности. Контроллер имеет процессор 486 с частотой 100 MHz и коммуникационные средства на



базе RS-485 или Ethernet. Предельный объем памяти расширен до 32 МВ. Совместимость «сверху-вниз» позволяет производить замену базового комплекта контроллера Ломиконт-110 на базовый модуль PRC-TM без изменения УСО и схемы подключения к объекту. Реализовано визуальное технологическое программирование на основе стандарта МЭК -1131/3, а также автоматическое горячее резервирование. В поставку контроллера входит базовая версия инструментальной системы TRACE MODE 5 для Windows NT/95/98/2000 на 64000 I/O. OPC-сервер позволяет подключить «Ломиконт TM» к любой современной SCADA по выбору заказчика.

Аппаратное обеспечение Ломиконт TM: процессор 486-100 MHz, ОЗУ - до 32 МВ, флэш-память - до 32 МВ, астрономический таймер-календарь, сторожевой таймер, сеть Ethernet IEEE 802.3 10 Mb/s, витая пара, последовательный порт COM1 (RS-232), последовательный порт COM2 (RS-232/RS-485/пульт), параллельный порт (LPT1), VGA порт, дискретные входы - до 1024, дискретные выходы - до 512, аналоговые входы - до 256, аналоговые выходы - до 128, рабочий диапазон температур от +5 °С до +50 °С, горячее резервирование.

Программное обеспечение включает: встроенный Микро MPB TRACE MODE 5, библиотеку из 150 технологических алгоритмов – фильтрацию, закон регулирования ПИД, нечеткое позиционное регулирование, ШИМ-преобразование, адаптивные и модальные регуляторы, управление устройствами (клапан, задвижка, привод, мотор, насос, группа моторов и т.д.) и транслятор "Микрол-Техно II".

Технические характеристики Ломиконта TM: конструкция - каркас Ломиконта 110 с 23 местами для установки модулей ввода-вывода и процессорного модуля, питание: - 187-242 V с частотой (50±1) Hz, потребляемая мощность не более 150 W, VGA-выход и разъем для подключения клавиатуры, разъем PC/104 для установки дополнительных модулей, блок клавиатуры и индикации, интерфейс связи RS-232, пленочная клавиатура 24 клавиши, четырехстрочный жидкокристаллический дисплей.

Таблица 2.2

Технические характеристики модулей контроллера Ломиконт

<i>Модули ввода дискретных сигналов</i>			
Тип модуля	Тип сигнала	Кол-во каналов	Гальванич. изоляция
ДЦП2	"сухой" контакт 24 V, вх. сопр. 2,4 kΩ	16	попарная
ДЦП32	"сухой" контакт 24 V	32	индивидуальная
ДЦПЗ	=110 V, =220 V, ~127 V, ~220 V	8	индивидуальная
<i>Модуль числоимпульсного ввода</i>			
Тип модуля	Тип сигнала	Кол-во каналов	Гальванич. изоляция
ИЦП2	1 или 2 последовательности импульсов 24 V, не более 50 kHz	2	
<i>Модули вывода импульсных сигналов</i>			
Тип модуля	Тип сигнала	Кол-во каналов	Гальванич. изоляция
ЦИП2	"сухой" контакт 48 V, 0.2 A	8	индивидуальная
ЦИП32	"сухой" контакт 48 V, 0.2 A	16	индивидуальная
<i>Модули дискретных выводов</i>			
Тип модуля	Тип сигнала	Кол-во каналов	Гальванич. изоляция
ЦДП2	"сухой" контакт 48 V, 0.2 A	16	попарная

ЦДП32		"сухой" контакт 48 V, 0.2 A	32	индивидуальная	
ЦДП3		"сухой" контакт 48 V, 0.2 A	16	индивидуальная	
ЦДП4		"сухой" контакт 48 V, 0.2 A	8	индивидуальная	
ЦДП5		"сухой" контакт 48 V, 0.2 A	16	индивидуальная	
Модули ввода аналоговых сигналов					
Тип модуля	Тип сигнала	Кол-во каналов	Гальванич. изоляция	Основная приведенная погрешность	Примечание
АЦП080	напряжение, ток	8	индивидуальная	0,4 %	настройка каналов встроенными средствами
АЦП3	напряжение, ток	8 (дифференциальных)	групповая	0,4 %	для тока - шунты внешние
АЦП2	напряжение	16	нет	0,4 %	
	ток		с РГ12 (2шт.), индив.	0,8 %	
МТП080	термо-э.д.с. термопары	8	индивидуальная	0,5 %	2 группы по 4 канала с независимой настройкой
МТС3	сопротивление 50M, 100M	8	групповая	0,5 %	трехпроводная схема подключения
МТС4	сопротивление 50П, 100П	8	групповая	0,5 %	трехпроводная схема подключения
МТС5	сопротивление 50M, 100M	16	групповая	0,5 %	трехпроводная схема подключения
Модули вывода аналоговых сигналов					
Тип модуля	Тип сигнала	Кол-во каналов	Гальванич. изоляция	Основная приведенная погрешность	Примечание
ЦАП3	напряжение	8	индивидуальная	0,5 %	для тока источник питания внешний
	ток			1,6 %	
ЦАП2	напряжение	8	нет	0,6 %	для тока - шунты внешние
	ток		с РГ22 (2шт.), индив.	1,6 %	

Программируемые микроконтроллеры семейства ADAM. IBM PC совместимый программируемый микроконтроллер ADAM 5510 предназначен для использования в локальных и распределенных системах автоматизации в качестве автономного контроллера [29].



Рис. 2.8. Внешний вид микроконтроллера ADAM 5510.

Внешний вид микроконтроллера ADAM 5510 показан на рис. 2.8. Он обеспечивает прием и выдачу аналоговых и дискретных сигналов, первичное преобразование сигналов по запрограммированным пользователем алгоритмам и обмен информацией по последовательным каналам связи на базе интерфейса RS-485. Контроллер имеет открытую архитектуру и может программироваться как с помощью традиционных языков программирования (С, Ассемблер), так и с помощью языков логического программирования в соответствии со стандартом МЭК 61131 (поддержка ADAM 5510 реализована в системах программирования UltraLogik и Paradym 31).

Таким образом, ADAM 5510 сочетает в себе качества программируемого логического контроллера с простой и открытой архитектурой IBM PC совместимых компьютеров.

Характеристики контроллера: процессор 80188 16-разрядный, память ОЗУ 256 КВ, флэш ПЗУ 256 КВ, операционная система ROM DOS, встроенные часы реального времени, встроенный сторожевой таймер, количество обслуживаемых модулей ввода/вывода 4, два последовательных порта RS-232 и RS-485, напряжение изоляции 3000 В.



Рис. 2.9 Внешний вид микроконтроллера ADAM 5511.

Микроконтроллер ADAM 5511 (рис. 2.9) представляет собой аналог ADAM 5510, в котором реализована программная поддержка популярного протокола ModBus, что позволяет обмениваться данными с любым программным обеспечением верхнего уровня (SCADA) без использования специальных драйверов. Кроме того, ADAM 5511 обеспечивает возможность удаленной загрузки, запуска, останова и завершения программ.

2.2. Микропроцессорные управляющие вычислительные субкомплексы МСКУ

В МСКУ М для организации нижнего уровня АСУ ТП предназначены *микропроцессорные субкомплексы контроля и управления (МСКУ)* - программируемые контроллеры с архитектурой IBM PC, осуществляющие прием и обработку информации от ТОО, управление технологическим оборудованием, регулирование параметров технологического процесса, защиту оборудования и блокировки аварийных процессов [30, 31]. Они могут также использоваться в качестве автономных систем контроля и управления. В МСКУ реализован агрегатно-модульный принцип компоновки изделий (функциональный модуль - плата Е2). Функционирование осуществляется под управлением микропроцессорных контроллеров в соответствии с рабочей программой, хранящейся в оперативной или многократно программируемой постоянной памяти.

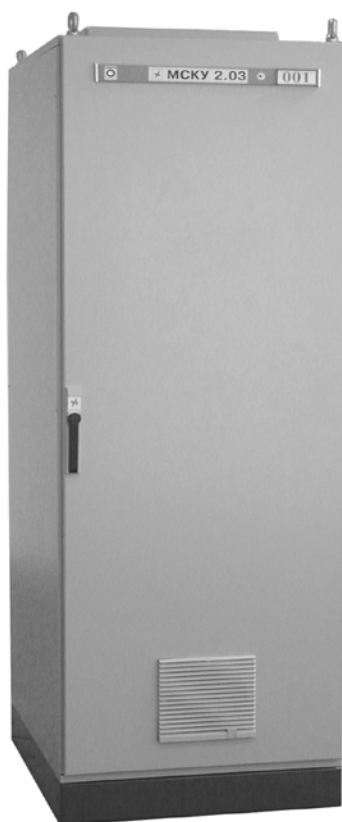
МСКУ реализуют широкий спектр функций, от простых операций ввода/вывода до сложных алгоритмов управления и регулирования. Они являются проектно-компоновочными изделиями (типы и количество каналов связи с объектом и конструктивное исполнение определяется требованиями пользователя). Они компонуются в зависимости от условий эксплуатации в шкафах со степенью защиты IP20, IP43, IP54, IP55. Передний вид шкафа МСКУ показан на рис. 2.10 а, а внутренний – на рис. 2.10 б. Если МСКУ в составе ПТК АСУ ТП используется только для ввода/вывода и первичной обработки информации по встроенным алгоритмам, то он не требует дополнительного программирования. В этом случае настройка МСКУ на задачу конкретного пользователя производится в виде таблиц настройки, формируемых в диалоговом режиме. Средства связи с объектом предназначены для ввода в МСКУ различных типов аналоговых и дискретных сигналов, характеризующих состояние объекта и состояние технологического оборудования, а также для формирования и выдачи управляющих воздействий на различные исполнительные механизмы и устройства.

По конструктивному исполнению средства связи с объектом включают:

- блоки связи с объектом (БСО), выполненные на печатных платах и предназначенные для установки в типовых шкафах МСКУ;
- выносные блоки и приборы, устанавливаемые на объекте в непосредственной близости от технологического оборудования;
- кроссовое оборудование, включающее различные типы соединительных панелей (кроссовых панелей), обеспечивающих прием внешних кабелей, и системных жгутов связи, а также кроссовые шкафы, выполненные как в напольном, так и в навесном исполнениях.

Совокупность блоков связи с объектом обеспечивает ввод/вывод практически всех типов аналоговых и дискретных сигналов, определенных действующими стандартами и включает в себя: блоки ввода (приема) аналоговых сигналов, блоки ввода частотных и импульсных сигналов, блоки ввода дискретных сигналов, блоки формирования дискретных, импульсных и аналоговых сигналов, а также комбинированные блоки.

а)



б)



Рис.2.10. Передний (а) и внутренний (б) вид шкафа МСКУ.

2.2.1 Структура МСКУ

МСКУ представляет собой комплекс технических средств, в состав которого входят: контроллеры микропроцессорные (КМп), блоки связи с объектом различного типа (рис. 2.11). Каждый блок является конструктивно автономным и функционально законченным изделием. Сведения о блоках МСКУ 2 (нового поколения МСКУ) приведены в приложении Д. Любая конфигурация МСКУ содержит ядро, состоящее из одного или трех микропроцессорных контроллеров, соединенных между собой специальными связями.

Подключение различных устройств ввода/вывода как дисплеев, устройств печати, функциональных клавиатур, пультов, щитовых устройств цифровой и алфавитно-цифровой индика-

ции, а также выход в локальные сети обеспечивается через блоки связи по интерфейсам RS-485, RS-232C, ИРПС, МАПС.

МСКУ имеют повышенные характеристики надежности и живучести за счет структурного резервирования (троирование контроллеров и межблочного внутрикрейтового интерфейса), развитой системы контроля и диагностики, автоматической реконфигурации при отказе резервированных частей, обеспечения электропитания от двух независимых фидеров постоянного или переменного тока, сохранения информации в ОЗУ при отключении электропитания и последующего автоматического восстановления работоспособности МСКУ после включения питания, цифровой коррекции результатов преобразования входных аналоговых сигналов. МСКУ выпускаются в нескольких вариантах, отличающихся между исполнением (напольные или навесные шкафы), условиями эксплуатации, числом входных и выходных каналов.

Исходными данными для компоновки конкретного варианта МСКУ является карта заказа. В ней приводятся требования к функциональному составу МСКУ и его техническим параметрам (число микропроцессорных контроллеров, емкость оперативной памяти, типы и количество каналов связи с объектом режим работы, требования к ЗИП и др.). По этим данным система автоматизированного проектирования (САПР-КД) синтезирует структуру МСКУ, определяет номенклатурный состав, рассчитывает ЗИП и показатели надежности.

После обработки заказа САПР выдает полный комплект конструкторской документации, позволяющий изготовить требуемое МСКУ, как законченное изделие.

Блоки связи с объектом, блоки последовательной связи, выходящие на интерфейс ИР, komponуются в двухэтажном монтажном каркасе. Размещение оборудования в МСКУ приведено на рис. 2.12.

Нерезервированный МСКУ строится на базе одного КМп. Резервированный МСКУ представляет собой отказоустойчивую конфигурацию, образованную путем резервирования технических средств. Ядро МСКУ содержит три КМп. Межконтроллерные связи КМп позволяют каждому из них обращаться к другим КМп, обмениваться массивами данных и специальными сообщениями для синхронизации и взаимоконтроля. В резервированных МСКУ для связи с объектом используются трехпортовые блоки связи с объектом, подключаемые к трем КМп как синхронные устройства. Интерфейс ИР предусматривает доступ к одним и тем же блокам с трех направлений.

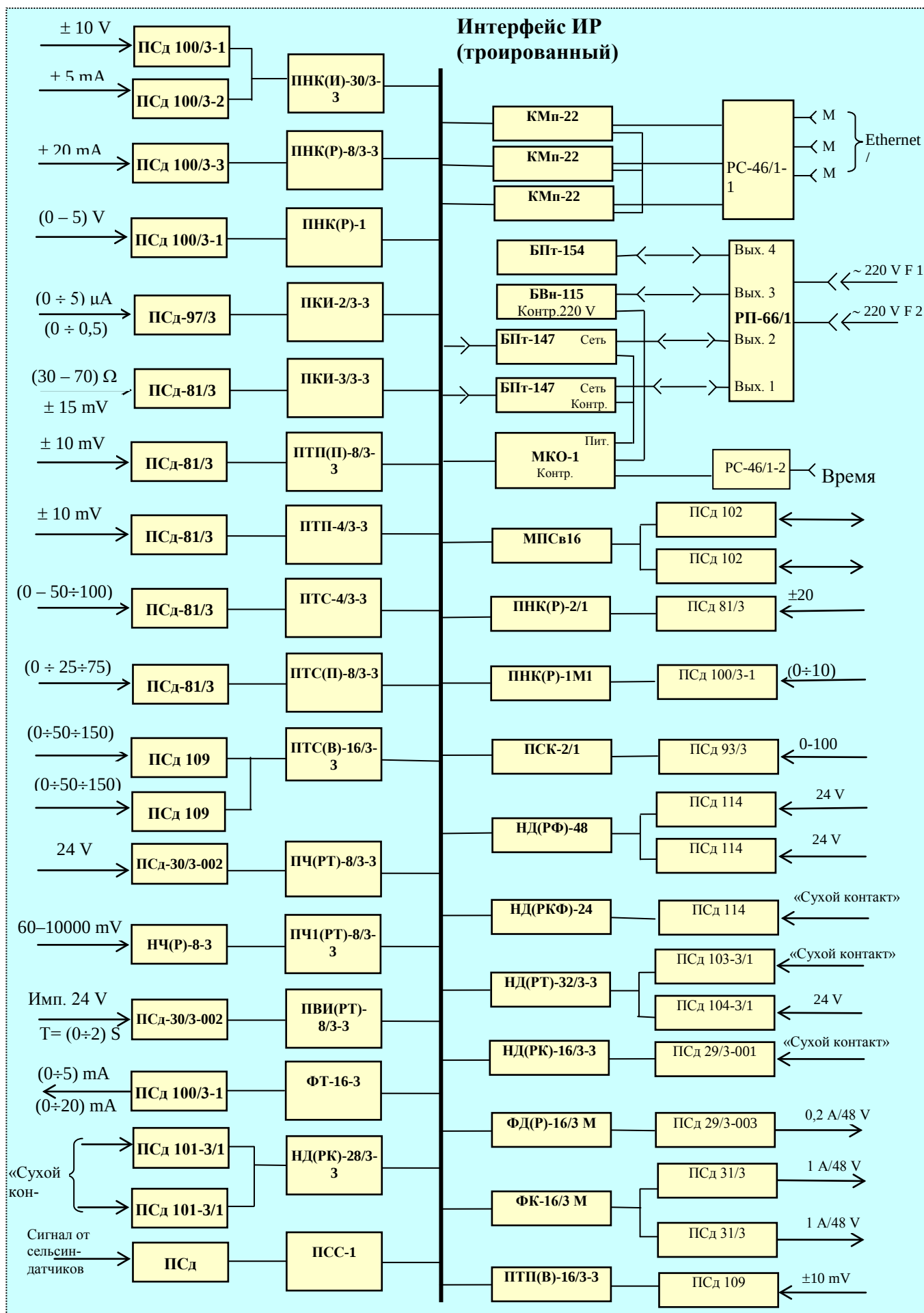


Рис. 2.11. Обобщенная структурная схема МСКУ.

2.2.2 Функциональные возможности МСКУ

Системы управления на основе МСКУ выполняют функции, набор которых достаточен для автоматизации большинства технологических объектов. МСКУ позволяет осуществлять управление, как отдельными технологическими процессами, так и их группами. Кроме того, с помощью МСКУ и локального операторского оборудования возможна организация диспетчерских систем контроля и управления для различных объектов. В зависимости от конкретной конфигурации технических средств и используемых программных средств, МСКУ обеспечивает выполнение необходимых пользователю функций из следующего набора: прием и преобразование в код аналоговых входных сигналов; прием дискретных, импульсных входных сигналов; формирование и выдача дискретных выходных сигналов; формирование и выдача импульсных выходных сигналов для управления исполнительными устройствами в системах регулирования; выполнение всех арифметических действий; обеспечение связи с рабочими станциями управления и другими МСКУ по локальным сетям; передачу по запросу из верхнего уровня информации о текущих значениях параметров управления; получение и выполнение команд от рабочей станции; самодиагностику оборудования с локализацией неисправности до сменного блока. Обеспечено логическое отключение любого из КМп в случае выхода его из строя. После замены отказавшего КМп обеспечивается автоматическое безударное включение его в рабочий режим. МСКУ включает также средства обмена информацией с верхним уровнем управления (сетевое оборудование МАПС и Ethernet – сетевые контроллеры); телекоммуникационные средства (блоки связи, обеспечивающие подключение к МСКУ дисплеев, пультов управления, устройств печати и т.п.). МСКУ функционирует под управлением управляющей системы (УС МСКУ), находящейся на FLASH-диске КМп. УС МСКУ включает комплекс стартовых программ (стартовое ПО) и исполнительную систему (ИС МСКУ).

Стартовое ПО включает программы начального тестирования контроллера и стартовую систему, обеспечивающую запуск МСКУ в требуемом режиме (по включению питания, в режиме восстановления контроллера). ИС МСКУ создается под каждое применение как результат настройки программ из библиотеки программ системы и объединения данных программ в единый загрузочный модуль с задачами пользователя. УС МСКУ загружается в оперативную память из электронного диска при включении питания МСКУ. УС МСКУ выполняет следующие функции: прием запросов от внешних абонентов по сети МАПС; прием запросов от программ пользователя для вывода информации абонентам МСКУ; ввод и обработку информации от блоков связи с объектом (опрос каналов); взаимодействие с соседними контроллерами МСКУ для синхронизации процессов и выравнивания введенных данных; вывод данных в блоки связи с объектом (выдача управляющих воздействий); запуск на выполнение задач пользователя; выполнение операций ввода-вывода для локальных средств связи с оператором и др. В МСКУ предусмотрены встроенные средства диагностики, сигнализации и индикации неисправностей в том числе при отказе составных частей. Более подробно УС МСКУ рассмотрена в главе 3.

2.2.3 Типы и основные характеристики МСКУ

В зависимости от количества входных/выходных каналов, конструктивного исполнения, конфигураций МСКУ поставляются в различных заказных исполнениях. Основные исполнения и технические характеристики МСКУ приведены в табл. 2.3.

Все модификации МСКУ рассчитаны на прием и выдачу аналоговых и дискретных сигналов. Формирование импульсных сигналов, например, для реализации функций импульсного регулятора, выполняется аппаратно либо программно. Эти сигналы поступают на исполнительные механизмы через дискретные выходы функциональных блоков МСКУ. Аппаратура ввода преобразует аналоговые и дискретные сигналы, поступающие на вход МСКУ, в цифровую форму. Аппаратура вывода осуществляет обратное преобразование.

Таблица 2.3

Основные исполнения и технические характеристики МСКУ

Тип МСКУ (тип КМп)	Конструктивное исполнение, конфигурация	Количество БСО, количество каналов ввода/вывода	Габаритные размеры, высота, ширина, глубина, mm	Категория защиты	Условия эксплуатации	Потребляемая мощность, V·A	Электропитание
МСКУ (КМп-14) ПС 1001.901.01 ПС 1001.903.01	шкаф напольный нерезервированный троированный	32/600	1800x800x450	IP20	t от 5 до 50 °C относительная влажность воздуха 95 %	300	Напряжение пост. Тока -24V; 220V Напряжение перем. тока 20V
ПС 1001.901.11 ПС 1001.903.11	нерезервированный троированный	32/600	То же	IP54	То же	300	То же
МСКУ (КМп-15) ПС 1001.901.12 ПС 1001.903.12	шкаф напольный нерезервированный троированный	32/600	То же	IP54	То же	300	То же
МСКУ (КМп-14) ПС 1001.901.21	шкаф напольный троированный	32/600	1800x800x450	IP54 серово дорого устойчивый	То же	300	То же
МСКУ 2. 01 (КМп-20) МСКУ 2. 03	шкаф напольный нерезервированный троированный	32/500	1800x600x800	IP42	t от 5 до 50 °C	400	Пост. ток 220V Перем. ток 220V

Все аналоговые и дискретные входы и выходы МСКУ полностью универсальны в том смысле, что в исходном состоянии они не "привязаны" к каким-либо функциям МСКУ. МСКУ последнего поколения - МСКУ 2.01, МСКУ 2.03 (с внутренней управляющей системой) обладают следующими динамическими характеристиками:

- поддержание единого времени с точностью не хуже ± 2 ms;
- время, затрачиваемое на ввод (включая обмен и выравнивание данных в резервированных МСКУ) для 2 групп каналов ввода дискретных сигналов (32 каналов ввода дискретных сигналов) – не более 8 ms; для 16 групп каналов ввода дискретных сигналов (256 каналов ввода дискретных сигналов) – не более 9 ms; для 100 каналов ввода аналоговых сигналов – не более 55 ms.

2.2.4 Микропроцессорные контроллеры

Основные технические характеристики микропроцессорных контроллеров приведены в табл. 2.4. КМп первых разработок содержали в своем составе аналого-цифровые преобразователи (АЦП), в КМп более поздних разработок функции АЦП реализуются в блоках связи с объектом. Каждый тип контроллера содержит функциональный процессор и процессор

связи. Функциональный процессор осуществляет: управление приемом информации от объекта, выдачу ее на объект, расчеты, связанные с первичной обработкой информации и управлением, контроль и диагностику МСКУ и ряд других функций. Процессор связи выполняет связанные функции МСКУ, формирование сообщений о состоянии МСКУ и объекта управления, ведение службы времени, обслуживание прерываний от блоков связи с объектом, контроль и диагностику КМп.

Для организации резервированных структур МСКУ контроллеры имеют параллельные порты для обмена данными между собой с целью выравнивания данных и синхронизации работы КМп одного МСКУ. Реализация одной из последних моделей КМп – КМп-20 рассмотрена в приложении Д.

2.2.5 Интерфейс ИР

В МСКУ обмен информацией между КМп и функциональными блоками осуществляется по магистральному асинхронному интерфейсу резервированному (ИР), обладающему следующими характеристиками: 1 - интерфейс состоит из одной (нерезервированный вариант) или трех идентичных равноприоритетных магистралей, к каждой из которых подключается только один КМп; 2 - способ адресации функциональных блоков радиальный. Максимальное количество адресуемых блоков составляет 32; 3 - магистраль состоит из 23 сигнальных шин и двух шин синхронизации; 4 - информация передается по 16 двунаправленным шинам (шины адреса и данных); 5 - на каждый сигнал интерфейса отводится два контакта разъема.

Более подробная информация об интерфейсе ИР приведена в приложении А.

Таблица 2.4

Характеристики микропроцессорных контроллеров МСКУ

Характеристики	КМп-14	КМп-15	КМп-20
Тип микропроцессора (сопроцессора)	1834BM86 (1810BM87)	I80386 (I80387)	от 486DX до 586DX133
Тактовая частота, <i>MHz</i>	8	20	от 33 до 133
Емкость ОЗУ, ПЗУ, <i>KB</i>	128-512	1024, +ПЗУ(flash)-256/512	до 64 000 +ПЗУ(flash) до 96 000
Интерфейс связи	RS-232C МАПС	RS-232C МАПС, ARCNET	RS-232C, МАПС, ETHERNET

2.3. Сети

Компоненты ПТК объединяются в единую систему отображения информации и управления на основе коммуникационных сетей и промышленных шин. Основным способом обмена информации является цифровой. В системах передачи данных применяются помехозащищенные протоколы передачи данных. Отключение магистрали не должно влиять на работоспособность подключенных к ней контроллеров. Отказ магистрали идентифицируется контроллером.

Наиболее распространенным сетевым решением является Ethernet, а последние достижения в области увеличения скорости работы Ethernet (до 100 *Mbod*) еще более увеличивает привлекательность этого стандарта. Ethernet широко используется в системах промышленной автоматизации, однако его применение ограничено тем, где циркулирует информация, требующая принятия решений в реальном времени. Это связано с недетерминированностью протокола Ethernet.

В системах, где необходима гарантированная доставка информации в течение заданного интервала времени, более подходящими оказываются протоколы Token Ring (4/16 *Mb/s*) или Arcnet (2,5 *Mb/s*). В настоящее время существует стандарт Arcnet II со скоростью до 20 *Mb/s*. Существуют также сетевые протоколы, специфичные для промышленных систем автоматизации нижнего уровня, такие, как среднескоростные CAN, FIP, BIT-bus, Profibus DP,

Profibus FMS и т.п. Все они часто называются одним термином - Fieldbus. Перспективным является стандарт CAN. Он имеет две основные конкурирующие реализации протоколов верхнего уровня DeviceNet фирмы Allen-Bradley и SDS фирмы Honeywell. В качестве дешевой низкоскоростной магистрали широко применяется интерфейс RS-485, который обеспечивает связь по витой паре сегментами длиной до 1,2 km и позволяет подключать до 32 узлов на сегмент. В качестве логических протоколов работы по RS-485 получили распространение Optomux, Promux и другие совместимые с ними.

Наличие сильных электромагнитных полей на объектах управления приводит к тому, что в качестве физической линии связи часто применяется оптоволокно, а невозможность или дороговизна прокладки кабелей может привести к необходимости использования в качестве линии связи радиоканала.

В состав ПТК должна входить система единого времени, предназначенная для синхронизации таймеров всех вычислительных средств ПТК. Она должна настраиваться по сигналам точного времени и выдавать автоматически синхронизирующие сигналы с заданным периодом для всех таймеров.

Рассмотрим промышленную локальную сеть нижнего уровня на примере сети Profibus фирмы Siemens [20].

2.3.1 PROFIBUS

PROFIBUS (PROcess Field BUS) – это промышленная сеть полевого уровня, отвечающая требованиям части 2 европейских норм EN 50170 и международного стандарта IEC 61158-3 Ed2. Она используется для организации связи между программируемыми контроллерами с одной стороны, и станциями распределенного ввода-вывода, операторскими станциями и другими приборами полевого уровня с другой. Кроме того, PROFIBUS позволяет выполнять дистанционное программирование и конфигурирование систем автоматизации, их отладку и диагностирование. PROFIBUS является открытой системой. Эта сеть позволяет использовать как электрические, так и оптические каналы связи. В последнем случае существенно возрастает стойкость сети к воздействию электромагнитных помех. Существенному снижению затрат на построение оптоволоконных каналов связи способствует наличие широкой гаммы интерфейсных модулей, коммуникационных процессоров и других сетевых компонентов, оснащенных встроенным оптическим интерфейсом. В PROFIBUS реализованы следующие протоколы передачи данных:

- PROFIBUS-FMS (Fieldbus Message Specification – спецификация сообщений полевого уровня);
- PROFIBUS-PA (Process Automation - автоматизация процессов);
- PROFIBUS-DP (Distributed I/O stations - скоростной протокол обмена данными с периферийным оборудованием).

Все протоколы могут быть использованы совместно в рамках одной сети. Протокол PROFIBUS-FMS используется для решения универсальных коммуникационных задач на полевом уровне. Протокол PROFIBUS-DP ориентирован на организацию связи с устройствами распределенного ввода/вывода. Он обеспечивает высокоскоростной циклический обмен небольшими объемами данных. Протокол PROFIBUS-PA используется для организации обмена данными с оборудованием, расположенным в зонах повышенной опасности. С его помощью производится передача данных по сети, отвечающей требованиям международного стандарта IEC 1158-2, а также согласование работы сетей PROFIBUS-DP и PROFIBUS-PA. Технические параметры сети PROFIBUS-DP приведены в табл. 2.5.

Характеристики PROFIBUS-DP: до 125 узлов на сеть, до 32 узлов на сегмент; метод эстафетной передачи данных между ведущими сетевыми узлами с использованием механизма ведущий-ведомый на нижнем уровне; передача данных по витой паре или оптоволоконному кабелю; дистанционное программирование по сети; автоматическое определение скорости передачи данных в сети; открытая система, позволяющая использовать в своем составе

оборудование других фирм-производителей; межузловая передача данных от ведомых устройств; неизменная продолжительность цикла.

Таблица 2.5

Технические параметры сети PROFIBUS-DP

Допустимые длины электрических каналов связи (витая пара)					
Скорость передачи данных в Kb/s	9,6	19,2	93,75	187,5	500
Длина сегмента в метрах	1000	1000	1000	1000	400
Скорость передачи данных в Mb/s	1,5	3	6	12	
Длина сегмента в метрах	200	100	100	100	
Допустимая длина оптических каналов связи					
Тип кабеля	Пластиковый (POF)		PCF	Стекланный	
Скорость передачи данных в Kb/s	до 12000		до 12000	до 12000	
Максимальная длина между двумя сетевыми узлами в метрах	50		300	2850 (световод с 62,5/125; 3,5 dB/km)	

Сетевые узлы. Активными узлами (ведущими устройствами) являются программируемые контроллеры S7-300, -400, M7-300, -400 с интерфейсом PROFIBUS-DP, программируемые контроллеры S5-115u/h, S5-135u, S5-155u/h, S5-95u/dp с интерфейсом ведущего устройства PROFIBUS-DP, программируемые контроллеры производства других фирм с интерфейсом ведущего устройства PROFIBUS-DP, устройства человеко-машинного интерфейса (HMI), программаторов PG 720/740/760 и промышленных компьютеров, системы числового программного управления SINUMERIK, системы автоматического регулирования SIMADYN.

Пассивные узлы (ведомые устройства) включают станции распределенного ввода/вывода ET 200, программируемые контроллеры S7-200 и S7-300, программируемые контроллеры S5-115U/135U/155U, программируемый контроллер S5-95U/DP с встроенным интерфейсом ведомого устройства (до 1,5 Mb/s), модули связи DP/AS-Interface, измерительные преобразователи SITRANS, включенные в сеть PROFIBUS-PA, другие приборы и изделия других фирм (например, преобразователи частоты, задвижки, регуляторы и блоки управления).

Принцип действия PROFIBUS. Все сетевые узлы используют общую среду передачи. Доступ активных устройств к шине разделен во времени. Для этой цели в PROFIBUS используются комбинированные процедуры доступа (метод эстафетной передачи на верхнем уровне и механизм ведущий-ведомый на нижнем уровне). Процедуры доступа разделяются между активными (ведущими устройствами) и пассивными (ведомыми устройствами) сетевыми узлами.

В любой момент времени операциями передачи данных в сети управляет активный узел, получивший эстафетную передачу. После истечения времени, отведенного на работу данного узла, эстафета передается следующему активному узлу и т.д. Если очередное активное устройство не готово к выполнению коммуникационных задач, то эстафета незамедлительно передается следующему активному узлу.

Пассивные узлы, такие как станции распределенного ввода/вывода ET 200, не способны управлять процессом передачи данных. Они способны лишь принимать и передавать данные своему ведущему устройству.

2.3.2 Локальные сети МСКУ М

Локальной вычислительной сетью нижнего уровня является сеть МАПС [32]. В качестве локальной вычислительной сети верхнего уровня в МСКУ М используется сеть Ethernet. Непосредственный выход к сетям верхнего уровня имеют рабочие станции ПС 5110, IBM PC совместимые ПЭВМ, ряд моделей МСКУ. Выход к сетям верхнего уровня других компонентов МСКУ М обеспечен через шлюз на базе ПС 5110 или IBM PC/AT.

Служба единого времени в МСКУ М организуется на базе сети единого времени с использованием сигналов спутниковых навигационных систем РФ (ГЛОНАСС) и США (NAVSTAR).

Сеть МАПС. Модульная асинхронная перестраиваемая сеть (МАПС) является открытой локальной промышленной сетью и предназначена для обмена информацией между компонентами МСКУ М. Сеть МАПС представляет собой совокупность следующих элементов: абоненты сети (МСКУ, рабочие станции, персональные компьютеры и др.), сетеобразующая аппаратура (контроллеры, модемы, мосты и др.) физическая среда передачи сигналов (кабели, тройники, соединители, жгуты), программное обеспечение сети МАПС, протоколы сети МАПС.

Сеть МАПС состоит из одной, двух или трёх магистралей МАПС. Объединение магистралей в сеть, как правило, осуществляется в целях повышения надёжности функционирования сети за счёт резервирования аппаратуры магистралей. Магистралы в составе сети нумеруются цифрами 1, 2 и 3. В дублированных и троированных сетях МАПС абоненты могут иметь выходы либо на все магистрали сети, либо только на некоторые из них. Каждый абонент сети МАПС имеет свой индивидуальный адрес, являющийся числом от 0 до 62 включительно. Структурная схема сети МАПС приведена на рис. 2.13.

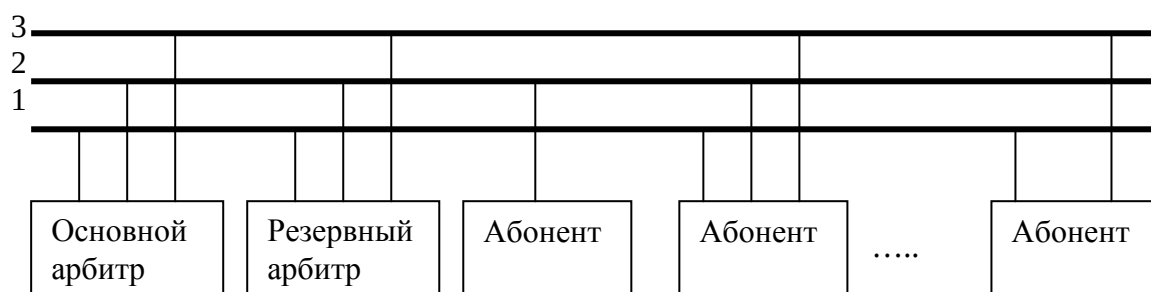


Рис. 2.13. Структурная схема сети МАПС.

Подключение абонентов сети к магистрали МАПС осуществляется при помощи контроллеров связи. Контроллеры связи имеют индивидуальные адреса на магистрали, совпадающие с адресами абонентов сети МАПС. Контроллеры связи выполняют функции абонентов магистрали, обеспечивая обмен данными между абонентами сети через магистраль МАПС. Управление работой магистрали МАПС осуществляет контроллер связи, выполняющий (помимо функций абонента магистрали), функции основного арбитра магистрали. С целью повышения надёжности на магистрали могут работать контроллеры связи, выполняющие, помимо функций абонентов магистрали, функции резервных арбитров магистрали и, в случае выхода из строя основного арбитра магистрали, принимающие его функции на себя. Арбитрам присваиваются начальные адреса, основным является арбитр с минимальным адресом. В системах с жёсткими требованиями к арбитражу многомагистральной сети МАПС один или несколько абонентов сети должны выполнять функции арбитров сети (основного и резервных). Резервирование арбитра в таких системах выполняется в рамках всей сети МАПС. Абоненты сети МАПС, выполняющие функции арбитров сети, должны иметь выходы на все магистрали сети. Магистраль МАПС строится на основе коаксиального радиочастотного кабеля. В табл.2.6. приведены основные особенности сети МАПС, а в табл. 2.7 ее характеристики. Архитектура сети МАПС приведена в приложении Б.

КСв-31 обеспечивает подключение рабочих станций ПС5110 (РС) к магистрали локальной промышленной сети МАПС [33]. Он может выполнять функции основного арбитра, резервного арбитра и абонента магистрали МАПС. Режим работы КСв-31 на магистрали МАПС определяется загружаемой в него из РС программой функционирования. В режиме абонента магистрали МАПС КСв-31 обеспечивает функционирование РС, к которой он подключен, в качестве абонента сети МАПС. В режиме основного арбитра магистрали МАПС КСв-31,

помимо абонентских функций, выполняет управление работой магистрали в соответствии с протоколом сети МАПС. В режиме резервного арбитра магистрали МАПС КСв-31, помимо абонентских функций, анализирует работоспособность основного арбитра, а в случае выявления неработоспособности последнего заменяет его. В интерфейсной области памяти КСв-31, доступной со стороны РС, постоянно хранятся и периодически обновляются текущая конфигурация и текущее время сети в соответствии с принимаемыми сообщениями от арбитра магистрали.

Таблица 2.6

Основные особенности сети МАПС

Структура	магистральная
Базовый сетеобразующий элемент	сегмент магистрали
Способ доступа	маркерный
Управление магистралью	централизованное (арбитр)
Физическая среда	радиочастотный кабель
Способ резервирования	аппаратный (дублирование, троирование)

Таблица 2.7

Характеристики сети МАПС

Максимальное количество абонентов в одной сети	63
Скорость передачи данных в линии связи	1 Mb/s
Максимальное количество абонентов на одном сегменте магистрали	16
Максимальная длина сегмента магистрали на базе радиочастотного кабеля при подключении:	
- 16 абонентов	1200 m
- 8 абонентов	1600 m
- 2 абонентов	1800 m
Количество магистралей в одном сегменте	до трех

При выдаче сообщения другому абоненту сети, РС формирует и записывает сообщение в буферную область памяти данных КСв-31, а последний осуществляет передачу сообщения адресату в свой сеанс владения магистралью. При приёме сообщения от другого абонента сети, КСв-31 записывает принятое сообщение в буферную область памяти данных и информирует об этом РС, а последняя выполняет чтение и обработку сообщения.

КСв-31 реализован на основе цифрового сигнального процессора ADSP-2181 фирмы AnalogDevices и многофункционального связного контроллера SCN2652 фирмы Philips Electronics. КСв-31 имеет оперативную память ёмкостью 80 KB, интегрированную в процессор ADSP-2181 (48 KB – память программ и 32 KB – память данных), которая является доступной для РС. Внешний вид контроллера связи КСв-31 показан на рис. 2.14.

Для взаимодействия КСв-31 с РС используется системный интерфейс промышленного стандарта ISA и подключается к ISA как 16-разрядное устройство ввода-вывода. В адресном поле устройств ввода/вывода РС КСв-31 использует четыре 16-разрядных порта со смежными адресами. Базовый адрес ввода/вывода КСв-31 выбирается из ряда: 220h, 228h, 2A0h, 2A8h, 320h, 328h, 3A0h или 3A8h. КСв-31 использует одну линию запроса прерывания процессора РС, выбираемую из ряда: IRQ3, IRQ4, IRQ5, IRQ7, IRQ10, IRQ11, IRQ12 или IRQ15. Он поддерживает одну скорость обмена по магистрали МАПС, равную 1 Mb/s.

В качестве среды передачи данных используется радиочастотный коаксиальный кабель. Функционально КСв-31 состоит из четырёх основных узлов: узел сопряжения с ISA, узел сигнального процессора ADSP-2181, узел связного контроллера SCN2652, узел сопряжения с МАПС (модем). Центральным узлом КСв-31 является узел сигнального процессора ADSP-2181, который управляет работой всего устройства. Структурная схема КСв-31 приведена на рис. 2.15.

ADSP-2181 – HARC-микропроцессор (Harvard Architecture Computer) фирмы Analog Devices, который обладает производительностью 32 MIPS при тактовой частоте 32 MHz и оптимизирован для высокоскоростной цифровой обработки сигналов. ADSP-2181 имеет базовую архитектуру семейства ADSP-2100 (три вычислительных устройства, два генератора адресов данных и блок управления последовательностью выполнения программы), дополненную двумя последовательными портами, 16-разрядным внутренним портом прямого доступа к памяти, 8-разрядным внешним портом прямого доступа к памяти, программируемым таймером, флаговыми входами-выходами, расширенными возможностями по обработке прерываний и встроенной памятью программ и данных.



Рис.2.14. Внешний вид контроллера связи КСв-31.

В КСв-31 используются следующие аппаратные ресурсы сигнального процессора ADSP-2181:

- память программ ёмкостью 48 KB (16384 24-разрядных слова) - применяется для хранения программы функционирования КСв-31;
- память данных ёмкостью 32 KB (16384 16-разрядных слова) - применяется для организации интерфейсной области памяти, обеспечивающей взаимодействие КСв-31 с РС, а также для буферирования сообщений, выдаваемых в магистраль и принимаемых из магистрали;
- три вычислительных устройства (арифметико-логическое устройство, умножитель аккумулятор и сдвигатель) - применяются для обработки и анализа данных в процессе функционирования КСв-31;
- внутренний порт прямого доступа к памяти - применяется для сопряжения КСв-31 с интерфейсом ISA;
- внешний параллельный порт - применяется для сопряжения ADSP-2181 со связным контроллером SCN2652;
- два последовательных порта, флаговые входы/выходы и входы запросов на прерывание, применяются для управления работой связного контроллера SCN2652 и модема.

Узел сопряжения с ISA обеспечивает взаимодействие КСв-31 с РС и выполняет следующие функции:

- дешифрация адреса КСв-31 и кода выполняемой на интерфейсе операции (запись-чтение данных, запись адреса данных, прерывание процессора КСв-31 или сброс КСв-31);
- согласование шин данных интерфейса ISA и внутреннего порта прямого доступа к памяти процессора ADSP-2181;

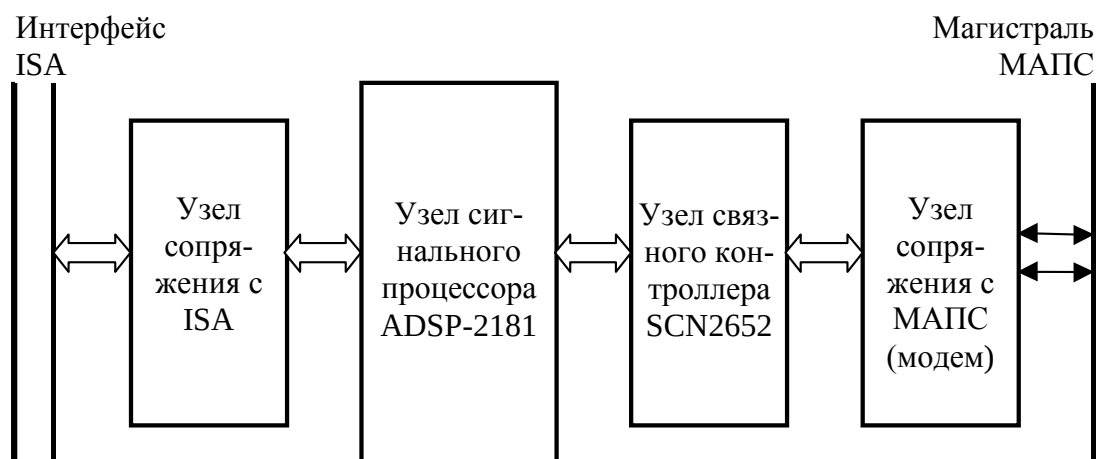


Рис. 2.15. Структурная схема контроллера связи КСв-31.

- формирование запроса прерывания (IRQ) процессора PC; формирование других необходимых сигналов интерфейса ISA;
- выбор базового адреса ввода-вывода КСв-31 и используемой линии запроса прерывания процессора PC путём установки джамперов (перемычек) на соединителях I/O и IRQ, соответственно.

Узел связного контроллера SCN2652 обеспечивает работу КСв-31 на магистрали МАПС в соответствии с принятым в сети МАПС бит-ориентированным протоколом обмена. SCN2652 – многопротокольный программируемый связной контроллер фирмы Philips Electronics, который обеспечивает передачу и приём последовательных синхронных данных со скоростью до 2 Mb/s и поддерживает несколько бит- и байт-ориентированных протоколов обмена. В КСв-31 связной контроллер SCN2652 работает в следующем режиме: бит-ориентированный протокол обмена SDLC (Synchronous Data Link Control) со скоростью приёма-передачи данных 1 Mb/s; автоматическое генерирование и распознавание символов начала и конца сообщения; автоматическое распознавание адреса КСв-31 на магистрали МАПС; автоматическая проверка контрольной суммы сообщения с использованием циклического полинома деления.

Узел сопряжения с МАПС (модем) выполняет модуляцию передаваемых последовательных данных и синхронизирующего сигнала в двухполярный сигнал с фазоразностным кодированием, выдаваемый в магистраль МАПС, и демодуляцию принимаемых последовательных данных и синхронизирующего сигнала из двухполярного сигнала с фазоразностным кодированием, принимаемого из магистрали МАПС. Способ кодирования сигналов в магистрали МАПС показан на рис. 2.16. Модем состоит из кодировщика-декодировщика (кодека) сигналов и блока приёма-передатчиков.

Кодек реализован на базе программируемой логической матрицы EP610 фирмы Altera и выполняет кодировку последовательных данных и синхросигнала в парафазные частотно-модулированные сигналы при передаче сообщения, а также декодировку последовательных данных и синхросигнала из частотно-модулированного сигнала при приёме сообщения.

Блок приёма-передатчиков БППд-17 представляет собой микросборку на основе печатной платы, устанавливаемую в качестве навесного элемента на КСв-31, и выполняет преобразование сформированных кодом парафазных частотно-модулированных сигналов в двухполярный сигнал, выдаваемый в магистраль МАПС, а также преобразование принимаемого из магистрали МАПС двухполярного сигнала в частотно-модулированный сигнал, подаваемый на кодек. В узел сопряжения с магистралью МАПС также входит одновибратор, обеспечивающий отключение передатчика блока БППд-17 в случае превышения допустимого времени выдачи сообщения в магистраль МАПС.

Взаимодействие КСв-31 с PC осуществляется через системный интерфейс ISA. КСв-31 подключается к ISA как 16-разрядное устройство ввода/вывода. В адресном поле устройств ВВО-

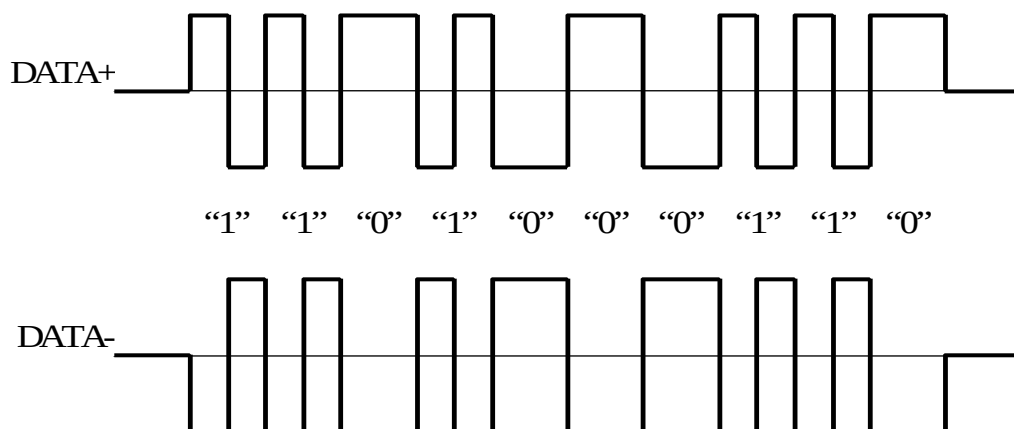


Рисунок 2.16 – Кодировка сигналов в магистрали МАПС

да/вывода PC КСв-31 использует четыре 16-разрядных порта со смежными адресами. Порты ввода/вывода КСв-31 имеют следующее назначение: BaseAddr - порт записи данных в память КСв-31 и чтения данных из его памяти; BaseAddr+2 - порт записи адреса данных в памяти КСв-31; BaseAddr+4 - порт прерывания процессора КСв-31; BaseAddr+6 - порт сброса КСв-31.

Обмен данными между КСв-31 и PC осуществляется через память КСв-31, интегрированную в сигнальный процессор ADSP-2181. Память КСв-31 состоит из памяти программ (16384 24-разрядных слова) и памяти данных (16384 16-разрядных слова). Операциям записи данных в память КСв-31 и чтения данных из его памяти должна предшествовать операция записи адреса данных. Операция записи адреса данных в памяти КСв-31 заключается в выполнении процессором PC команды записи в порт BaseAddr+2 16-разрядного слова, в котором 14-й разряд указывает тип памяти КСв-31 (0 – память программ, 1 – память данных), а разряды 0-13 указывают адрес требуемой ячейки памяти. Записанный таким образом адрес данных в памяти КСв-31 защёлкивается в специальном регистре сигнального процессора ADSP-2181.

В процессе функционирования КСв-31 использует линию запроса прерывания процессора PC для информирования его о факте выполнения различных процедур. Причина прерывания указывается в специальной ячейке интерфейсной области памяти данных. Линия запроса на прерывание остаётся в активном состоянии до получения от PC через специальную ячейку интерфейсной области памяти данных подтверждения об обслуживании прерывания. Для обеспечения оптимальной организации взаимодействия с PC и с магистралью МАПС КСв-31 создаёт в своей памяти очереди передаваемых и принятых сообщений. Таким образом, имеется возможность сохранять сообщения в памяти КСв-31 в течение определённого времени без потери данных. Количество сообщений, находящихся в очередях, ограничивается ёмкостью буферной области памяти КСв-31.

Локальная сеть ETHERNET. В ПТК на базе МСКУ М может использоваться промышленная реализация сети Ethernet. Она соответствует требованиям стандартов Ethernet IEEE 802.3 industry standard 10 Mbps baseband CSMA/CD (10Base-T) and 100 Mbps baseband CSMA/CD (100Base-TX, FX) standards. Пропускная способность - 10/100 Mb/s. Физическая среда – витая пара, оптоволокно. Коммутаторы Ethernet 10/100Base-T обеспечивают построение сети на базе витой пары и оптоволокна. Количество портов - 10 с возможностью установки коммутирующих модулей 10-T(8), 100-TX(8), 10-FL(4), 100-FX(4), Gigabit(1). Сеть ETHERNET

на оптоволокне реализована при помощи модульного многопортового коммутатора HP ProCurve Switch 8000M (рис. 2.17).

Модульный коммутатор Ethernet предназначен для построения высокопроизводительных коммутируемых сетей рабочих групп. Он имеет 40 портов с пропускной способностью 100 Mb/s. Длина линий связи до 2 km. Эффективная пропускная способность 4.67 млн. пакетов в секунду (пакеты по 64 байта). С использованием коммутатора можно построить коммутируемую сетевую инфраструктуру путем подключения к коммутатору концентраторов, других коммутаторов, маршрутизаторов, а также рабочих станций, серверов и сетевых принтеров с обеспечением необходимой для этих устройств полосы пропускания. Основные технические характеристики коммутатора: архитектура Store-and-Forward (промежуточная буферизация пакетов) с временем задержки менее 10 μ s; процессор Intel i960JD с тактовой частотой 66 MHz; оперативная память – 16 MB; операционная система во флэш-памяти емкостью 2 MB с возможностью обновления; 10 слотов для установки коммутирующих модулей, поддерживающих следующие сетевые стандарты:

1. IEEE 802.3 – 10Base_T (10 Mb/s, витая пара);
2. IEEE 802.3j – 10Base_FL (10 Mb/s, оптоволокно);
3. IEEE 802.3u – 100Base_TX и 100Base_FX (100 Mb/s, витая пара и оптоволокно);
4. IEEE 802.3ab – 1000Base_T (1000 Mb/s, витая пара);
5. IEEE 802.3z – Gigabit_SX и Gigabit_LX (1000 Mb/s, оптоволокно).

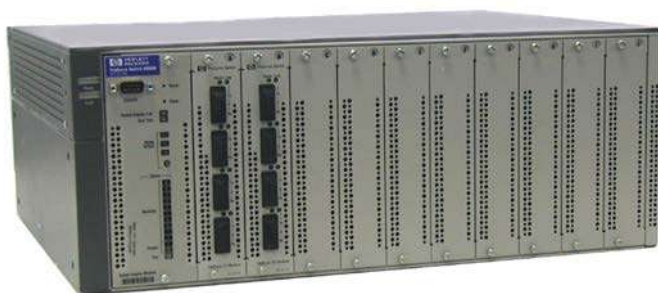


Рис. 2.17. Внешний вид многопортового коммутатора HP ProCurve Switch 8000M.

Общая пропускная способность генмонтажной платы – 3,8 Gb/s. Количество и типы портов определяются количеством и типами установленных коммутирующих модулей. Имеется возможность установки и извлечения коммутирующих модулей без отключения питания коммутатора.

Максимально возможное количество портов: 100Base_TX и 10Base_T – 80; 100Base_FX и 10Base_FL – 40; 1000Base_T, Gigabit_SX и Gigabit_LX – 10. Коммутатор может автоматически создавать таблицы сетевых адресов в режиме самообучения (максимум 8000 адресов). Он имеет широкие возможности по управлению и конфигурированию коммутатора с использованием:

- Web-интерфейса – встроенный графический интерфейс коммутатора, доступный при помощи Web-браузера (например, Internet Explorer или Netscape Navigator);
- консольного интерфейса – полнофункциональный интерфейс терминала VT-100, доступный через последовательный порт коммутатора или непосредственно через сеть Ethernet по протоколу Telnet;
- пакета программ HP TopTools for Hubs & Switches – графический интерфейс на основе протокола SNMP, обеспечивающий управление сетью Ethernet.

Обеспечена поддержка до восьми виртуальных локальных сетей для объединения абонентов сети Ethernet в логические группы.

Технология Store-and-Forward предполагает следующую последовательность прохождения информационных пакетов сети Ethernet через коммутатор: прием всего пакета в буферную память порта коммутатора, к которому подключен источник пакета; анализ структуры, контрольной суммы, заголовка, адресов источника и приемника пакета; трансляция пакета через генмонтажную плату коммутатора в буферную память порта, к которому подключен приемник пакета и выдача пакета в линию связи. Использование технологии Store-and-Forward позволяет обеспечивать взаимодействие между устройствами с различными скоростями передачи

данных (10, 100 и 1000 Mb/s), осуществлять фильтрацию пакетов с некорректной структурой и ошибочной контрольной суммой, осуществлять буферирование пакетов в памяти портов при пиковых нагрузках, осуществлять разделение пакетов по приоритетности. Каждое устройство в сети Ethernet имеет уникальный адрес, состоящий из шести байтов и называемый *MAC адресом*. MAC адрес присваивается устройству при его изготовлении и не может быть изменен в последствии.

При приеме пакетов коммутатор автоматически определяет адреса отправителей и сохраняет в своей таблице сетевых адресов MAC адреса новых устройств, подключенных к его портам непосредственно, через концентраторы или через другие коммутаторы. В таблице сетевых адресов также фиксируются соответствия между MAC адресами устройств и номерами портов коммутатора, к которым они подключены. Емкость таблицы – 8000 MAC адресов. При изменениях конфигурации сети Ethernet (подключение новых устройств, отключение ранее подключенных устройств, переключение устройств с одного порта коммутатора на другой) коммутатор автоматически корректирует таблицу сетевых адресов. Когда коммутатор принимает пакет от какого-либо подключенного к его порту устройства, он по заголовку пакета определяет адрес устройства, которому этот пакет предназначен. В зависимости от того, к какому порту коммутатора подключено устройство, которому адресован пакет, возможны три варианта дальнейших действий:

1. Трансляция пакета (forward) – если устройство, которому адресован пакет, подключено к порту коммутатора не совпадающему с портом, через который пакет был принят, то этот пакет переписывается из буферной памяти входного порта в буферную память выходного порта и выдается в линию связи. Если пакет адресован нескольким устройствам, подключенным к разным портам коммутатора, то он переписывается в буферную память всех этих выходных портов и выдается во все соответствующие этим портам линии связи.

2. Фильтрация пакета (filter out) – если устройство, которому адресован пакет, подключено к тому же порту коммутатора, через который пакет был принят, то этот пакет игнорируется.

3. Размножение пакета (flood) – если адрес устройства, которому адресован пакет, отсутствует в таблице сетевых адресов коммутатора, то этот пакет переписывается в буферную память всех остальных портов коммутатора и выдается в подключенные к ним линии связи; когда устройство, которому адресован пакет, получит его, оно пошлет ответный или подтверждающий пакет и коммутатор сохранит его MAC адрес в своей таблице сетевых адресов.

Коммутатор позволяет сконфигурировать до восьми виртуальных локальных сетей (IEEE 802.3q VLAN – Virtual Local Area Network). Каждая VLAN представляет собой "логический коммутатор" – изолированный широковещательный домен. Операции трансляции, фильтрации и размножения пакетов, описанные выше для физического коммутатора, точно также работают и внутри VLAN. Также как и для отдельных физических коммутаторов, не существует связи между портами коммутатора, относящимися к различным VLAN, если только эти VLAN не соединены через внешний маршрутизатор. Коммутатором поддерживаются следующие дополнительные функции: управление и мониторинг на базе протоколов SNMP (Simple Network Management Protocol) и RMON (Remote Monitoring), Web-технологий: управление потоками (IEEE 802.3x Flow Control) - является одной из важнейших функций коммутатора, которая обеспечивает управление сложным трафиком и помогает предотвратить переполнение буферной памяти коммутатора и потерю пакетов при заторах; управление широковещательными потоками ABC (Automatic Broadcast Control) - позволяет лимитировать число широковещательных пакетов, разрешенных к пересылке через каждый из портов коммутатора; алгоритм основного дерева (IEEE 802.1 Spanning Tree), который позволяет контролировать образующиеся в топологии сети Ethernet петли и может быть использован для обеспечения резервных путей передачи данных; фильтрация пакетов, передаваемых в выходные порты коммутатора (индивидуальные настройки для каждого порта), по следующим признакам: широковещательные пакеты, адреса источников пакетов, сетевые протоколы пакетов (IP, ARP, DEC LAT, AppleTalk, SNA, NetBEUI, IPX). Оптимизация производительности сети Ethernet обеспечивается использованием следующих технологий: объединение портов (Port Trunking) - позволяет устанавливать магистральные

каналы связи, рассматривая несколько параллельных каналов как единый путь передачи данных, объединение портов помогает обеспечить отказоустойчивость, если отказавший канал принадлежит такому составному магистральному каналу, то передаваемый по нему трафик немедленно распределяется между другими каналами этой магистрали; объединение нескольких коммутаторов ячеистой топологией (Switch Meshing), когда существует несколько путей прохождения пакетов между коммутаторами, что позволяет повысить производительность и отказоустойчивость сети Ethernet; технология Switch Meshing - используется только совместно с алгоритмом Spanning Tree; фильтрация пакетов многоадресной рассылки с использованием протокола IGMP (Internet Group Management Protocol) - позволяет автоматически конфигурировать фильтры для многоадресного трафика по протоколу IP (например, при широковещательной рассылке аудио- и видеопотоков), что помогает организовать доставку мультимедийных данных непосредственно на рабочие станции в сложных мультимедийных приложениях; использование классов услуг (CoS – Class of Service) - позволяет организовать разделение пакетов на два потока по их приоритету; защита от несанкционированного подключения к сети Ethernet путем задания для каждого порта коммутатора списка MAC адресов, подключенных к нему устройств

Сеть единого времени выполняет функцию службы времени в ПТК реального времени, где ряд процессов «привязан» к астрономическому времени.

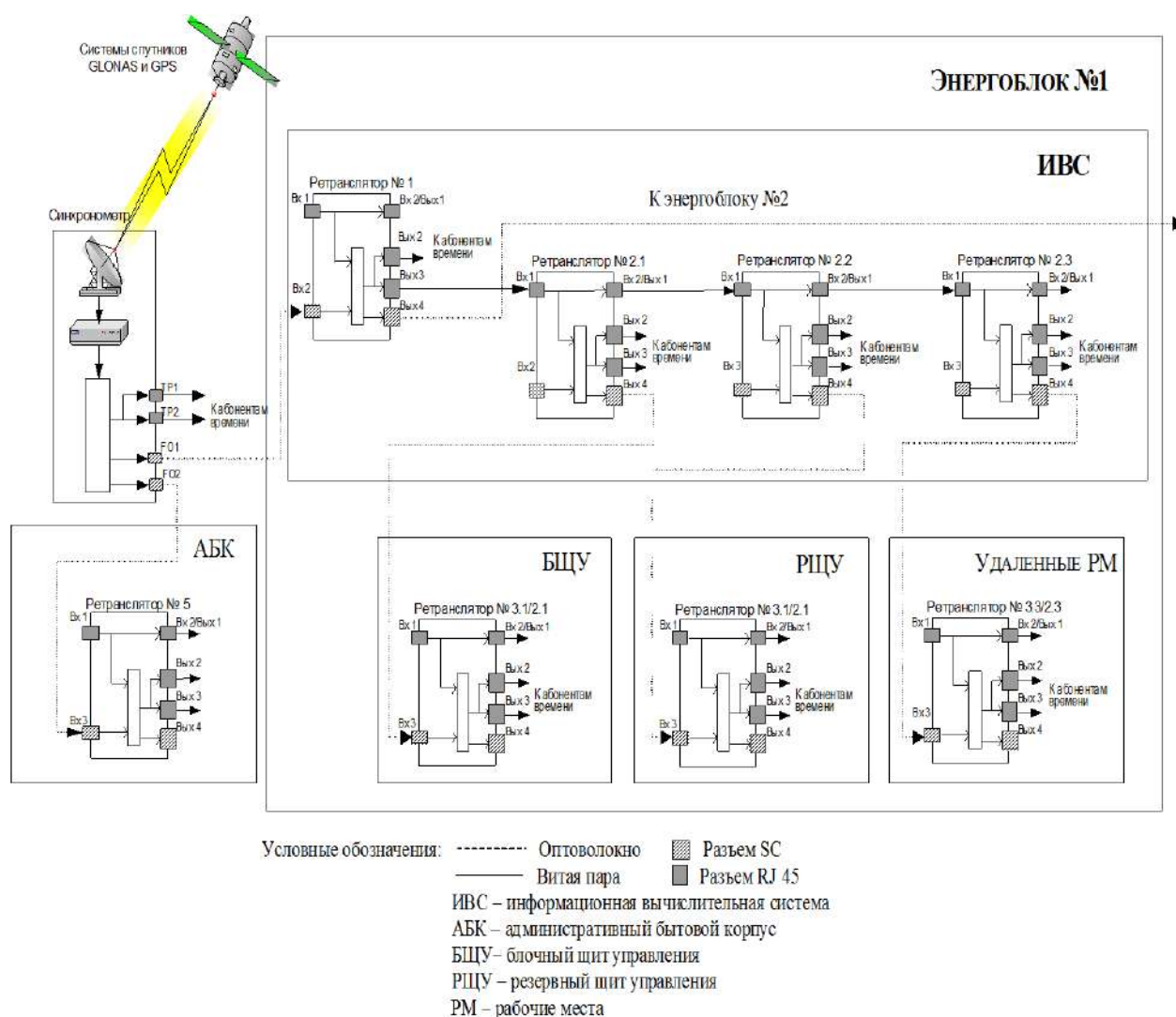


Рис. 2.18. Структура сети единого времени МСКУ М.

На рис. 2.18 представлена структура сети единого времени, реализованной для ИВС энергоблока АЭС. Выдачу сигналов в сеть единого времени производится синхрометром СХр-1 с

погрешностью относительно UTC (шкалы всемирного координированного времени) не более $1 \mu s$. Синхрометр СХр-1 реализован на базе приемовычислителя СН-3833 (рис.2 19).

Прием сигналов временной синхронизации реализован от спутниковых навигационных систем России (ГЛОНАСС) и США (NAVSTAR). Удаление приемной антенны от синхрометра может быть до 65 м. Линии связи реализованы на основе витой пары и оптоволокну. Усилитель-ретранслятор сигналов магистральный (УРСМ) обеспечивает усиление и размножение сигналов для построения разветвленной сети единого времени с единым источником с максимальной задержкой сигнала $5 \mu s$. Он имеет один вход и три выхода для линий связи на витой паре и один вход и один выход для линий связи на оптоволокне. Длина линий связи между ретрансляторами: витая пара – 100 м (псевдомагистраль); оптоволокно – 1 км (радиальная связь).

Рис.2.19. Внешний вид синхрометра СХр-1.

Прием сигналов из сети единого времени в ПС5110 осуществляет блок приема времени БПВр-1. Прием сигналов из сети единого времени в МСКУ 2 осуществляет модуль контроля оборудования МКО-1. Точность привязки событий к единому времени системы не хуже $\pm 2 ms$.



2.4 Технические средства для компоновки верхнего уровня ПТК

Номенклатура изделий и конструктивов для построения верхнего уровня ПТК обеспечивает возможности проектной компоновки рабочих мест операторов-технологов, инженерных станций, файл-серверов, межсетевых шлюзов, информационно-вычислительных комплексов и пр. Основные средства человеко-машинного взаимодействия komponуются на базе цветных мониторов в промышленном исполнении, алфавитно-цифровых и функциональных клавиатур.

Информационно-вычислительный комплекс (ИВК) предназначен для ведения информационной технологической базы данных, а также для выполнения различных расчетов по прикладным программам. ИВК в одних случаях может быть одиночным компьютером с дополнительными устройствами подключения к каналам связи, в других случаях – большой вычислительной системой (мэйнфрейм) или локальной сетью рабочих станций и серверов.

Эргономика конструкций должна учитывать особенности работы операторов-технологов АСУ ТП. Должны быть обеспечены: многоуровневая защита от несанкционированного доступа; соответствие требованиям стандартов по электромагнитной совместимости и безопасности.

В настоящее время на верхнем уровне иерархии АСУ ТП применяются мощные компьютеры промышленного исполнения. Они предназначены для операторского управления производственными и технологическими процессами в масштабах производственного участка, цеха или завода. Такие компьютеры имеют повышенную надежность, предназначены для круглосуточной работы в условиях запыленности, больших перепадов температуры, вибрации и других неблагоприятных факторов. На мировом рынке подобные компьютеры представлены фирмами Hewlett-Packard, Advantech, ICP, Intecolor, Texas Micro и др.

Визуализация и контроль процесса осуществляются при помощи средств, имеющих одинаковую пользовательскую оболочку, как в центральных диспетчерских пунктах, так и непосредственно около управляемой установки. Настройка всей системы осуществляется централизованно на основе общей для всей системы базы данных. Подсистемы визуализации и обслуживания реализуются на базе мощных персональных компьютеров промышленного исполнения с процессорами Pentium III, Pentium IV и полной гаммы периферийных устройств. Основные технические характеристики типовых промышленных ПК:

- тип процессора - Pentium III; Pentium IV;

- тактовая частота - до 1900 MHz;
- кэш память - 1 MB;
- емкость ОЗУ - 1000 MB;
- емкость жесткого диска - до 72 GB;
- монитор - 15"; 17"; 19"; 20"; 21"; 29"; 37"; 42";
- память видеоконтроллера - 4 MB;
- количество подключенных мониторов - 1 - 4.

Для решения задач локального управления в непосредственной близости от технологического процесса в зависимости от условий окружающей среды могут применяться стандартные персональные компьютеры или панели оператора в соответствующем промышленном исполнении. В центральных диспетчерских пунктах, как правило, применяются системы с архитектурой типа "клиент-сервер", которые также реализованы на базе персональных компьютеров различных модификаций.

2.4.1 Рабочие станции ПС 5110

Рабочие станции ПС 5110 (НПО «Импульс») – IBM PC совместимые промышленные компьютеры на базе процессоров INTEL PENTIUM ® III с частотой 1,26 GHz и выше с широким набором периферийных устройств [34, 35]. ПС 5110 обладают высокой коррозионной стойкостью, стойкостью к воздействию повышенной влажности, вибрации, пыли, перенапряжений и провалов в питающей сети и электромагнитных помех. Используются технологии человеко-машинного взаимодействия на базе: цветных XVGA - совместимых мониторов в промышленном исполнении, алфавитно-цифровых и функциональных клавиатур, манипуляторов типа Durapoint, мышь.

Состав рабочей станции: шасси, генмонтажная плата, системная плата, дисковые накопители, контроллеры ввода-вывода, устройства ввода, связанные контроллеры, устройства отображения, сетевое оборудование, источник питания, устройство бесперебойного питания, стол, тумбы, шкаф. Аппаратная архитектура соответствует Intel x86. Условия эксплуатации: температура от +15 °C до +50 °C; относительная влажность до 75 % при 30 °C; барометрическое давление от 86 до 108 kPa, сейсмостойкость до 6 баллов при высоте 30 m над отметкой "0". ПС 5110 изготавливаются в заказных вариантах. Исполнения отличаются составом оборудования и его компоновкой в конструктивах. Состав оборудования и вариант компоновки задаются пользователем при заказе ПС 5110. На рис. 2.20 показан конфигуратор ПС 5110 – комплект технических средств для компоновки ПС 5110.

В состав ПС 5110 могут входить от одного до трех системных блоков. Системный блок – основное устройство рабочей станции ПС 5110. Системный блок содержит в своем составе блок процессорный, который управляет функционированием рабочей станции ПС 5110. Системный блок обеспечивает возможность подключения к нему других устройств, входящих в состав рабочей станции ПС 5110, а также линий связи с другими вычислительными системами. Системные блоки komponуются на основе шасси промышленного компьютера, в которое устанавливается источник питания, генмонтажная плата, блок процессорный, контроллеры и дисковые накопители. Генмонтажная плата системного блока может иметь до 14 слотов системных интерфейсов PCI и ISA в различных комбинациях. Максимальное количество устанавливаемых в системном блоке дисковых накопителей равно пяти (три – 5,25" и два 3,5").

Блок процессорный имеет в своем составе основные системные устройства: процессор, набор системных микросхем (chip set), слоты для установки модулей оперативной памяти, интерфейсы дисковых накопителей (EIDE, FDD), интерфейсы устройств ввода-вывода, watch-dog timer, DiskOnChip и другие. В случае необходимости, блок процессорный может иметь интерфейс Ultra 160 SCSI (только для подключения HDD) и видеоконтроллер с шиной AGP (видеопамять до 4 MB).

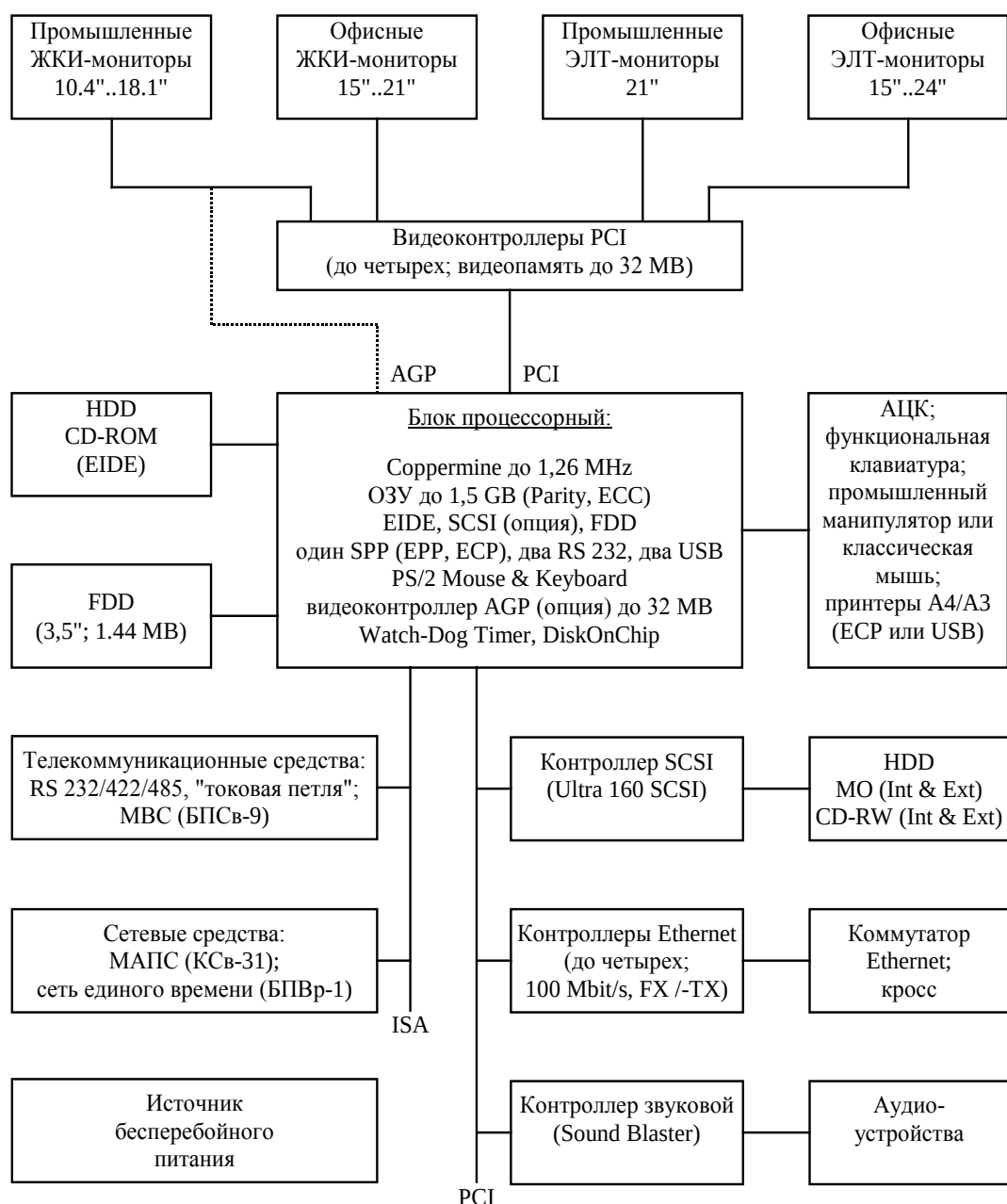


Рис. 2.20. Конфигуратор PC 5110.

В состав PC 5110 могут входить: до четырех видеоконтроллеров для шины PCI; контроллер Ultra 160 SCSI, обеспечивающий подключение до 15 внутренних и внешних дисковых накопителей (HDD, MO, CD-RW); контроллер, совместимый с Sound Blaster; до четырех контроллеров Ethernet для оптоволоконных (100Base-FX) и медных (100Base-TX) линий связи со скоростью передачи данных 100 Mbit/s; коммуникационный контроллер, включающий два дополнительных последовательных порта (RS 232/422/485 или "токовая петля"); контроллеры связи для подключения к сети МАПС (КСв-31); блок приема времени из сети единого времени (БПВр-1); мониторы плоскопанельные на основе жидкокристаллической матрицы (ЖКИ-мониторы), мониторы на основе электронно-лучевой трубки (ЭЛТ-мониторы); лазерные (черно-белые) и струйные (цветные) принтеры форматов A3 и A4, подключаемые к системному блоку как через параллельный (ECP), так и через последовательный (USB) интерфейсы; источник бесперебойного питания типа "on-line" (с двойным преобразованием

входного напряжения и нулевым временем переключения на резервный источник питания). ПС 5110 может компоноваться как в одном, так и в нескольких базовых конструктивах.

Шасси представляет собой стальной высокопрочный корпус с алюминиевой лицевой панелью и двумя вентиляторами со сменными фильтрами (рис. 2.21). Шасси имеет 5 отсеков для дисководов в корзине на противоударной подвеске. В шасси устанавливается PICMG-генмонтажная плата на 14 слотов с интерфейсом PCI/ISA и PS/2-источник питания.

Генмонтажные платы - четырехслойные с внутренними слоями – цепи питания и земли (в виде экранов), интерфейсы: PCI rev. 2.1; ISA IEEE P996 (рис.2.22). Есть светодиодная индикация наличия номиналов питания. Обеспечена поддержка источников питания AT и ATX.

Системные платы содержат процессоры Intel: Pentium III (Tualatin) - 1,26 GHz, Chipset Intel 82440BX AGPset с поддержкой 100 MHz FSB, ОЗУ до 1,5 GB SDRAM с ECC, интерфейсы: PCI, ISA, PC/104, EIDE, FDD, внешние порты: 2S+1P+2USB и дополнительные устройства - RTC, DiskOnChip, Watch-Dog (рис. 2.23).



Рис. 2.21. Шасси МАПС.

В качестве *дисковых накопителей* используются:

- HDD: Ultra3 SCSI, 160 Mb/s; IBM UltraStar, Seagate Barracuda; емкость 9 GB/18 GB/36 GB; внутренний буфер данных 4 MB; среднее время доступа 4,2 ms; установившийся внешний поток данных 21.7-36.1 Mb/s; наработка на отказ не менее 1 млн. часов;
- CD-RW: CD-W 540E (Teac), Ultra2 SCSI, 40x/12x/48x, ext & int;
- MO: MDE3130SS (Fujitsu), Ultra2 SCSI, 640 MB/1.3 GB; ext & int;
- CD-ROM CD-540E540E (Teac).

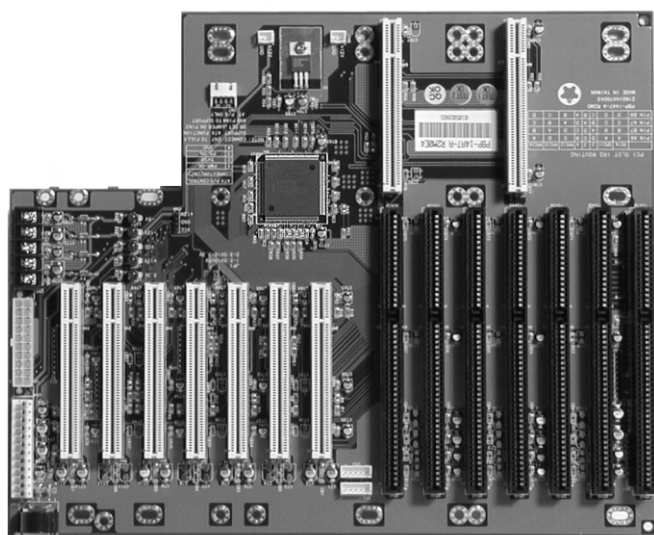


Рис.2.22. Генмонтажная плата.

Контроллерами ввода-вывода являются: SCSI: ASC 29160 (Adaptec), Ultra 3, PCI-bus, 160 MB/s, LVDS & SE, int & ext, до 15 устройств; Видеоконтроллер Matrox G450 DVI (Matrox): PCI-bus, 16-32 MB, SDRAM, до 1600x1200, 85 Hz, поддержка многомониторного режима (до трех мониторов); Sound: Creative, Live!, PCI-bus, звуковой процессор EMU 10K1, 16-bit 48 KHz, Full Duplex.

Устройства ввода - функциональные клавиатуры с двумя

типами исполнений: настольные и встраиваемые (рис. 2.24). Они имеют по 128 кнопок и светодиодов (конфигурируемых при заказе)

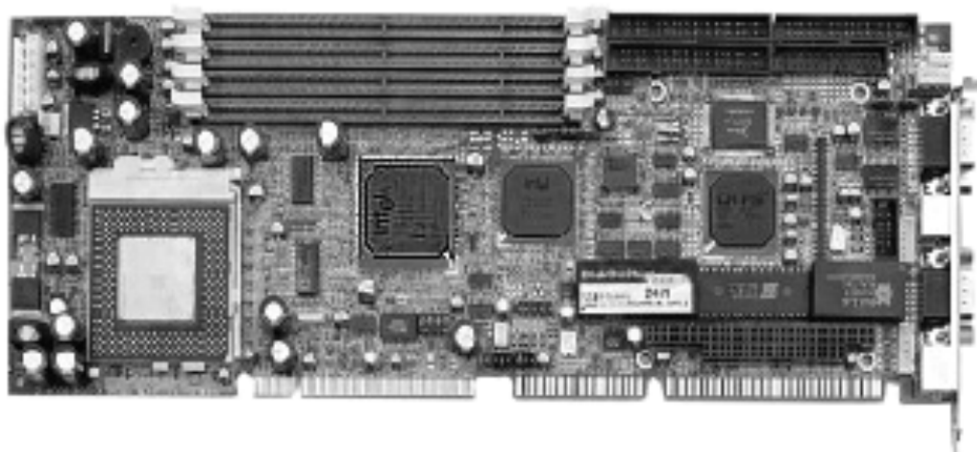


Рис.2.23. Системная плата.

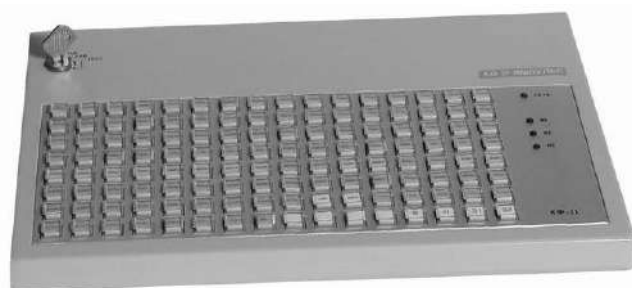


Рис. 2.24. Клавиатура устройства ввода.

Принтеры: HP LaserJet – лазерные; HP DeskJet – струйные; форматы A4, A3; интерфейсы ECP/EPP, USB, Ethernet.

Источником питания ПС 5110 является дублированный AT/ATX источник питания PS/2. Он содержит два модуля 250 W с “горячей” заменой, имеется звуковая и

светодиодная сигнализация неисправности модулей (рис.2.27). Параметры входа - 180-264 V частотой, 47-63 Hz.



Рис. 2.25. Внешний вид устройства бесперебойного питания.

Интерфейсы RS-232 и ИРПС. Клавиатура КФ-4 – мембранная с пленочной лицевой панелью, степень защиты IP42. Клавиатура КФ-11 – полноходовая с возможностью замены этикеток кнопок, степень защиты IP20.

Устройство бесперебойного питания в промышленном исполнении с запираемой дверью на лицевой панели для ограничения доступа (рис.2.25). Управление осуществляется через интерфейс RS-232. Имеется защита выходов от короткого замыкания.

Манипуляторами служит «мышь» в промышленном исполнении DuraPoint & Trackball (рис.2.26).

Устройствами отображения в ПС5110 служат: ЖКИ мониторы, имеющие следующие характеристики - 21.3", 18", 17", 15" NEC MultiSync; 15", Astech, яркость до 250 cd/m², контрастность 250..400:1; интерфейс аналоговый; ЭЛТ мониторы: 21" Varco ICD321I, UXGA; промышленное исполнение.

Конструктивы:

Стол: рабочее место оператора; расположение тумбы справа или слева (рис. 2.28).

Тумба-подставка: размещение мониторов и системных блоков операторских рабочих станций (рис. 2.29).

Тумба: размещение серверных рабочих станций, шлюзов и сетевого оборудования.

Шкаф: размещение мониторов, системных блоков и сетевого оборудования (рис.2.30).



Рис. 2.26. Внешний вид манипулятора.



Рис. 2.27. Внешний вид источника питания.



Рис. 2.28. Рабочее место оператора.

Рис. 2.29. Внешний вид тумбы с монитором.



Поставка рабочих станций производится в виде заказных проектно-компонуемых и конструктивно законченных исполнений, ориентированных на использование в составе ПТК или различных систем автоматизации для АЭС и других применений.

Рис. 2.30. Шкаф: размещение мониторов, системных блоков и сетевого оборудования.



2.5 Посты управления и обслуживания

Предназначены для оперативного управления технологическим процессом и обслуживания ПТК. Предусматривается центральный щит управления, щит оперативного обслуживания и наладки для ПТК и местные щиты управления.

Центральный щит управления является основным постом оперативного управления. С центрального щита управления производится управление всем оборудованием со всеми вспомогательными подсистемами и механизмами в режимах пуска, нормальной эксплуатации, планового останова и аварийных ситуациях. При этом оператор имеет доступ к любой информации, необходимой для контроля и управления, и возможность воздействия на любые исполнительные механизмы. В оперативном контуре управления ЦЩУ размещаются средства управления и представления информации, а также резервные средства контроля и управления.

Резервные средства контроля и управления предусматриваются для прямого доступа к особо ответственным исполнительным устройствам и сигналам. Объем резервных средств задается отдельно для каждого объекта Генеральным разработчиком АСУ ТП.

Щит оперативного обслуживания и наладки включает в себя: технические средства контроля функционирования ПТК и АСУ ТП в целом, средства корректировки программного обеспечения и документации, средства наладки ПТК и АСУ ТП в целом. Щит включается в неоперативный контур ЦЩУ. Он обеспечивает: тестирование аппаратуры ПТК, документирование и отображение на экранах мониторов сообщений о повреждениях в системе управления, документирование и отображение сигналов, характеризующих состояние АСУ ТП и технологического объекта, изменение в режиме диалога структуры систем автоматического управления и регулирования, корректировку и расширение прикладного программного обеспечения.

Местные щиты предназначены для проведения испытаний и пусков после капитальных ремонтов и длительных простоев. В нормальных режимах работы на местных щитах персонал отсутствует. Там может размещаться аппаратура ПТК.

2.6 Организация электропитания и заземления ПТК

2.6.1 Организация электропитания

Электропитание комплекса технических средств ПТК осуществляется от однофазной сети переменного тока напряжением 220 (+22, -33%) V с частотой (50±1) Hz и коэффициентом несинусоидальности не более 10 %. В качестве первичных источников питания ПТК могут использоваться цеховые распределительные подстанции, распределительные щиты, питающие сборки системы электроснабжения автоматизированного объекта, к которым не подключена посторонняя нагрузка (мощные электродвигатели, электропечи и т.п.).

Для организации системы электропитания помещение, предназначенное для установки комплекса технических средств, должно быть оборудовано распределительным щитом с установленными на нем выключателями, розетками 220 V и автономной магистралью защитного заземления. Схема организации электропитания и заземления МСКУ приведена на рис. 2.31. Электропитание МСКУ рекомендуется осуществлять от двух распределительных щитов РЩ. Электропитание МСКУ может быть организовано:

- от одной или двух независимых взаиморезервирующих однофазных сетей переменного тока напряжением от 187 до 242 V и частотой (50±2) Hz с глухо заземленной нейтралью;
- от одного (нерезервированного) или от двух независимых фидеров напряжением 220 V постоянного и переменного токов.

Сетевое напряжение поступает на блоки питания БПт-147 через РП с последующим преобразованием во вторичные питающие напряжения: +5 V, +24 V и минус 24 V. БПт-147 осуществляет указанное выше преобразование по каждому фидеру электропитания независимо.

Выходы одноименных вторичных напряжений по каждому фидеру объединены через разделительные диоды.

Для обеспечения защиты внешних цепей питания от короткого замыкания в цепях питания МСКУ предусмотрены плавкие вставки, которые устанавливаются в РП. В случае эксплуатации МСКУ в заведомо тяжелых условиях (относительно электромагнитных воздействий) рекомендуется осуществлять первичное электропитание МСКУ от отдельного трансформатора. Вторичная и экранизирующая обмотки трансформатора соединяются с контуром защитного заземления. Организация электропитания и заземления ПТК, устанавливаемого в одном помещении, представлена на рис.2.32. Организация электропитания и заземления ПТК, устанавливаемого в нескольких помещениях, представлена на рис.2.33.

Питание датчиков и исполнительных механизмов осуществляется с учетом требований соответствующих технических описаний и инструкций по эксплуатации.

2.6.2. Защитное заземление

Защитное заземление устанавливается с помощью заземлителей. Последние надежно соединяются с контуром заземления. В качестве заземлителей используются стальные трубы с толщиной стенки не менее 3,5 *mm* или стержни с поперечным сечением не менее 48 *mm*². количество заземлителей должно быть не менее 5. Для контура заземления используются металлические шины прямоугольного сечения не менее 24 *mm*² и толщиной 2 *mm*.

Магистраль заземления должна обеспечивать сопротивление между корпусом любой составной части и землей не более 3 Ω в любое время года. В качестве заземляющих проводников рекомендуется применять оголенные медные провода или многожильные изолированные провода со следующими сечениями в зависимости от длины: 4 *m* - 3 *mm*², 15 *m* - 8 *mm*², 9 *m* - 5 *mm*², 25 *m* - 10 *mm*². Заземляющие проводники должны быть защищены от механических воздействий.

В комплексах на базе МСКУ предусмотрено два вида заземления: защитное и рабочее (логическое). Рабочее и логическое заземление - это общая цепь. Провода рабочего заземления по возможности должны быть удалены от элементов конструкции корпуса, проводов защитного заземления и проводов первичного электропитания. При разводке внешних проводов рабочего и защитного заземления предпочтение следует отдавать радиальному подключению этих проводов.

2.6.3 Рекомендации по повышению помехозащищенности измерительных цепей

Уровень электромагнитных помех, возникающих в измерительной цепи (линии связи между датчиком, измерительным оборудованием ВК), зависит от энергоемкости объекта, типов технологического оборудования, защитных средств в местах возникновения помех, типа кабеля, трассы его прокладки и длины и взаимного влияния измерительных цепей. Помехи в измерительных линиях могут возникать под действием электромагнитных полей, обусловленных работой промышленных электрических установок, а также из-за наличия емкостных связей, между различными цепями, расположенными в одном кабеле. В схемах управления на переменном токе, в которых имеются длинные кабельные линии, содержащие цепи с одним общим проводом, могут образовываться ложные цепи (из-за наличия емкостных связей), в результате чего могут происходить ложные срабатывания реле, элементов сигнализации и других исполнительных устройств. Учесть эти факторы и указать тип кабеля, который был бы экономически оправдан, невозможно без проведения соответствующих измерений на конкретном объекте. На уровень сигнала в измерительном кабеле действуют два вида помех: помехи нормального $E_{\text{ПНВ}}$ и общего $E_{\text{ПОВ}}$ вида. Помеха нормального вида возникает в результате воздействия на измерительную линию электромагнитного поля Φ , которое приводит к возникновению ЭДС помехи (рис. 2.34). Помеха общего вида возникает в

результате протекания по измерительному проводу тока, вызванного разностью потенциалов между точками заземления датчика U_1 и измерительного устройства U_2 (рис. 2.35). В большинстве случаев на промышленных объектах это напряжение частоты 50 Hz. К помехам общего вида относятся также микросекундные и наносекундные помехи, образующиеся в процессе коммутации индуктивных нагрузок. В конечном итоге, помехи общего вида $E_{\text{пов}}$ приводят к появлению помех нормального вида $E_{\text{пнв}}$, действующих последовательно с ЭДС датчика E_{ϕ} .

При внедрении АСУ ТП на действующих объектах характер и уровни помех можно определить экспериментальным путем. С этой целью от датчика к измерительному устройству прокладывается выбранный тип кабеля по трассе, максимально приближенной к последующему реальному монтажу линий связи между датчиком и блоками связи с объектом МСКУ. Измерение помех производится следующим образом. На действующем объекте к двум резервным жилам кабеля, который предполагается использовать для ввода сигналов в БСО, подключается резистор величиной 2 kΩ (на стороне подключения датчика). На противоположном конце линии к этим жилам подключается осциллограф, с помощью которого производится измерение амплитуды помех нормального вида. Экран этой пары жил подключается к общему проводу осциллографа, а сам осциллограф заземляется.

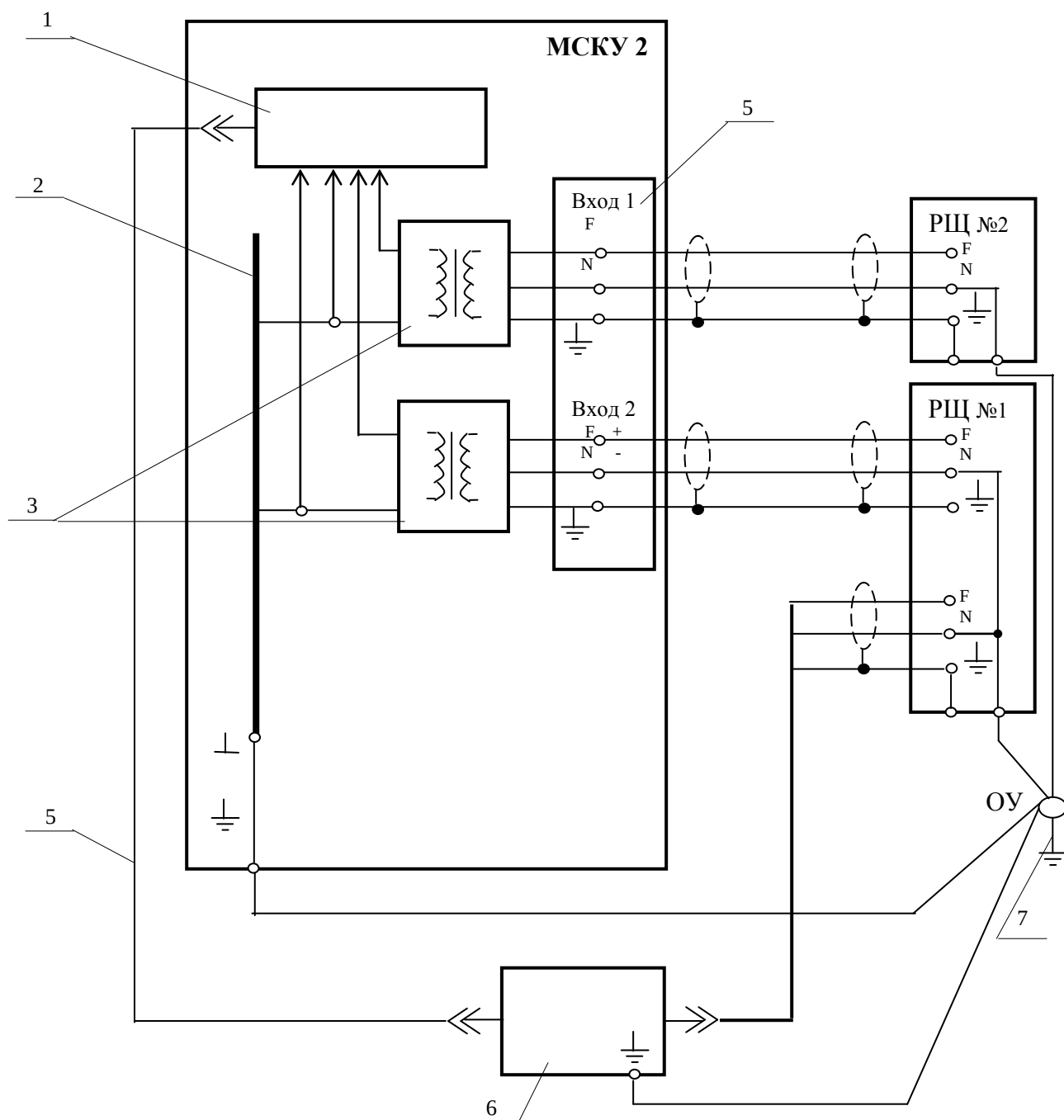


Рис. 2.31 Схема организации электропитания и заземления МСКУ

На рис. 2.31 обозначено: Др – высокочастотный дроссель; РЩ – распределительный щиток питания; ОУ – опорный узел (нулевая шина); 1 – логические блоки; 2 – шина; 3 – блоки питания БПт-147; 4 – информационный кабель (сеть МАПС); 5 – распределитель питания РП-66 (РП-66/1); 6 – средства оперативного управления (ПЭВМ и т.п.); 7 – контур защитного заземления.

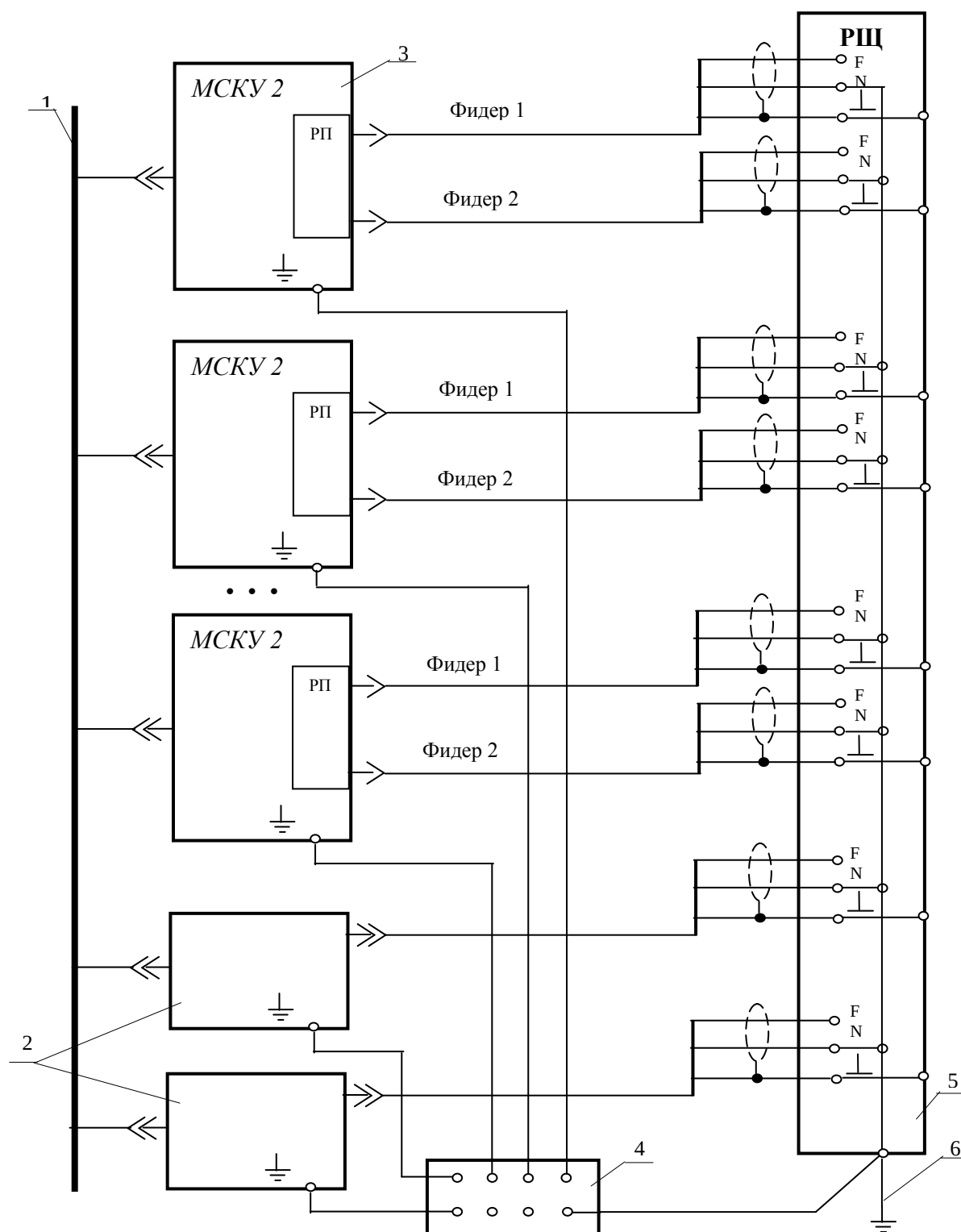


Рис. 2.32. Схема организации электропитания и заземления ПТК, установленного в одном помещении.

На рис. 2.32 обозначено: 1 – информационная магистраль (сеть МАПС); 2 – средства оперативного управления (ПЭВМ, рабочие станции и т.п.); 3 – МСКУ; 4 – опорный узел комплекса (нулевая шина); 5 – распределительный щиток электропитания; 6 – контур защитного заземления.

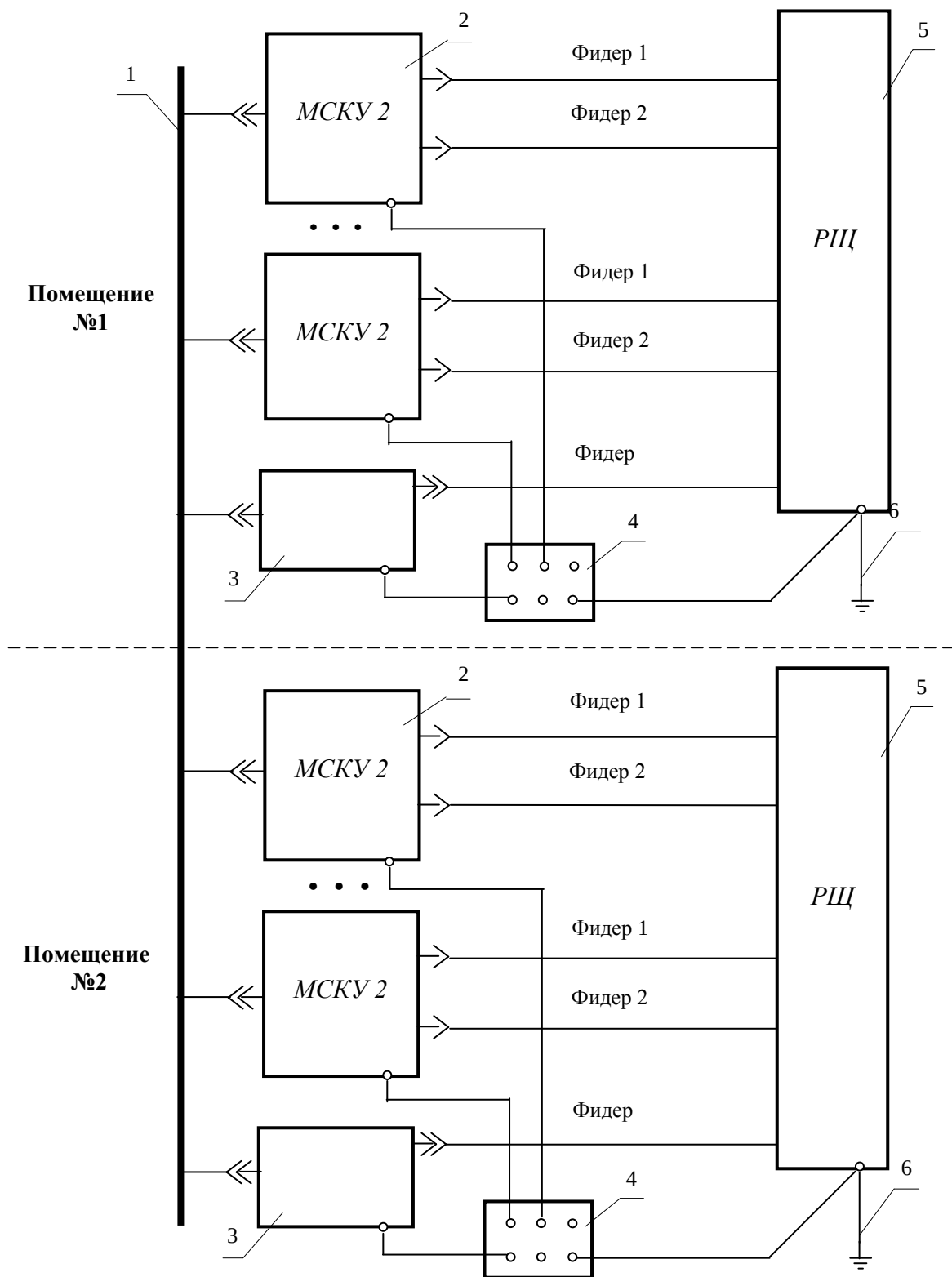


Рис. 2.33. Схема организации электропитания и заземления ПТК, установленного в нескольких помещениях.

На рис. 2.33 обозначено: 1 – информационная магистраль (сеть МАПС); 2 – МСКУ; 3 – средства оперативного управления; 4 – опорный узел комплекса (нулевая шина); 5 – распределительный щиток питания; 6 – контур защитного заземления.

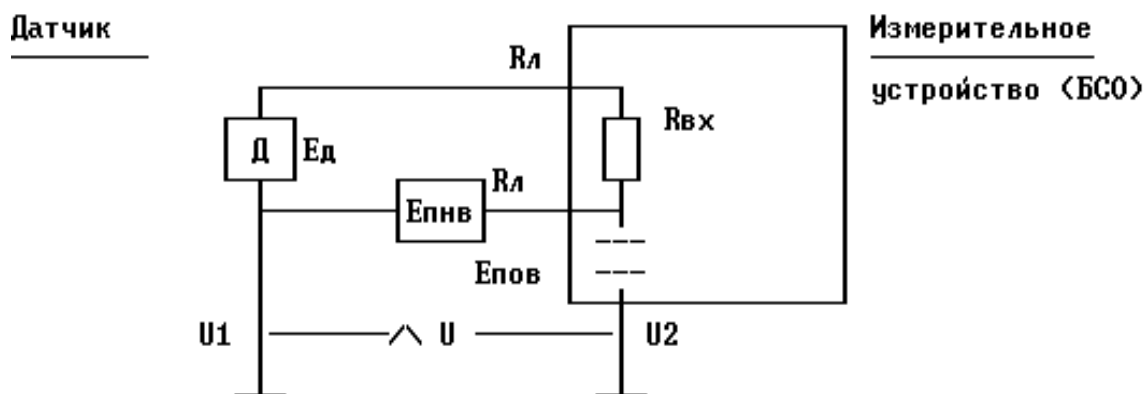


Рис. 2.34. Схема измерительного тракта с учетом помехи нормального вида.

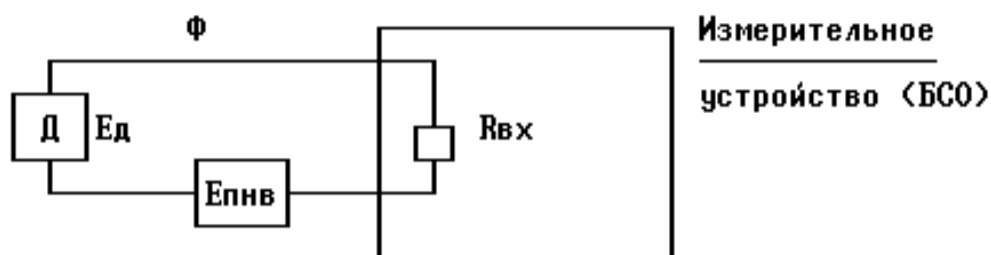


Рис. 2.35. Схема измерительного тракта с учетом помехи общего вида.

Кабель, не имеющий экранированных пар, более подвержен перекрестным помехам, чем с экранированными парами. Наиболее сильные перекрестные помехи возникают в том случае, когда одна из измерительных линий заземлена на одном конце кабеля, а вторая - на противоположном. Такая ситуация возникает тогда, когда к одному кабелю подключены заземленные и изолированные датчики. Для измерения помех в таком кабеле необходимо одну линию заземлять на стороне датчика, а на второй (соседней скрутке) производить измерение помех с помощью осциллографа, как было описано выше.

Измерение уровня помех желательно производить на различных типах кабелей (с общим изолированным экраном, со скрученными парами, общим экраном и т.п.), а затем уже выбирать тип кабеля, обеспечивающий необходимую защиту от помех. Для вновь строящихся объектов и в других случаях, когда оценить характер и влияние различных видов помех не представляется возможным, необходимо руководствоваться следующими рекомендациями:

- хорошей помехозащищенностью обладает кабель со скрученными парами и общим экраном, который имеет снаружи изоляцию (изолированный экран), исключающую касание экрана к металлоконструкциям, по которым производится прокладка кабелей. Стальная лента бронированных кабелей не является экраном;

- большей помехозащищенностью обладает кабель, имеющий скрученные пары, из которых каждая имеет экран, изолированный от экранов других пар. Желательно, чтобы кабель имел общий изолированный экран.

Во всех случаях, когда оценить влияние помех невозможно, измерительные цепи необходимо прокладывать отдельными кабелями в защитных трубах. Рекомендуется прокладывать эти кабели вдали от силовых кабелей и мощного электроэнергетического оборудования. Для уменьшения влияния помех на измерительные каналы ПТК необходимо правильно заземлить измерительные линии, выбрать помехоподавляющее оборудование

измерительных каналов. Для избежания появления на измерительной линии высоких блуждающих потенциалов, что может привести к пробоем полупроводниковых выходных ключей блоков связи с объектом, измерительная линия должна быть заземлена в одной точке цепи. Не допускается заземление измерительной цепи в двух и более точках. Экраны измерительной линии заземляются в той точке, в которой заземлена измерительная линия.

Выбор марки проводов, кабелей и необходимых сечений их жил производится в зависимости от способа и условий прокладки электрических проводок, токовых нагрузок, электрического сопротивления и падения напряжения в них, особенностей эксплуатации. Сигнальные провода и силовые линии не должны прокладываться одним кабелем.

Провода дискретных и аналоговых сигналов должны прокладываться отдельными кабелями. Для повышения помехозащищенности рекомендуется прокладывать такие кабели на расстоянии не менее 100 *mm* друг от друга. Для аналоговых сигнальных линий следует использовать экранированные кабели. Наилучших результатов дает прокладка кабелей в кабельных каналах, выполненных из проводящего материала. Кабельный канал не должен иметь изолирующих стыков, и оба его конца должны быть заземлены на контур защитного заземления. Необходимо, чтобы изоляция, защитные оболочки и наружные покрытия проводов и кабелей отвечали условиям окружающей среды и принятому способу выполнения электропроводок.

Если МСКУ эксплуатируется в условиях промышленного производства, отличающегося повышенным уровнем электрических помех, следует учитывать возможность отрицательного воздействия помех на его функционирование. Потенциальными источниками помех являются индуктивные нагрузки, такие как: реле, соленоиды, электродвигатели и их пускатели. На рис. 2.36 приведена схема подавления электрических помех, которая может быть использована для нейтрализации перенапряжений в нагрузке средней мощности (до 100 *W*), работающих от источника переменного тока. Для нейтрализации более мощных контакторов использование резистивно-емкостной цепи оказывается недостаточным. В подобных случаях эту цепь следует дополнить параллельно подключенным варистором, который ограничивает колебания переходного напряжения (рис. 2.37). Для источников помех, которые питаются постоянным напряжением, необходимо подключить диод параллельно нагрузке, как это показано на рис. 2.38.

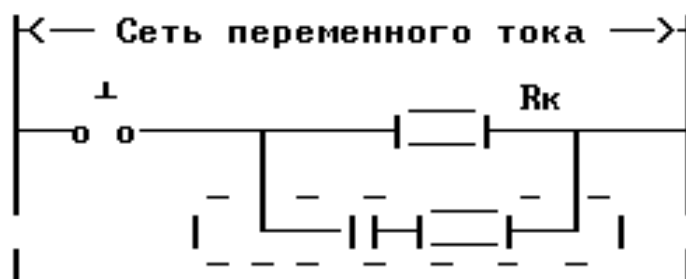


Рис. 2.36. Типовой вариант подавления электрических помех в индивидуальной нагрузке ($\leq 100 \text{ V}\cdot\text{A}$).

При выборе напряжения питания датчиков, с точки зрения улучшения помехозащищенности, предпочтительнее более высокое напряжение. При выборе напряжения источника питания с учетом его нестабильности должны выполняться следующие условия (для подсистем ввода дискретной информации):

$$(U_n - U_{\text{нест}}) - U_{\text{вх min}} > U_{\text{ном}} \quad (2.1)$$

$$(U_n + U_{\text{нест}}) + U_{\text{ном}} = U_{\text{вх max}} \quad (2.2)$$

где U_n - минимальное напряжение питания датчика; $U_{нест}$ - напряжение нестабильности источника питания; $U_{вх\ min}$ - минимальное входное напряжение модуля; $U_{пом}$ - напряжение помехи.

Если датчики дискретной информации позиционные, то длину линии связи можно рассчитать по формуле:

$$l = \frac{(U_n - U_{нест}) - U_{вх\ min} - U_{пом}}{2I_{вх} R_0} \quad (2.3)$$

где l - длина одного провода линии связи, km ; $I_{вх}$ - входной ток модуля, A ; R_0 - сопротивление жилы кабеля линии связи длиной $1\ km$, Ω/km .



Рис. 2.37. Типовой вариант подавления электрических помех в индивидуальной нагрузке большой мощности.

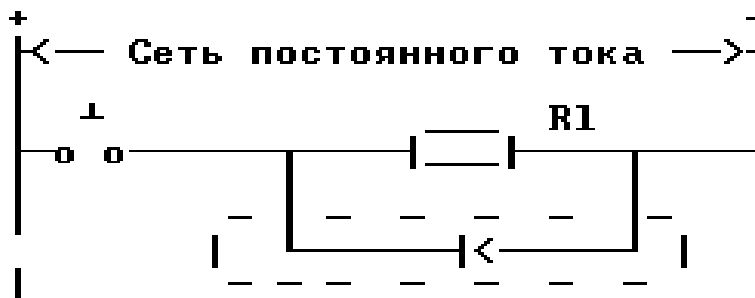


Рис. 2.38. Типовой вариант подавления электрических помех в индуктивной нагрузке постоянного тока.

Для датчиков типа "сухой контакт" длину линии связи можно определить по формуле:

$$l = \frac{150}{2 \cdot R_0} [km] \quad (2.4)$$

Примечание: Максимальное обратное напряжение диода должно, по меньшей мере, в два раза превышать приложенное напряжение постоянного тока.

Длина линии связи для модулей вывода дискретной информации рассчитывается по формуле:

$$l = \frac{U_{пит} - U_H}{2 \cdot R_0 \cdot I_H} [km] \quad (2.5)$$

где $U_{пит}$ - напряжение источника питания нагрузки, В; U_H - напряжение, падающее на нагрузке при номинальном токе, В; I_H - номинальный ток нагрузки, А.

Для формирователей аналоговой информации длина линии связи определяется следующим образом

$$l = \frac{R_{\max} - R_H - R_{доб}}{2R_0} [km], \quad (2.6)$$

где $R_{\max}$ - максимально допустимое сопротивление нагрузки датчика, равное 2000Ω ; R_H - сопротивление нагрузки исполнительного механизма, Ω ; $R_{доб}$ - добавочные сопротивления, предназначенные для отвода сигнала на контроль, на аналоговые показывающие и регистрирующие приборы и т.д., Ω .

В подсистемах ввода аналоговой информации для токовых каналов максимальная длина кабелей определяется допустимой величиной сопротивления нагрузки датчика за вычетом величины входного нормирующего сопротивления на приемной стороне, которое равно 499Ω для сигнала 5 мА и 249Ω - для 20 мА , а также значения входного сопротивления вторичных приборов последовательно включенных в измерительную цепь.

Если же по уравнению (2.1) напряжение помехи превышает допустимое значение, то необходимо применять дополнительные меры по уменьшению влияния помех, а именно: использовать изолированные датчики, фильтрацию измерительной информации, гальваническую развязку цепей, т.е. необходимо предусматривать в системах связи с объектом каналы с помехоподавляющим оборудованием.

В МСКУ практически все блоки связи с объектом, осуществляющие ввод и вывод аналоговых и дискретных сигналов, снабжены элементами, обеспечивающими устойчивость к воздействиям помех, как промышленной частоты (50 Hz), так и импульсным помехам повышенных частот. Для подавления помех нормального вида в цепях ввода аналоговых сигналов на панелях соединительных ПСд установлены помехоподавляющие RC-фильтры по каждому каналу. Для защиты входных цепей ввода дискретных сигналов по каждому каналу на ПСд установлены ограничители напряжения. В цепях подавления помех общего вида во всех блоках связи с объектом установлены элементы гальванического разделения входных-выходных цепей блока в виде оптронов или трансформаторов. При этом гальваническое разделение, в зависимости от типа блока и сигнала, осуществляется либо индивидуально по каждому каналу, либо по группе каналов. Наличие RC-фильтров в цепях ввода аналоговых сигналов обеспечивает подавление помех нормального вида $40, 66 \text{ dB}$ (в зависимости от типа блока). Коэффициент подавления помех общего вида для большинства блоков ввода/вывода аналоговых и дискретных сигналов достигает $100\text{-}120 \text{ dB}$. Испытательное напряжение гальванической развязки $500\div 2500 \text{ V}$.

Вопросы

1. Что входит в состав КТС АСУ ТП?
2. Какова роль контроллеров в АСУ ТП?
3. Какими основными параметрами характеризуются контроллеры?
4. Назовите предпосылки возникновения программируемых логических контроллеров.
5. В чем состоит эволюция ПЛК?
6. Охарактеризуйте ПЛК семейства Modicon.
7. Охарактеризуйте ПЛК семейства SIMATIC.
8. Охарактеризуйте ПЛК семейства GE Fanuc 90-70.
9. Охарактеризуйте контроллер Ломиконт.
10. Каковы причины широкого распространения ПЛК с архитектурой IBM PC?
11. Перечислите принципы открытой модульной архитектуры?
12. Охарактеризуйте IBM PC- совместимый ПЛК Ломиконт-ТМ.
13. Охарактеризуйте микроконтроллеры семейства ADAM.

14. Каковы основные особенности семейства микропроцессорных субкомплексов контроля и управления?
15. Какая проблемная ориентация МСКУ?
16. За счет чего достигается повышенная надежность МСКУ?
17. Каковы структурные особенности МСКУ?
18. Охарактеризуйте функциональные возможности МСКУ?
19. Какой состав блоков МСКУ?
20. Как осуществляется связь МСКУ с верхним уровнем ПТК?
21. Какие особенности управляющей системы МСКУ?
22. Какие функции управляющей системы МСКУ?
23. Назовите основные исполнения и технические характеристики МСКУ.
24. Каковы динамические характеристики МСКУ?
25. Какая роль КМп в МСКУ?
26. Каковы основные технические характеристики КМп?
27. Каково назначение функционального процессора и процессора связи?
28. Каковы назначение и характеристики ИР?
29. Какова роль промышленных сетей в ПТК?
30. Охарактеризуйте сетевые протоколы, для промышленных систем автоматизации нижнего уровня.
31. Какие физические линии связи применяются в промышленных сетях?
32. Каковы отличительные особенности PROFIBUS?
33. Каковы отличительные особенности сети МАПС?
34. Какие компоненты входят в сеть МАПС?
35. Какими средствами достигается повышение надежности сети МАПС?
36. Какие функции арбитра МАПС?
37. Какие функции контроллера связи КСв-31?
38. Охарактеризуйте принцип работы КСв-31.
39. Какие основные технические характеристики коммутатора HP ProCurve Switch 8000M?
40. Охарактеризуйте принцип работы коммутатора HP ProCurve Switch 8000M.
41. Каково назначение сети единого времени в ПТК?
42. В чем состоят особенности работы операторов-технологов АСУ ТП?
43. Охарактеризуйте подсистемы визуализации и обслуживания верхнего уровня АСУ ТП.
44. Назовите отличительные особенности рабочих станций ПС 5110.
45. Назовите состав рабочих станций ПС 5110.
46. Какие характеристики имеет системный блок ПС 5110?
47. Каковы конструктивные особенности ПС 5110?
48. Как организовано электропитание комплекса технических средств ПТК?
49. Как организовано электропитание и заземление МСКУ?
50. Какие требования к защитному заземлению ПТК?
51. От чего зависит уровень электромагнитных помех в измерительных линиях?
52. Что может быть источниками помех для ПТК?
53. Какие основные виды помех известны?
54. Каким образом можно определить уровни помех на действующих объектах?
55. Какие существуют способы подавления помех?
56. Какие способы подавления помех реализованы в МСКУ?

БАЗОВОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПТК

3.1 Операционные системы

Операционные системы (ОС) выполняют основные функции по интерфейсу с оператором, управлением запуском программ, распределению памяти, поддержке файловой системой и т.д. ОС, используемая в промышленных применениях, должна обладать повышенной надежностью, быстрым реагированием на какие-либо внешние события или изменения в параметрах управляемых процессов и обеспечивать многозадачный режим работы. ОС реального времени (ОС РВ) представляет собой операционную систему, разработанную специально для систем высокой реактивности, используемых в производстве и телекоммуникациях. Кроме того, ОС РВ, как правило, меньше по объему и количеству функций, аттестованы и надежнее универсальных. ОС РВ должна, в первую очередь, обеспечить гарантированное время реакции системы на событие. В зависимости от объекта (технологического процесса) время реакции должно находиться от нескольких микросекунд до нескольких десятых долей секунды. К другим показателям ОС РВ относятся надежность, гибкость и удобство в работе, наличие в них развитой среды разработки, графических интерфейсов, сетевой поддержки, возможность работы на многопроцессорных средствах.

Существует несколько десятков ОС РВ, разработанных для архитектуры IBM PC. Среди них наиболее известны OS-9, RTMX, AMX, RTKernel, FlexOS, VRTX, QNX и др. [36]. Широко известная ОС WINDOWS NT не в полной мере отвечает требованиям систем реального времени (в частности, по гарантированному времени реакции на событие) и поэтому может быть использована лишь на верхнем уровне АСУ ТП - в операторских станциях. Существует два типа ОС РВ. Первый тип ОС представляет собой ядро реального времени. Ядра реального времени предоставляют пользователю такие базовые функции, как планирование и синхронизация задач, межзадачные коммуникации, управление памятью и т.п. Второй тип ОС - собственно операционные системы - в дополнение к этому имеют файловую систему, сетевую поддержку, интерфейс с оператором и другие средства высокого уровня. Четкой границы между этими типами нет. К системам первого типа можно отнести RTKernel, AMX, к системам второго - OS-9, QNX.

По своей внутренней структуре ОС РВ можно условно разделить на следующие: монолитные, ОС на основе микроядра и объектно-ориентированные. При проектировании ОС РВ все чаще используется идеология микроядра. Такой подход увеличивает надежность программного обеспечения и позволяет использовать только те компоненты операционной системы, которые необходимы в каждом конкретном случае. Например, модуль, управляющий файловой системой, запускается как одна из задач и может быть легко удален. Все драйверы также функционируют как независимые задачи. Типовой размер микроядра операционных систем - менее 10 КВ.

Когда возникает проблема выбора необходимой ОС РВ, то одним из первых требований, которое должен определить разработчик, является время реакции на воздействия процессов РВ. В этом смысле производительность имеющихся ОС РВ лежит в очень широких пределах и, в общем случае, обратно пропорциональна их сложности и объему выполняемых функций. Причем, как правило, приводимые характеристики производительности ОС касаются только функций, выполняемых на уровне ядра системы. Наивысшая производительность лежит в области ядер РВ, а наименьшая - в области полнофункциональных резидентных ОС РВ.

С ростом сложности и объема ПО РВ в создаваемой системе недостающие функции вынужденно реализуются в составе прикладного ПО. Это часто приводит к значительным трудностям в реализации, к потере преимуществ производительности базовой ОС, ухудшению свойств переносимости и обслуживания разработанного ПО РВ.

3.1.1 Стандартизация ОС РВ

Существует два подхода к переносимости программного обеспечения: поддержка одной ОС одновременно нескольких аппаратных платформ и обеспечение стандартного интерфейса между прикладными программами и ОС. В качестве такого стандарта используется программный интерфейс, разработанный для UNIX - подобных операционных систем и получивший название POSIX (Portable Operating System Interface for UNIX). POSIX - это развивающийся набор стандартов на операционные системы, каждый из которых охватывает различные аспекты [37]. Их можно разбить на три группы. К первой группе относятся *базовые стандарты*, ко второй – *языковые интерфейсы* и к третьей – *операционная среда открытых систем*.

В *базовых стандартах* определяются системные интерфейсы, связанные с различными сторонами операционных систем. Базовый стандарт специфицирует синтаксис и семантику системного интерфейса таким образом, что прикладные программы могут непосредственно обращаться к системным службам. Способ реализации этих служб в базовом стандарте не определяется; системные разработчики свободны в выборе способа реализации.

Языковые интерфейсы - эти стандарты, которые обеспечивают действующие интерфейсы к различным языкам программирования операционных систем.

Операционная среда открытых систем. Эти стандарты включают сведения по операционной среде POSIX и прикладным профилям. Прикладной профиль - это стандарты POSIX вместе с опциями и параметрами, которые необходимы для конкретной прикладной среды. Программа, написанная в соответствии со стандартом POSIX, должна работать на любой аппаратной платформе при условии применения POSIX - совместимой операционной системы. В настоящее время существует четыре стандарта для приложений реального времени.

3.1.1.1. Стандарт POSIX.4

Этот стандарт определяет системные интерфейсы для поддержки приложений реального времени. В нем специфицируются три механизма диспетчеризации параллельных процессов, удовлетворяющих специфике реального времени. У каждого процесса есть атрибут диспетчеризации, который должен устанавливаться в соответствии с одним из трех механизмов:

- **SCHED_FIFO** - это механизм приоритетной диспетчеризации с фиксированными приоритетами, при которой процессы с одинаковыми приоритетами обрабатываются по принципу «первым пришел - первым вышел» (FIFO), этот механизм должен обслуживать минимум 32 уровня приоритетов;

- **SCHED_RR**. - этот механизм подобен механизму SCHED_FIFO, но в нем для диспетчеризации равноприоритетных процессов используется метод квантования времени (круговая диспетчеризация), этот метод обладает 32 уровнями приоритетов;

- **SCHED_OTHER**. - этот механизм диспетчеризации определяется конкретной реализацией.

Используя эти методы диспетчеризации, а также функции установки приоритета каждого процесса и функции включения (разрешения) нужного метода, в операционных системах в стандарте POSIX можно выполнять диспетчеризацию приложений реального времени.

Блокирование виртуальной памяти. Использование виртуальной памяти очень эффективно при работе программ, не относящихся к программам реального времени, но приводит к непредсказуемости времени реакции системы. Для того чтобы ограничить время доступа к памяти, в стандарте POSIX.4 определяются функции блокировки (фиксации) в памяти всего адресного пространства процесса или отдельных его областей. Эти функции используются для критичных ко времени процессов, а также для процессов, с которыми синхронизируются критичные ко времени процессы. В этом случае время их реакции может быть предсказуемым.

Синхронизация процессов. В стандарте POSIX.4 определяются функции управления синхронизацией процессов с помощью семафоров-счетчиков. Эти семафоры идентифицируются по имени, находящемуся в некотором пространстве имен, определенном при реализации стандарта. *Семафор-счетчик* - это общий механизм синхронизации, который позволяет реализовать взаимно исключающий доступ к разделяемым ресурсам, передачу сигналов, ожидание процессов и другие механизмы синхронизации. Одним из наиболее распространенных применений семафоров является совместное использование данных процессами. Согласно POSIX.4, это можно реализовать с помощью объектов разделяемой памяти, используемых вместе с семафорами. Семафоры-счетчики допускают так называемую «инверсию приоритетов», при наступлении которой высокоприоритетный процесс вынужден ожидать завершения некоторого действия, выполняемого процессом с более низким приоритетом. Для задач жесткого реального времени такие длительные задержки обычно неприемлемы. Для предотвращения таких ситуаций можно использовать специальные протоколы, позволяющие процессу резервировать конкретный ресурс для монопольного использования. Другим механизмом синхронизации, который предотвращает инверсию приоритетов, является так называемый *мутекс* (mutex).

Разделяемая память. В стандарте POSIX.4 определяются объекты разделяемой памяти, представляющие собой участки физической памяти, которые могут отображаться на адресное пространство процесса. Когда два или несколько процессов отображают один и тот же объект, они разделяют связанный с ним участок памяти. Как и в случае семафоров, объекты разделяемой памяти идентифицируются по имени, принадлежащему некоторому пространству имен, которое определяется при реализации стандарта. Если к объектам данных, находящихся в разделяемой памяти, нужен взаимно исключающий доступ, для управления этим доступом можно использовать семафоры. В адресное пространство процесса могут отображаться и файлы - для совместного использования с другими процессами.

Сигналы реального времени. Так как многие системы реального времени должны обеспечивать быстрый обмен событиями, интерфейс сигналов в стандарте расширен для получения следующих возможностей: сигналы реального времени устанавливаются в очередь, поэтому события не теряются; необработанные сигналы реального времени извлекаются из очереди по приоритетам, где в качестве приоритета служит номер сигнала, это предоставляет возможность быстрого отклика на события, требующие неотложной реакции; сигналы реального времени содержат дополнительное поле данных, которое может использоваться прикладной системой для обмена между генератором сигнала и его обработчиком; расширен диапазон, который доступен прикладной системе сигналов.

Взаимодействие процессов. Для взаимодействия процессов определяется простой механизм очередей сообщений. Очереди сообщений идентифицируются по имени, принадлежащему некоторому пространству имен, определяемому при реализации стандарта. Сообщения имеют связанное с ними поле приоритета и извлекаются из очереди в соответствии с приоритетом. Это способствует уменьшению вероятности возникновения инверсии приоритета, возникающего в системе. Передача и получение сообщений могут блокироваться и разблокироваться; передача и получение не синхронизируются, то есть отправитель не ждет, когда получатель действительно извлечет сообщение из очереди. Максимальный размер сообщений и очередей определяется пользователем, а необходимые для поддержания очереди ресурсы могут выделяться заранее во время разработки приложения; это повышает предсказуемость работы с очередями сообщений.

Часы и таймеры. Определяемые стандартом часы реального времени должны обеспечивать разрешение минимум 20 ms. Для отсчета временных интервалов на основе часов реального времени могут создаваться таймеры. По истечении заданного интервала времени они генерируют сигнал, направленный процессу, создавшему данный таймер. Существует несколько опций таких как: периодическая сигнализация, единичный сигнал и т.д., которые позволяют реализовать генерацию периодических событий. Для приостановки вызывающего

процесса на некоторый заданный период времени определяется функция относительного «сна» (nanosleep).

Асинхронный ввод/вывод. В стандарте определяются функции, которые обеспечивают возможность совмещать прикладную обработку и операции ввода/вывода, инициированные данным приложением. Асинхронные операции ввода/вывода подобны обычным операциям ввода/вывода, за исключением того, что после того, как процесс инициировал асинхронную операцию ввода/вывода, он продолжает выполняться параллельно этой операции. Когда операция завершается, данному приложению может быть послан сигнал.

В стандарте определяются и другие функции, такие как синхронизированный ввод/вывод, файлы реального времени и т.п. Детальное описание этих функций изложено в [37].

3.1.1.2. Threads-расширение

В стандарте POSIX.4a определяется интерфейс для поддержки нескольких параллельных действий внутри каждого POSIX-процесса. Этот интерфейс называется «нитеями» (threads). Нити, определяемые в POSIX.4a, имеют некоторое состояние взаимодействия, которое обладает меньшим контекстом, чем состояние процесса. Все нити внутри одного процесса разделяют одно и то же адресное пространство. Их можно реализовать с меньшими временами контекстных переключений и временами создания/уничтожения, по сравнению с аналогичными временами работы процессов. Стандарт POSIX.4a специально разработан для удовлетворения нужд разделения памяти при многопроцессорной работе. По своим характеристикам такая модель «нитей» в большей степени приближена к модели параллельности, используемой в коммерческих ядрах реального времени, чем модель процессов.

Управление нитями. Эти функции позволяют управлять созданием и завершением выполнения нитей, а также связанными с ними операциями. Определяются функции создания, ожидания завершения, нормального завершения, открепления нити (то есть, связанная с нитью память может после завершения нити перераспределяться) или создания конкретной нити только в том случае, если она еще не была создана. Другие функции позволяют управлять идентификаторами нитей. Определяются также функции для управления атрибутами при создании нитей, такими, как размер стека, возможность перераспределения памяти, занимаемой нитью после ее создания, и т.п.

Диспетчеризация нитей. Для нитей определяются те же методы диспетчеризации, что и для процессов в стандарте POSIX.4 (приоритетное вытеснение с FIFO- или круговым обслуживанием равноприоритетных нитей). Так как в системе могут существовать два планировщика (диспетчера) - планировщик процессов и планировщик нитей - определяется концепция конкурентного пространства. Конкурентное пространство некоторой нити - это набор нитей, с которыми она конкурирует за центральный процессор. При реализации может возникнуть три основных вида различных конкурентных пространств:

- глобальная диспетчеризация - все нити имеют глобальное конкурентное пространство. Диспетчеризация каждой нити производится с учетом всех остальных нитей, имеющих в системе, независимо от того, какому процессу они принадлежат, планировщик работает только на уровне нитей, а параметры диспетчеризации процессов игнорируются;

- локальная диспетчеризация - нити конкурируют только с другими нитями того же самого процесса. Диспетчеризация производится на двух уровнях, сначала диспетчируются все процессы по отношению друг к другу, затем нити выбранного процесса конкурируют друг с другом за центральный процессор;

- смешанная диспетчеризация - некоторые нити имеют глобальное конкурентное пространство, а другие – локальное. Диспетчеризация проводится на двух уровнях: на первом уровне диспетчеризации подвергаются процессы и глобальные нити; на втором уровне происходит диспетчеризация локальных нитей выбранного процесса.

Наилучшие результаты для приложений реального времени дает глобальная и смешанная диспетчеризация, так как в этих случаях можно производить диспетчеризацию на одном и том

же уровне всех конкурирующих объектов, обладающих жесткими временными ограничениями. В системах со смешанной диспетчеризацией диспетчеризация отдельных нитей может осуществляться и локально. Локальная диспетчеризация быстрее и эффективнее глобальной. Эту возможность следует использовать только для групп нитей, глобальный приоритет которых меньше или больше приоритетов остальных групп нитей, имеющих в системе (когда ни одна из других нитей в системе не имеет уровня приоритета, попадающего между уровнями приоритетов каких-либо нитей этой группы). Для диспетчеризации группы нитей с локальным конкурентным пространством будет использоваться приоритет процесса, а не приоритеты нитей. Это относится и к системам с локальной диспетчеризацией.

Синхронизация нитей. Для нитей определяется два примитива синхронизации: мутексы и условные переменные. Мутексы используются для синхронизации взаимно исключающего доступа нитей к разделяемым ресурсам, а условные переменные используются для сигнализации и ожидания событий среди нитей. Ожидание условной переменной, с которой должен быть связан сигнал, можно задать с помощью тайм-аута. Эти примитивы можно использовать для нитей, относящихся к разным процессам.

Мутексы определяются с помощью трех альтернативных протоколов синхронизации:

- **NO_PRIO_INHERIT** - приоритет нити не зависит от ее владения мутексом;
- **PRIO_INHERIT** - нить, владеющая некоторым мутексом, наследует приоритеты нитей, ожидающих захвата этого мутекса, это протокол наследования приоритетов;

- **PRIO_PROTECT** - когда нить блокирует захваченный ею мутекс, она наследует приоритет, соответствующий данному мутексу, этот приоритет определяется прикладной программой как атрибут мутекса; в случае использования соответствующего значения, равного наименьшему целому числу, превышающему значения приоритетов, получается протокол приоритетной защиты, называемый также эмуляцией протокола с перекрытием приоритета [38].

Используя один из двух последних протоколов можно избежать инверсии приоритета, что увеличивает эффективность работы системы с жесткими требованиями к характеристикам реального времени. Протокол приоритетной защиты вместе с подходящими определениями значения перекрытия приоритета можно использовать и для избежания особого вида инверсии приоритета, возникающего в многопроцессорной среде и называемого удаленным блокированием.

Для управления специфическими данными, связанными с нитями, для завершения нитей, для отправки сигналов нитям и управления реентерабельными функциями, в стандарте POSIX.4a определяются и другие функции.

3.1.1.3 Дополнительные расширения стандарта POSIX.4b для приложений реального времени

В стандарте определяются дополнительные расширения, необходимые для поддержки переносимости приложений реального времени.

Тайм-ауты. Некоторые утилиты операционной системы могут приостанавливать процесс на неопределенное время, пока не освободятся и не станут доступными необходимые ресурсы. В системах, критичных по времени, необходимо ограничить максимальное время, в течение которого процесс может ожидать завершения работы одной из таких утилит. В POSIX.4b определяются новые версии некоторых из блокирующих утилит со встроенными таймаутами. Последние задают максимальное время, в течение которого процесс может приостанавливаться в ожидании завершения работы такой утилиты. К таким утилитам относятся: ожидать семафор для разблокирования, ожидать прибытия сообщения в очередь сообщений, послать сообщение для постановки его в очередь, ожидать мутекс для разблокирования.

Часы времени выполнения. Для управления часами времени выполнения в стандарте POSIX.4 используются интерфейсы часов и таймеров. На базе этих часов могут определяться таймеры, они могут измерять излишний расход времени процессом или нитью, позволяя динамически выявлять программные ошибки или ошибки вычисления максимально

возможного времени выполнения. Часы времени выполнения позволяют выявить возникновение перерасхода времени и способствовать выявлению некорректностей в системе.

Спорадический сервер. В стандарте определяется новый механизм диспетчеризации (SCHED_SPORADIC), в котором реализуется алгоритм диспетчеризации спорадического сервера. Этот механизм может использоваться для обработки аperiодических событий на нужном уровне приоритета. При этом гарантированно удовлетворяются временные требования, предъявляемые к низкоприоритетным задачам. Спорадический сервер обеспечивает быстрое время отклика и делает поведение системы, в которой обрабатываются аperiодические события, предсказуемым.

Управление прерываниями. Системы реального времени должны иметь возможность захватывать прерывания, сгенерированные устройствами, и обрабатывать их в прикладной программе. Предложенные в стандарте функции позволяют процессу или нити обслужить прерывание с помощью разработанной пользователем подпрограммы обработки прерываний, блокироваться в ожидании наступления прерывания и защитить критичные участки кода от прерывания. Из-за множества отличий в обработке прерываний в разных архитектурах, определяемые интерфейсы не позволяют достигнуть полной переносимости прикладных программ. Однако с помощью данного интерфейса, благодаря эталонной модели, переносимость приложений улучшается. В этом случае количество изменений, необходимых для переносимости приложения, уменьшается.

Управление устройствами ввода/вывода. В системах реального времени взаимодействие со средой происходит через драйверы устройств, которые создаются разработчиком приложения. В POSIX.4b определяется функция, которая позволяет прикладной программе передать управляющую информацию в драйвер и из драйвера устройства. При использовании стандартного эталонного интерфейса взаимодействия с драйверами переносимость прикладных программ улучшается.

3.1.1.4. Профили прикладных контекстов реального времени

Стандарт POSIX вместе с расширениями позволяет реализовывать большие системы реального времени. Однако для некоторых относительно небольших систем реального времени было бы предпочтительнее использование только некоторого подмножества этого стандарта. Подмножества возможностей базовых стандартов, которые нужны для конкретного прикладного контекста, даются в стандарте POSIX.13 как профили прикладных процессов реального времени (Application Environment Profiles, AEP). В POSIX.13 определяются четыре AEP реального времени:

- минимальная система - соответствует небольшой встроенной системе без диспетчера памяти, файловой системы и терминала ввода/вывода, разрешается только один процесс с возможностью параллельного выполнения нескольких нитей;
- контроллер реального времени - соответствует управляющей системе специального назначения, подобен минимальному профилю реального времени, но с добавлением файловой системы и терминала ввода/вывода, разрешается только один процесс с несколькими нитями;
- специализированная система - соответствует большой встроенной системе без файловой системы, в ней выполняется несколько процессов и нитей;
- многоцелевая система - соответствует большой системе реального времени со всеми поддерживаемыми функциями.

С помощью профилей реального времени можно реализовать множество систем реального времени с различной аппаратной поддержкой, соответствующих стандарту POSIX. Прикладные программы будут переносимы между этими системами, при условии, что они соответствуют тем же профилям или новая платформа включает все возможности предыдущей. Например, приложение можно будет перенести из минимальной системы в контроллер реального времени или в специализированную систему реального времени. Приложение, которое предназначено

для выполнения в небольшой специализированной системе, для отладочных целей можно запускать в полноценной системе.

Стандарт POSIX позволяет создавать поддающиеся анализу и предсказуемые системы реального времени, и переносимые прикладные программы.

3.1.2 Основные свойства ОС РВ

Многозадачные ОС РВ оперируют с таким объектом как «задача». Обычно под задачей понимается набор операций (машинных инструкций), предназначенных для выполнения логически законченной функции системы. Задача конкурирует с другими задачами за получение контроля над ресурсами вычислительной системы. Как правило, различают две разновидности задач: *процессы* и *потoki*. Процесс представляет собой отдельный загружаемый программный модуль (файл), который во время исполнения имеет в памяти свои независимые области для кода и данных. В отличие от этого, потоки могут пользоваться общими участками кода и данных в рамках единого программного модуля.

Множество потоков способно размещаться внутри одного EXE-модуля, это позволяет экономить ресурсы как внешней, так и внутренней памяти. Использование потоками общей области памяти позволяет эффективно реализовать межзадачный обмен сообщениями (достаточно передать указатель на сообщение). Процессы не имеют общей области памяти, поэтому ОС должна целиком скопировать сообщение из области памяти одной задачи в область памяти другой, либо должны быть предусмотрены специальные механизмы, которые позволили бы одной задаче получить доступ к сообщению из области памяти другой задачи. Как правило, контекст потоков меньше, чем контекст процессов, а значит, время переключения между задачами потоками меньше, чем между задачами-процессами. Так как все потоки, а иногда и само ядро РВ размещаются в одном EXE-модуле, значительно упрощается использование программ-отладчиков. Как правило, потоки не могут быть подгружены динамически. Чтобы добавить новый поток, необходимо произвести соответствующие изменения в исходных текстах и перекомпилировать приложение. Процессы в отличие от потоков, подгружаемы, что позволяет динамически изменять функции системы в процессе ее работы. Кроме того, так как процессам соответствуют отдельные программные модули, они могут быть разработаны различными компаниями, чем достигается дополнительная гибкость и возможность использования ранее разработанного программного обеспечения. То, что потоки имеют доступ к областям данных друг друга, может привести к ситуации, когда некорректно работающий поток способен испортить данные другого потока. В отличие от этого, процессы защищены от взаимного влияния, а попытка записать в «не свою» память приводит, как правило, к возникновению специального прерывания по обработке «исключительных ситуаций».

Реализация механизмов управления процессами и потоками, возможность их взаимного сосуществования и взаимодействия определяются конкретным ПО РВ [38].

3.1.2.1. Основные особенности задач систем реального времени

Как правило, вся важная, с точки зрения операционной системы, информация о задаче хранится в унифицированной структуре данных - управляющем блоке (Task Control Blok, TCB). В нем хранятся такие параметры, как имя и номер задачи, верхняя и нижняя границы стека, ссылка на очередь сообщений, статус задачи, приоритет и т.п. Приоритет используется, в основном, планировщиком задач для определения того, какая из готовых к работе задач должна получить управление. Различают системы с динамической и статической приоритетностью. В первом случае приоритет задач может меняться в процессе исполнения, в то время как во втором приоритет задач жестко задается на этапе разработки или во время начального конфигурирования системы.

Контекст задачи - это набор данных, содержащий всю необходимую информацию для возобновления выполнения задачи с того места, где она была ранее прервана. Часто контекст хранится в ТСВ и включает в себя такие данные, как счетчик команд, указатель стека, содержимое регистров и т.п. Планировщик задач в случае необходимости сохраняет контекст текущей активной задачи и восстанавливает контекст задачи, назначенной к исполнению. Такое переключение контекстов является, по сути, основным механизмом ОС РВ при переходе от выполнения одной задачи к другой.

Состояние (статус) задачи. С точки зрения операционной системы, задача может находиться в нескольких состояниях. Число и название этих состояний различаются от одной ОС к другой. Тем не менее, практически в любой ОС РВ загруженная на выполнение задача может находиться, по крайней мере, в трех состояниях:

- активная задача - это задача, выполняемая системой в текущий момент времени.
- готовая задача - это задача, готовая к выполнению и ожидающая запуска на исполнение.
- блокированная задача - это задача, выполнение которой приостановлено до наступления определенных событий.

Таковыми событиями также могут быть: освобождение необходимого задаче ресурса, поступление ожидаемого сообщения, завершение интервала ожидания и т.п.

Пустая задача (Idle Task) - это задача, запускаемая самой операционной системой в момент инициализации и выполнения только тогда, когда в системе нет других готовых для выполнения задач. Пустая задача запускается с самым низким приоритетом и, как правило, представляет собой бесконечный цикл «ничего не делать». Наличие пустой задачи предоставляет операционной системе удобный механизм отработки ситуаций, когда нет ни одной готовой к выполнению задачи.

Многократный запуск задач. Многозадачные ОС РВ позволяют запускать несколько копий одной и той же задачи. При этом для каждой такой копии создается свой ТСВ и выделяется своя область памяти. В целях экономии памяти может быть предусмотрено совместное использование одного и того же исполняемого кода для всех запущенных копий. В этом случае программа должна обеспечить повторную входимость (реентерабельность). Кроме того, программа не должна использовать временные файлы с фиксированными именами и должна корректно осуществлять доступ к глобальным ресурсам.

Реентерабельность означает возможность без негативных последствий временно прервать выполнение какой-либо функции или подпрограммы, а затем вызвать эту функцию или подпрограмму снова. Частным проявлением реентерабельности служит *рекурсия*, когда тело подпрограммы содержит вызов самой себя. Классическим примером нереентерабельной системы является MS DOS, а типичной причиной нереентерабельности служит использование глобальных переменных. Ошибки, связанные с нереентерабельностью, трудно обнаружить, а последствия они могут вызвать самые катастрофические.

3.1.2.2 Планирование задач

Важной частью любой ОС РВ является планировщик задач. Несмотря на то, что в разных источниках он может называться по-разному (диспетчер задач, супервизор и т.п.), его функции остаются теми же: определить, какая из задач должна выполняться в системе в каждый конкретный момент времени. Самым простым методом планирования, не требующим никакого специального ПО и планировщика как такового, является использование циклического алгоритма.

Каждая задача, представляющая собой отдельную подпрограмму, выполняется циклически. При этом должны соблюдаться следующие правила:

- подпрограммы не должны содержать циклов ожидания;
- подпрограммы должны выполнять свою работу как можно быстрее, чтобы дать возможность работать следующей подпрограмме;

- при необходимости подпрограмма может сохранять свое окружение и текущие результаты, чтобы в следующем цикле возобновить работу с того же места.

Можно отметить следующие преимущества циклического алгоритма:

- простота использования и легкость понимания. Если исключить из рассмотрения прерывания, система полностью детерминирована;

- задачи всегда вызываются в одной и той же последовательности, что позволяет достаточно просто произвести анализ «наихудшего случая» и вычислить максимальную задержку;

- минимальные размеры кода и данных.

Кроме того, в отличие от алгоритмов с вытеснением, для всех задач необходим только один стек. Отсутствуют ошибки, обусловленные «гонками».

К недостаткам циклического алгоритма можно отнести отсутствие приоритетности и очередей. К тому же задачи вызываются независимо от того, должны ли они в данный момент что-либо делать или нет, а на прикладного программиста ложится максимальная ответственность за работоспособность системы.

Одним из алгоритмов планирования является режим разделения времени. Существуют различные реализации этого алгоритма. Как правило, он реализуется следующим образом. Каждой задаче отводится определенное количество квантов времени, в течение которых задача может монопольно занимать процессорное время. После того как заданный интервал времени истекает, управление передается следующей готовой к выполнению задаче, имеющей наивысший приоритет. Та, в свою очередь, выполняется в течение отведенного для нее промежутка времени, после чего все повторяется в стиле round robin. Такой алгоритм работы может привести к определенным проблемам: низкоприоритетные задачи могут никогда не получить управление, так как существует вероятность того, что высокоприоритетные задачи будут делить все процессорное время между собой. Единственная возможность для получения управления низкоприоритетных задач предоставляет ситуация, когда все высокоприоритетные задачи находятся в заблокированном состоянии. Для решения этой проблемы применяется прием, получивший название *равнодоступность* (fairness). При этом реализуется принцип адаптивной приоритетности, когда приоритет задачи, которая выполняется слишком долго, постепенно уменьшается, позволяя менее приоритетным задачам получить свою долю процессорного времени. Равнодоступность применяется главным образом в многопользовательских системах и редко применяется в системах реального времени.

Кооперативная многозадачность - это еще один алгоритм переключения задач. В данном алгоритме задача, получившая управление, выполняется до тех пор, пока она сама по своей инициативе не передаст управление другой задаче. По сути, это продолжение идеологии round robin. В чистом виде алгоритм кооперативной многозадачности мало применяется в системах реального времени.

Приоритетная многозадачность с вытеснением - это наиболее часто используемый в операционных системах реального времени принцип планирования. Основная идея этого принципа состоит в том, что высокоприоритетная задача, как только для нее появляется работа, немедленно прерывает (вытесняет) низкоприоритетную. Если какая-либо задача переходит в состояние готовности, она немедленно получает управление, если текущая активная задача имеет более низкий приоритет. Такое «вытеснение» происходит, например, когда высокоприоритетная задача получила ожидаемое сообщение (освободился запрошенный ею ресурс, произошло связанное с ней событие, исчерпался заданный интервал времени и т.п.).

Необходимо отметить, что диапазон систем реального времени весьма широк, начиная от полностью статических систем, где все задачи и их приоритеты заранее определены, до динамических, где набор выполняемых задач, их приоритеты и даже алгоритмы планирования могут меняться в процессе функционирования. Существуют, например, системы, где каждая отдельная задача может участвовать в любом из трех алгоритмов планирования или их комбинации (вытеснение, разделение времени, кооперативность). В общем случае алгоритмы планирования должны соответствовать критериям оптимальности функционирования системы.

В зависимости от критериев оптимальности могут применяться алгоритмы планирования задач, отличные от рассмотренных. Например, может оказаться, что планировщик должен анализировать момент выдачи критичных по времени управляющих воздействий и запускать на выполнение ту задачу, которая отвечает за ближайшие из них (алгоритм *earliest deadline first*, EDF). Как правило, разработчики систем реального времени стараются свести свою систему к наиболее простым конфигурациям, характерным для систем «жесткого» реального времени, иногда даже в ущерб эффективности использования вычислительных ресурсов. Сложные динамические системы весьма трудно анализировать и отлаживать, поэтому лучше использовать более мощный процессор, чем иметь в последствии проблемы из-за непредвиденного поведения системы. В связи с этим большинство существующих систем представляют собой статические системы с фиксированными приоритетами. Часто в системе реализуется несколько режимов работы, каждый из которых имеет свой набор выполняемых задач с заранее заданными приоритетами. Значительная часть особо ответственных систем реализуется вообще без применения коммерческих ОС РВ.

3.1.2.3 Синхронизация задач

Хотя каждая задача в системе, как правило, выполняет какую-либо отдельную функцию, часто возникает необходимость в согласованности действий, выполняемых различными задачами (синхронизации). Такая синхронизация необходима, в основном, в следующих случаях:

- функции, выполняемые различными задачами, связаны друг с другом. Например, если одна задача подготавливает исходные данные для другой, то последняя не выполняется до тех пор, пока не получит от первой задачи соответствующего сообщения. Одна из вариаций в этом случае - это когда задача при определенных условиях порождает одну или несколько новых задач;
- необходимо упорядочить доступ нескольких задач к разделяемому ресурсу;
- необходима синхронизация задачи с внешними событиями. В основном для этого используется механизм прерываний;
- необходима синхронизация задачи по времени. Диапазон различных вариантов в этом случае достаточно широк: от привязки момента выдачи какого-либо воздействия к точному астрономическому времени до простой задержки выполнения задачи на определенный интервал времени. Для решения этих вопросов в конечном итоге используются специальные аппаратные средства, называемые таймером.

Взаимное согласование задач с помощью сообщений является одним из важнейших принципов операционных систем реального времени. Способы реализации межзадачного обмена отличаются большим разнообразием. Можно встретить такие понятия, как сообщение (*message*), почтовый ящик (*mail box*), сигнал (*signal*), событие (*event*) и т. п.

Объем информации, передаваемой в сообщениях, может меняться от 1 *bit* до всей свободной памяти системы. Во многих ОС РВ компоненты операционной системы, так же как и пользовательские задачи, способны принимать и передавать сообщения. Сообщения могут быть асинхронными и синхронными. В первом случае доставка сообщений задаче производится после того, как она в плановом порядке получит управление, а во втором случае циркуляция сообщений оказывает непосредственное влияние на планирование задач. Например, задача, пославшая какое-либо сообщение, немедленно блокируется, если для продолжения работы ей необходимо дождаться ответа, или если низкоприоритетная задача посылает высокоприоритетной задаче сообщение, которого последняя ожидает, то высокоприоритетная задача, если используется приоритетная многозадачность с вытеснением, немедленно получает управление. Иногда сообщения передаются через отведенный для этого буфер определенного размера (почтовый ящик). При этом обычно новое сообщение затирает старое, даже если последнее не было обработано.

Однако наиболее часто используется принцип, когда каждая задача имеет свою очередь сообщений, в конец которой ставится всякое вновь полученное сообщение. Стандартный принцип обработки очереди сообщений по принципу «первым вошел, первым вышел» (FIFO) не всегда оптимально соответствует поставленной задаче. В некоторых ОС РВ предусматривается такая возможность, когда сообщение от высокоприоритетной задачи обрабатывается в первую очередь (сообщение наследует приоритет пославшей его задачи). Иногда полезным оказывается непосредственное управление приоритетом сообщений.

Очень важной есть правильная организация взаимодействия различных задач при доступе к общему ресурсу. *Ресурс* - это общий термин, описывающий физическое или логическое устройство, который может одновременно использоваться только одной задачей. Процессорное время тоже представляет собой конкурентно используемый ресурс вычислительной системы. При использовании общего ресурса несколькими задачами возможно появление некорректных ситуаций, обусловленных времязависимыми ошибками (time dependent error), или «гонками». Они характерны для многозадачных ОС, применяющих алгоритмы планирования с вытеснением (системы с разделением времени также можно отнести к категории «вытесняющих»). Ошибки, обусловленные «гонками», характерны для работы с любыми ресурсами, доступ к которым имеют несколько задач, и происходят только в результате совпадения определенных условий, а поэтому с трудом обнаруживаются на этапе отладки. Возможны следующие пути решения такой проблемы:

- не использовать алгоритмы планирования задач с вытеснением. Это решение может быть не всегда приемлемо;

- использовать специальный сервер ресурса, то есть задачу, ответственную за упорядочивание доступа к ресурсу. В этом случае запрос на изменение глобальных данных посылается этому серверу в виде сообщения. Аналогичный подход применим и для физических устройств;

- запретить прерывания на время доступа к разделяемым данным. В системах реального времени не всегда это возможно;

- использовать для упорядочивания доступа к глобальным данным семафоры. Это наиболее часто встречаемое решение в системах реального времени. Понятие «семафор» достаточно хорошо описано в литературе, например [39].

Участки кода программ, где происходит обращение к разделяемым ресурсам, называются *критическими участками*. Так как процессы обычно не имеют доступа к данным друг друга, а ресурсы физических устройств управляются специальными задачами (драйверами), наиболее типична ситуация, когда «гонки» за доступ к глобальным переменным устраивают различные потоки, исполняемые в рамках одного программного модуля. Для того чтобы гарантировать, что критическая секция кода выполняется в каждый момент времени только одним потоком, используют механизм взаимоисключающего доступа, называемого мутексом (Mutual Exclusion Locks, Mutex). Практически мутекс представляет собой разновидность семафора, который сигнализирует другим потокам, что критическая секция кода кем-то уже выполняется. Если мутекс захвачен, то поток, пытающийся войти в критическую секцию, блокируется. После того как мутекс освобождается, один из стоящих в очереди потоков (если таковые накопились) разблокируется и получает доступ к глобальным ресурсам.

Средства синхронизации доступа к общим ресурсам достаточно хорошо описаны в [39]. Необходимо отметить, что, кроме POSIX - совместимых операционных систем реального времени, существует большое ни с чем не совместимых ОС, поэтому наличие средств синхронизации и особенности их реализации должны рассматриваться отдельно для каждой конкретной ОС РВ.

Синхронизация с внешними событиями. Известно, что применение аппаратных прерываний является более эффективным методом взаимодействия с внешним миром, чем метод опроса. В системах реального времени в подавляющем большинстве случаев используется механизм аппаратных прерываний. При этом можно проследить следующие тенденции: стремление обеспечить максимально быструю и детерминированную реакцию

системы на внешние события и минимизация возможных периодов времени, когда прерывания в системе запрещены.

Подпрограммы обработки прерываний выполняют необходимый минимальный объем функций за минимально возможным период времени. Это обусловлено несколькими причинами. Во-первых, не все ОС РВ обеспечивают возможность «вытеснения» во время работы подпрограмм обработки прерываний. Во-вторых, приоритеты аппаратных прерываний не всегда соответствуют приоритетам задач, с которыми они связаны. В-третьих, задержки с обработкой прерываний могут привести к потере данных. Как правило, закончив элементарно необходимые действия, подпрограмма обработки прерываний генерирует в той или иной форме сообщение для задачи, с которой это прерывание связано, и немедленно возвращает управление. Если это сообщение перевело задачу в разряд готовых к исполнению, планировщик в зависимости от используемого алгоритма и приоритета задачи принимает решение о том, необходима или нет немедленная передача управления задаче, которая получила сообщение. Это всего лишь один из возможных вариантов, так как каждая ОС РВ имеет свои особенности при обработке прерываний. Кроме того, свою специфику может накладывать используемая аппаратная платформа.

Синхронизация по времени. В настоящее время контроллеры и компьютеры имеют часы с батарейной поддержкой и многофункциональный таймер с разрешением до единиц микросекунд. Как правило, в ОС РВ задается эталонный интервал (квант) времени, который иногда называют «тиком» (Tick) и который используется в качестве базовой единицы измерения времени. Размерность этой единицы для разных ОС РВ может быть разной, как и разными могут быть набор функций и механизмы взаимодействия с таймером. Функции по работе с таймером используют для приостановки выполнения задачи на какое-то время, для запуска задачи в определенное время, для относительной синхронизации нескольких задач по времени и т.п. Множество задач одновременно могут запросить сервис таймера, поэтому если для каждого такого запроса используется элемент в таблице временных интервалов, то накладные расходы системы по обработке прерываний от аппаратного таймера растут пропорционально размерности этой таблицы и могут стать недопустимо большими. Для решения этой проблемы можно вместо таблицы использовать связный список и алгоритм так называемого дифференциального таймера, когда уменьшается только один счетчик интервала времени в зависимости от времени тика.

Для точной синхронизации таймера вычислительной системы с астрономическим временем могут применяться специальные часы с подстройкой по радиосигналам точного времени или навигационные приемники GPS, которые позволяют воспользоваться атомными часами на борту орбитальных космических аппаратов.

Тестирование. Прежде чем устанавливать ОС РВ на реальном объекте, необходимо проверить ее работоспособность при помощи тестов. Во время такого тестирования желательно смоделировать наиболее неприятные и «тяжелые» режимы работы, аварийные ситуации и т.п. Необходимо проверять такую характеристику, как время обработки прерывания, и получить следующие оценки: время от возникновения запроса на прерывание до передачи управления по вектору прерывания; время сохранения контекста текущей задачи и передачи управления подпрограмме обработки прерывания; время до завершения подпрограммы обработки прерывания и передачи сообщения, связанного с прерыванием задачи; время получения задачей управления (в предположении, что она является наиболее приоритетной) до момента начала реальной обработки события.

3.1.3 Операционная система QNX

Операционная система QNX является разработкой канадской компании QNX Software System. Она разработана специально для архитектуры IBM PC (не является многоплатформной), что позволяет достигать высокой эффективности и скорости обработки данных. Система QNX может загружаться из ПЗУ, FLASH-памяти или с помощью удаленной

загрузки по сети. Операционная система QNX представляет собой 16/32-разрядную операционную систему, которую пользователь может конфигурировать по своему усмотрению. Система QNX построена по технологии FLEET - Fault-tolerance (отказоустойчивая), Load-balancing (регулирующая нагрузку), Efficient (эффективная), Extensible (расширяемая), Transparent (прозрачная). Система QNX соответствует требованиям стандарта POSIX. Она ориентирована для приложений реального времени и обеспечивает мультизадачность, приоритетную диспетчеризацию, быстрое переключение контекстов, то есть все необходимые составляющие системы реального времени. Она является гибкой, эффективной, модульной и простой системой за счет двух фундаментальных принципов: архитектуры типа микроядра и связей между процессами, базирующихся на сообщениях.

В табл.3.1 приведены экспериментальные данные, полученные для некоторых типов процессоров (время указано в микросекундах).

Таблица 3.1

Характеристики некоторых типов процессоров

Процессор	Pentium/100	486/33	386/33
Реакция на прерывание	3	8	11
Переключение контекста	3,9	12	26

3.1.3.1. Архитектура

Система QNX [40] состоит из малого ядра (микроядра) управляющего группой взаимодействующих процессов. Размер ядра системы QNX - менее 8 KB. Оно предназначается только для двух существенных функций:

- передачи сообщений - ядро руководит маршрутизацией всех сообщений между всеми процессами во всей системе;
- диспетчеризации - диспетчер является частью ядра и вызывается всякий раз, когда процесс изменяет состояние в результате полученного сообщения или прерывания.

В отличие от процессов ядро никогда само не вызывается диспетчером для выполнения. Вход в него возможен только посредством прямого вызова ядра из процесса или по аппаратному прерыванию.

Системные процессы. Управление всем сервисом QNX, за исключением предоставляемого ядром, производится как управление стандартными процессами системы QNX. Типичная конфигурация системы QNX имеет следующие системные процессы: Менеджер Процессов (Proc); Менеджер Файловой системы (Fsys); Менеджер Устройств (Dev) и Менеджер Сети (Net).

Системные процессы практически ничем не отличаются от программ, написанных пользователем - они не имеют личного или скрытого интерфейса, который недоступен для пользовательских процессов. Такая архитектура обеспечивает высокую гибкость. Так, например, большинство сервиса ОС обеспечивается стандартными процессами системы QNX. Это позволяет пользователю просто наращивать ОС собственными программами, чтобы обеспечить новый сервис. Фактически, граница между операционной системой и приложениями весьма размыта. Единственное существенное различие между системным сервисом и приложениями состоит в том, что ОС управляет ресурсами клиентов.

Драйверы устройств являются процессами, которые избавляют операционную систему от специфики, требуемой для поддержки конкретного оборудования. Поскольку драйверы запускаются как стандартные процессы, добавление нового драйвера в систему QNX не влияет на другие части операционной системы. Единственное изменение, которое необходимо выполнить в окружении системы QNX, это запуск нового драйвера. После завершения инициализации драйверы могут выполнить одно из следующих действий: исчезнуть как стандартный процесс, становясь просто расширением того системного процесса, который он поддерживает; сохранить свою индивидуальность как стандартный процесс.

Сетевые возможности. В простейшей форме локальная сеть обеспечивает механизм для совместного использования файлов и периферийных устройств несколькими соединенными между собой компьютерами. Система QNX идет дальше этих простых понятий и объединяет всю сеть в единое однородное множество ресурсов.

Каждый процесс на любой машине в сети может непосредственно пользоваться тем или иным ресурсом на любой другой машине. Перспективным для приложений является то, что не существует разницы между локальным и удаленным ресурсом и нет необходимости встраивать специальные возможности в приложения для использования удаленных ресурсов. Фактически, программа должна была бы иметь специальную часть, которая определяла бы присутствие некоторого ресурса, как файл или устройство, на локальном компьютере или он находится в другом узле сети.

Пользователи могут иметь доступ к файлам в любом месте сети, захватывать любое периферийное устройство и запускать приложения на любой машине в сети (при условии, что они имеют соответствующие полномочия). Процессы могут связываться друг с другом единообразно по всей сети. Система QNX конструируется из мелких частей как сетевая операционная система.

Сети можно соединять, используя различное оборудование и стандартные промышленные протоколы. Поскольку это полностью прозрачно для прикладных программ и пользователей, то новые сетевые архитектуры можно вводить в любое время без нарушения операционной системы. Каждому узлу сети системы QNX присваивается уникальный номер, который становится его идентификатором. Это число является единственным явно видимым средством для определения, запущена ли QNX как сеть или как однопроцессорная операционная система. Во многих системах важнейшие функции, такие, как работа в сети, IPC или передача сообщений, строятся над ОС, а не объединяются с ее ядром. Результат часто бывает неэффективным. Система QNX, напротив, базируется на эффективной связи.

Ядро системы QNX обеспечивает (рис. 3.1):

- связь между процессами - ядро наблюдает за маршрутизацией сообщений, оно также управляет двумя другими формами IPC - доверенностями и сигналами;
- сетевые связи на нижнем уровне - ядро доставляет все сообщения, предназначенные процессам в других узлах сети;
- диспетчеризацию процессов - диспетчер ядра решает, какой процесс будет выполняться следующим;
- обработку прерываний первого уровня - все аппаратные прерывания и отказы сначала направляются в ядро, а затем передаются соответствующему драйверу или системному менеджеру.

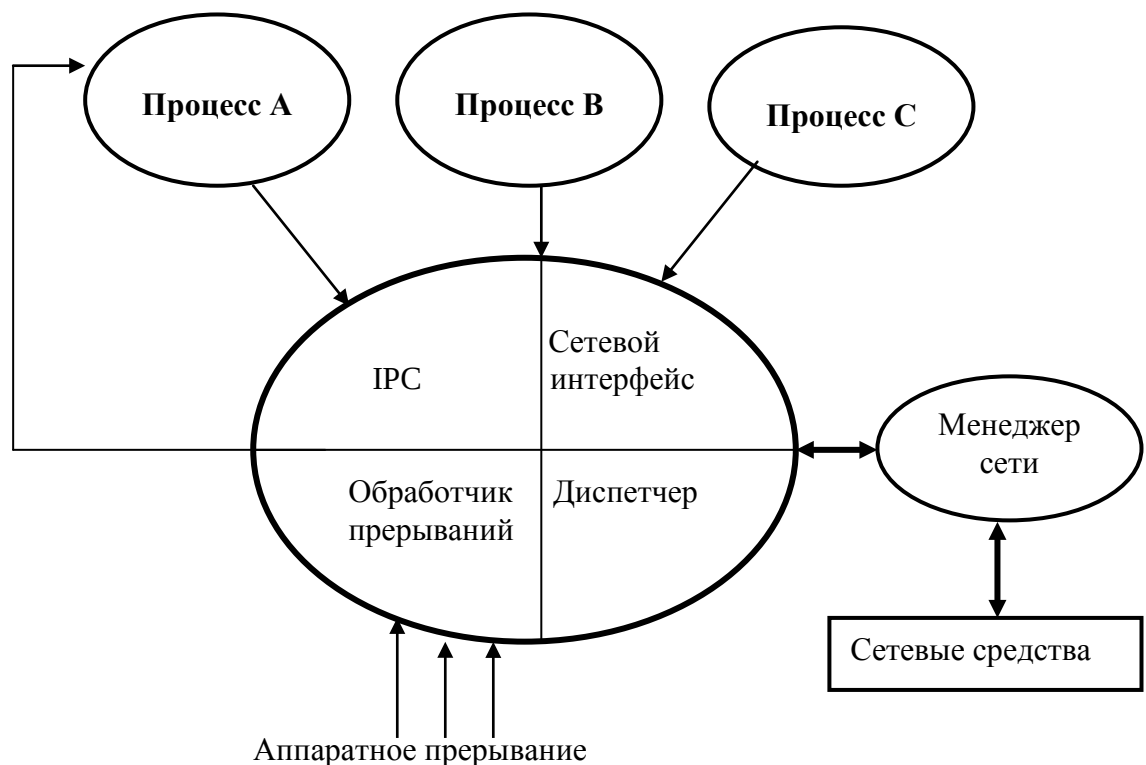


Рис. 3.1. Структура микроядра системы QNX.

Связь между процессами. Если несколько процессов запущены одновременно, как в типичной многозадачной системе реального времени, то операционная система должна обеспечить механизмы связи процессов. Связь между процессами (Interprocess communication - IPC) - это ключевое средство для множества взаимодействующих процессов, в котором каждый процесс обрабатывает четко определенную часть целого.

Сообщение в системе QNX - это пакет байтов, передаваемый от одного процесса к другому. Система QNX не придает никакого специального смысла содержимому сообщения - данные в сообщении имеют смысл для отправителя этого сообщения и его получателя, но ни для кого больше. Передача сообщений не только позволяет процессам передать данные друг другу, но также обеспечивает смысловую синхронизацию выполнения нескольких процессов. Посылая сообщения, принимая их и отвечая на них, процессы подвергаются различным "изменениям состояния", которые влияют на то, когда и как долго они могут быть запущены. Ядро системы QNX поддерживает три основных типа связи между процессами: сообщения, доверенности и сигналы. Сообщения - это основная форма IPC в системе QNX. Они обеспечивают синхронную связь между взаимодействующими процессами, где процесс, посылающий сообщение, требует подтверждения приема и, возможно, ответа на это сообщение. Доверенности (проху) - это специальная форма сообщений. Они особенно подходят для отметки событий, когда посылающий процесс не требует взаимодействия с получателем. Сигналы - это традиционная форма IPC. Они используются для поддержки асинхронной связи между процессами.

Чтобы связываться непосредственно друг с другом, взаимодействующие процессы пользуются функциями языка Си: Send() - для отправления сообщений; Receive() - для приема сообщений; Reply() - для передачи ответа процессам, которые прислали сообщения. Эти функции могут быть использованы локально или через сеть. Библиотека Си в QNX компонуется с модулем обмена сообщениями - процессы косвенно используют обмен сообщениями, когда пользуются стандартным сервисом таким, как канал. Передача сообщений не только позволяет процессам передавать данные друг другу, но и обеспечивает средства синхронизации выполнения нескольких взаимодействующих процессов. Если процесс А

выполнил запрос `Send()`, ему не разрешается возобновлять выполнение до тех пор, пока он не получит ответ на отправленное им сообщение. Этим гарантируется, что обработка, выполняемая процессом *B* для процесса *A*, завершится до того, как процесс *A* сможет продолжить выполнение. Более того, раз процесс *B* выполняет свой запрос `Receive()`, он не может продолжать обработку до тех пор, пока не получит другое сообщение. Если процессу не разрешается продолжать выполнение - поскольку он должен ожидать завершения некоторой части протокола обмена сообщениями - то процесс является блокированным.

Процесс-сервер в обычном состоянии является `RECEIVE`-блокированным, ожидая запроса от клиента, чтобы выполнить некоторое задание. Этот механизм называется "обмен сообщениями, управляемый посылкой". Процесс-клиент инициирует действие посылкой сообщения, а завершение этого действия происходит, когда сервер ответит на это сообщение. Возможно также использование и другой формы обмена сообщениями: обмен сообщениями, управляемый ответом, при котором действие инициируется посредством `Reply()`. По этому методу "рабочий" процесс посылает серверу сообщение, указывающее, что он готов к работе. Сервер не отвечает сразу же, а "запоминает", что рабочий процесс прислал управляющее сообщение. Через некоторое время сервер может решить, что пора начать некое действие, и пошлет ответ рабочему процессу. Рабочий процесс выполнит это действие и завершит его посылкой серверу сообщения с результатами.

Система QNX обеспечивает также следующие расширенные возможности для обмена сообщениями: прием сообщений по условию; чтение или запись части сообщения; сообщения, состоящие из нескольких частей.

Доверенность - это форма неблокированного сообщения, специально приспособленного для извещения о событии, когда посылающий процесс не нуждается в ответе получателя. Единственной функцией доверенности является отправление фиксированного сообщения указанному процессу, который понимает эти доверенности. Подобно сообщениям, доверенности работают по всей сети. Используя доверенности, процесс или обработчик прерывания может послать сообщение другому процессу без блокирования и без необходимого ожидания ответа. Примеры использования доверенности: процесс хочет известить другой процесс о том, что некоторое событие произошло, но не может позволить себе становиться `SEND`-блокированным, пока получатель вызывает `Receive()` и `Reply()`; процесс хочет переслать данные другому процессу, но не нуждается ни в ответе, ни в каком-либо другом извещении о том, что получатель принял это сообщение; обработчик прерываний хочет сообщить процессу, что некоторые данные стали доступны для обработки.

Доверенности создаются Си-функцией `qnx_proxu_attach()`. Любые другие процессы или обработчики прерываний, которым известна идентификация доверенностей, могут использовать доверенность для доставки своих сообщений при помощи Си-функции `Trigger()`. Ядро обрабатывает запрос `Trigger()`. Доверенность может быть задействована более одного раза - она посылает сообщение по каждому включению. Обработчик доверенностей может очереждать до 65535 сообщений для доставки. Сигналы являются традиционным способом асинхронной связи, который доступен в течение многих лет в различных операционных системах. Система QNX поддерживает множество сигналов, согласующихся с `POSIX`, вдобавок к специфичным сигналам QNX.

Сигнал предназначается для доставки процессу, когда выполнилось действие, предусмотренное процессом по этому сигналу. Процесс может сам установить сигнал. Он может принять сигнал одним из трех способов, в зависимости от того, как он определяет свое окружение, обрабатывающее сигналы: 1 - если процесс не имеет специальных действий по обработке сигнала, то по этому сигналу выполняются подразумеваемые действия. Обычно таким действием является прекращение процесса; 2 - процесс может игнорировать сигнал. Если процесс игнорирует сигнал, то доставка сигнала не влияет на процесс; 3 - если процесс содержит обработчик для некоторого сигнала, то говорят о способности "захватить" этот сигнал. Любой процесс, захватывающий сигнал, является в действительности формой программного прерывания. С сигналом не передаются никакие данные. В промежутке между

временем, когда сигнал был выработан, и временем, когда он был доставлен, сигнал является подвешенным. В данный момент времени для некоторого процесса может быть подвешено несколько вполне определенных сигналов. Сигналы доставляются процессу, когда процесс готов к запуску диспетчером.

Между сигналами и сообщениями существует взаимодействие. Если процесс является SEND-блокированным или RECEIVE-блокированным при генерации сигнала - и имеется обработчик сигналов - то выполняются следующие действия: процесс разблокируется; выполняется обработка сигнала; выполняется возврат из Send() или Receive() с ошибкой. Если процесс был в это время SEND-блокированным, то получатель не получит сообщения. Но если бы процесс был REPLY-блокированным, неизвестно было ли обработано посланное сообщение, или нет. Процесс может действовать как сервер (т.е. принимать сообщения). В таком случае процесс-клиент станет SIGNAL-блокированным с подвешенным сигналом, и процесс-сервер получит специальное сообщение, описывающее тип сигнала. Затем процесс-сервер может выбрать одно из следующих действий:

- нормально завершить исходный запрос - отправитель уверен, что сообщение обработалось правильно;
- высвободить связанные ресурсы и вернуть ошибку, указывающую, что процесс был разблокирован по сигналу - отправитель получит точную индикацию ошибки.

Если сервер отвечает процессу, который был SIGNAL-блокированным, то сигнал будет действовать непосредственно после возврата из Send() отправителя.

Взаимодействие через сеть. Приложение системы QNX может взаимодействовать с процессом на другом компьютере в сети точно так же, как с другим процессом на своей машине. В сущности, нет разницы между локальными и удаленными ресурсами. Такой уровень прозрачности стал возможным благодаря виртуальным каналам связи (VC), являющимся путями, которыми обеспечивает Менеджер Сети для передачи сообщений, доверенностей и сигналов по всей сети. Канал связи VC способствуют эффективному повсеместному использованию ресурсов в сети QNX по нескольким причинам. Когда создается канал связи VC, он дает возможность обрабатывать сообщения до указанного размера; а это значит, что можно заранее выделить ресурсы для обработки сообщения. Несмотря на это, если необходимо послать сообщение большего размера, чем максимально указанный, VC автоматически изменит размер, чтобы приспособиться к большим сообщениям. Если два процесса, размещенные в разных узлах, связываются друг с другом более чем по одному каналу связи VC, то последние являются совместно используемыми. В действительности существует только один виртуальный канал. Такая ситуация возникает в общем случае, когда процесс имеет доступ к нескольким файлам удаленной файловой системы. Если процесс подсоединяется к существующему совместно используемому VC и требует буфер, больший, чем текущий используемый, то размер буфера автоматически увеличивается. Когда процесс завершается, соответствовавший ему VC автоматически освобождается. Посылающий процесс является ответственным за установку VC между собой и тем процессом, с которым он хочет связаться. Чтобы сделать это, посылающий процесс обычно выполняет вызов функции `qnx_vc_attach()`. Вдобавок к созданию VC этот вызов создает еще и виртуальный идентификатор процесса, или VID, на обоих концах канала. Что касается процессов на обоих концах виртуального канала, то VID на соответствующем конце появляется для того, чтобы получить возможность доступа к идентификатору удаленного процесса (PID), с которым процесс хочет связаться. Процессы связываются друг с другом через эти VID. Каждый VID поддерживает связь, которая несет следующую информацию: локальный pid; удаленный pid; удаленный nid (ID узла); удаленный vid.

Приложение не принимает непосредственного участия в создании или использовании канала связи VC. Если приложение устанавливает положение некоего сервера с помощью Си-функции `qnx_name_locate()`, автоматически создается VC от имени этого приложения. По отношению к приложению VC появляется просто для того, чтобы служить PID.

3.1.3.2. Диспетчеризация процессов

Диспетчер Ядра управляет процессом, если: 1 - процесс становится неблокированным; 2 - истекает квант времени, выделенный процессу; 3 - запущенный процесс имеет привилегии. В QNX каждому процессу назначается приоритет. Диспетчер выбирает очередной процесс для запуска, просматривая приоритеты всех процессов, которые находятся в состоянии готовности (READY). READY-процесс - это такой процесс, который способен использовать процессор. Для запуска выбирается процесс с наивысшим приоритетом. QNX обеспечивает три метода диспетчеризации: FIFO-диспетчеризацию; «карусель» и - динамическую диспетчеризацию. Каждый процесс в системе может быть запущен с помощью любого из этих методов. Если становится готовым процесс с наивысшим приоритетом, он немедленно получает преимущество перед всеми процессами с более низкими приоритетами. При FIFO-диспетчеризации процесс, выбранный для запуска, продолжает выполнение до тех пор, пока добровольно не отдаст управление (т.е. блокируется), не появится процесс с более высоким приоритетом. Два процесса, запущенные с одинаковыми приоритетами, могут пользоваться FIFO-диспетчеризацией. При диспетчеризации методом «карусель» процесс, выбранный для запуска, продолжает выполняться до тех пор, пока добровольно не отдаст управление, не появится процесс с более высоким приоритетом и не истечет его квант времени. После истечения кванта времени, управление передается следующему готовому процессу. Квант времени составляет 100 ms. При динамической диспетчеризации процессы ведут себя следующим образом. Если процесс исчерпал свой квант времени (т.е. он не блокирован), то его приоритет уменьшается на 1, если другой процесс с таким же приоритетом готов. Это называется "разложением приоритета". Если процесс "разложился" и остается недиспетчеризованным в течение одной секунды, его приоритет повышается на 1 (приоритет процесса никогда не станет выше первоначального). Если процесс блокируется, он немедленно возвратит свой первоначальный приоритет. Динамическая диспетчеризация дает процессам с интенсивными вычислениями достаточный доступ к процессору.

В QNX большинство взаимодействий между процессами происходит по модели клиент-сервер. Серверы обеспечивают некоторую форму сервиса, и клиенты посылают сообщения этим серверам, чтобы запросить какой-либо сервис. В общем, серверы являются более надежными и жизнеспособными, чем клиенты. Клиентов обычно больше, чем серверов. В результате, общепринято запускать сервер с приоритетом, превышающим приоритеты всех его клиентов. Метод диспетчеризации может быть одним из трех вышеописанных, но "карусель" будет, вероятно, наиболее общим.

Работа в реальном времени. В системах реального времени важно, чтобы процессорное время использовалось эффективно. Важна минимизация времени от возникновения некоторого внешнего события до передачи управления программе, ответственной за реакцию на это событие. Состояние ОС в этот период называют скрытым состоянием. В системе QNX есть несколько форм скрытых состояний. Скрытое состояние прерывания - это время от получения аппаратного прерывания до выполнения первой команды программного обработчика прерываний. QNX оставляет прерывания полностью открытыми почти все время, так что скрытое состояние прерывания обычно незначительно. Но отдельные критические участки требуют, чтобы прерывания временно были запрещены. Максимальное время такого запрета обычно определяет наихудший случай скрытого состояния прерывания - в QNX оно очень мало. В некоторых случаях низкоуровневый обработчик аппаратного прерывания должен инициировать запуск высокоуровневого процесса. В таком положении обработчик прерываний выполняет возврат с индикацией посылки доверенности. Это вторая форма скрытого состояния - скрытое состояние диспетчеризации. Скрытое состояние диспетчеризации - это время между завершением обработчика прерываний и выполнением первой команды драйвера-процесса. Обычно это время, которое используется для запоминания контекста текущего выполняемого процесса и восстановление контекста требуемого драйвера-процесса. Это время в системе QNX также мало, хотя и больше, чем скрытое состояние прерывания.

3.1.3.3. Менеджер процессов

Менеджер процессов работает в тесном контакте с ядром. Хотя он и пользуется тем же адресным пространством, что и ядро, и является единственным процессом, делающим это, Менеджер Процессов запускается, как истинный процесс. А раз так, то он диспетчируется ядром подобно всем другим процессам. Менеджер Процессов отвечает за создание новых процессов в системе и за распределение наиболее фундаментальных ресурсов, соответствующих процессу. Этот сервис полностью обеспечивается через сообщения. Например, если запущенный процесс хочет создать новый, он делает это, посылая сообщение, содержащее подробности о новом процессе, который будет создан. QNX поддерживает такие примитивы создания процесса, как: `fork()`, `exec()`, `spawn()`. `Fork()` и `spawn()` определяются стандартом POSIX, а применение `spawn()` уникально для QNX. Примитив `fork()` создает новый процесс, который является точной копией образа вызывающего процесса. Новый процесс выполняет те же коды, что и вызывающий, и наследует копию данных вызывающего процесса. Примитив `exec()` замещает образ вызывающего процесса образом нового процесса. Из успешно выполнившегося `exec()` нет возврата, т.к. образ вызывающего процесса перекроется образом нового процесса. В POSIX-системах практикуется создание нового процесса - без уничтожения вызывающего процесса - вызывая вначале `fork()`, а затем, получая процесс-потомок от `fork()` вызовом `exec()`. Примитив `spawn()` создает новый процесс, потомок вызывающего процесса. Он может избежать необходимости в `fork()` и `exec()`, давая в результате более быстрое и эффективное создание нового процесса.

В отличие от `fork()` и `exec()`, которые в силу своей сущности действуют в том же узле, что и вызывающий процесс, примитив `spawn()` может создать процессы на любом узле в сети. Любой процесс проходит через четыре фазы: 1 - создание; 2 - загрузка; 3 - выполнение; 4 - завершение. Создание процесса включает выделение идентификатора для нового процесса и уточнения информации, определяющей окружение нового процесса. Большая часть этой информации наследуется от родителей нового процесса. Загрузка образов процесса выполняется цепочкой (нитью) загрузчика. Загрузчик находится в Менеджере Процессов, а цепочка запускается по идентификатору нового процесса. Это позволяет Менеджеру Процессов обрабатывать другие запросы во время загрузки программ. Как только программа загрузится, процесс становится готовым к выполнению; он начинает конкурировать с другими процессами. Заметим, что все процессы запущены одновременно со своими родителями. Гибель родительского процесса не вызывает автоматической гибели его процесса-потомка. Процесс завершается одним из двух способов: процессу доставляется сигнал, действием которого является завершение процесса; процесс вызывает Си-функцию `exit()` либо непосредственно, либо неявным действием возврата из `main()`. Любой процесс всегда находится в одном из следующих состояний:

- READY (готов) - данный процесс способен использовать процессор;
- BLOCKED (блокирован) - процесс находится в одном из блокированных состояний;
- HELD (захвачен) - данный процесс получил сигнал SIGSTOP, пока он не выйдет из HELD-состояния, ему не предоставляется процессор;
- WAIT-блокированный - процесс вызвал `wait()` или `waitpid()`, чтобы ожидать состояние от одного или более его процессов потомков.
- DEAD(умер) - данный процесс завершился, но не может послать свое выходное состояние процессу-родителю, т.к. процесс-родитель не вызвал `wait()` или `waitpid()`.

QNX способствует созданию приложений, которые расщепляются на совместно выполняющиеся процессы. Процессы идентифицируются локальными (в узле) и глобальными (в сети) именами. Чтобы можно было пользоваться глобальными именами, по крайней мере, в одном узле сети должен быть запущен процесс, называемый локатором имен процессов (т.е. утилита `nameloc`). Этот процесс сохраняет запись всех зарегистрированных глобальных имен.

Таймеры. В QNX управление временем основано на системном таймере, поддерживаемом операционной системой. Таймер содержит текущее Скоординированное Универсальное Время (UTC) относительно 0 часов, 0 минут, 0 секунд, 1 января 1970 г. Чтобы установить локальное время, функции управления временем используют переменную окружения TZ (которая описана в Руководстве Пользователя). Сценарии Shell и процессы могут делать паузы на указанное число секунд, пользуясь простыми возможностями работы со временем. Для Shell эти возможности предоставляются утилитой sleep; для процессов - Си-функцией sleep(). Процесс также может создавать таймеры, задавая для них интервал времени, и удалять таймеры. Такие расширенные возможности работы со временем основываются на спецификации POSIX Std 1003.

Обработчики прерываний. Обработчики прерываний обслуживают систему аппаратных прерываний компьютера. Обработчики прерываний физически оформлены как часть стандартного процесса QNX (например, драйвера), но они всегда запускаются асинхронно с процессом, с которым они связаны. Обработчик прерывания вызывается far-вызовом, а не непосредственно по прерыванию; запускается в контексте с процессом, в который он встроен, и потому имеет доступ ко всем глобальным переменным этого процесса; запускается при открытых прерываниях, но уступает инициативу только в случае, если возникает прерывание более высокого приоритета; не может общаться непосредственно с контроллером прерываний.

К одному и тому же прерыванию можно прикрепить несколько процессов (если это поддерживается аппаратно). Когда возникает физическое прерывание, каждый обработчик прерывания по очереди получит управление. Можно привязать обработчик прерываний непосредственно к системному таймеру так, что этот обработчик будет вызываться по каждому прерыванию таймера. Чтобы установить период, можно воспользоваться утилитой ticksize. Можно также привязаться к прерыванию масштабированного таймера, которое активируется каждые 100 ms, не обращая внимания на размер тика. Такие таймеры являются альтернативой таймерам POSIX 1003.4.

Пространство имен ввода/вывода. Ресурсы ввода/вывода в QNX не являются встроенными в ядро. Вместо этого сервис ввода/вывода обеспечивается процессами, которые могут быть запущены динамически, пока запущена система. Поскольку файловая система в QNX является не обязательной, то в эту систему встроено пространство имен путей, как это сделано в большинстве монолитных систем. В QNX пространство имен путей делится на области полномочий. Любой процесс, который желает обеспечить файл-ориентированный сервис ввода/вывода, будет регистрировать префикс с указанием Менеджеру процессов части пространства имен, которой он желает управлять. Эти префиксы составляют дерево префиксов, которое сохраняется в памяти каждого компьютера, запускающего QNX. Когда процесс открывает файл, к дереву префиксов прилагается имя пути файла, чтобы указать open() путь к соответствующему менеджеру ресурсов. Например, Менеджер символьного устройства (Dev) обычно регистрирует префикс /dev. Если процесс вызывает open() с /dev/xxx, появляется префикс, соответствующий /dev, и open() будет направлен к Dev (его владельцу). QNX поддерживает идею суперкорня или корня сети, которая позволяет применять имя пути к дереву префиксов специфического узла, указываемого именем пути, начинающимся с двух косых, за которыми следует номер узла. Это также позволяет легко получить доступ к файлам и устройствам, которые не находятся в обычном пространстве имен путей вашего узла. Например, в типичной сети QNX следующие пути были бы проложены так:

- /dev/ser1 - локальный последовательный порт;
- //10/dev/ser1 - последовательный порт на узле 10;
- //0/dev/ser1 - локальный последовательный порт;
- //20/usr/dtdodge/test - файл на узле 20.

Если программа выполняется отдельно, желательно, чтобы все имеющиеся имена путей были решены в контексте пространства имен путей его собственного узла.

3.1.3.4. Менеджер файловой системы

Менеджер Файловой системы (Fsys) обеспечивает стандартные средства размещения и доступа к данным на дисковых подсистемах. Fsys отвечает за обработку всех запросов на открытие, закрытие, чтение и запись файлов. В QNX файл - это объект, в который можно записывать, из которого можно читать или то и другое вместе. Различают шесть типов файлов, пять из них управляются Fsys:

- регулярные файлы - состоят из последовательности байтов с произвольным доступом к ним и не имеют предопределенной структуры;
- каталоги - содержат информацию, необходимую для отыскания регулярных файлов; они также содержат состояние и информацию об атрибутах для каждого регулярного файла;
- символические связи - содержат имя пути к файлу или каталогу, к которому будет получен доступ вместо этого файла символической связи;
- каналы и FIFO - служат каналами ввода/вывода между взаимодействующими процессами;
- файлы специальных блоков - относятся к устройствам, таким, как дисковые накопители, ленты, и разделы дисковых накопителей.

Шестой тип, специальный файл символов, управляется Менеджером Устройств.

Доступ к файлу. Доступ к регулярным файлам и каталогам управляется битами режима, помещенными в информационный узел (inode) файла. Эти биты разрешают чтение, запись и выполнение, основываясь на действительных идентификаторах пользователей и групп. Существует три группы доступа: только пользователь, только группа и другие.

Процесс может запускаться с ID пользователя или с ID группы файла, а не с идентификатором его родительского процесса. К механизму, разрешающему это, ссылаются как к `setuid` (установить ID пользователя на выполнение) и `setgid` (установить ID группы на выполнение).

Регулярные файлы и каталоги. QNX рассматривает регулярный файл как произвольно доступную последовательность байтов, которая не имеет никакой предопределенной внутренней структуры. За понимание структуры и содержимого каждого отдельного регулярного файла ответственны прикладные программы. Регулярные файлы составляют большинство файлов, находящихся в файловых системах. Файловые системы поддерживаются Менеджером Файловой системы и выполняются самыми первыми в файлах специальных блоков, которые определяют разделы диска. Хотя каталог ведет себя почти, как стандартный файл, Менеджер Файловой системы налагает некоторые ограничения на операции, которые можно выполнять в каталоге. А именно, нельзя открыть каталог на запись, а также связаться с каталогом с помощью Си-функции `link()`. Чтобы прочитать элемент каталога, необходимо пользоваться множеством определенных POSIX Си-функций, которые обеспечивают доступ к элементам каталога независимо от ОС. Это следующие функции: `opendir()`, `readdir()`, `rewinddir()` и `closedir()`.

Поскольку каталоги QNX являются просто файлами, которые содержат "известную" информацию, можно также читать элементы каталога, пользуясь непосредственно Си-функциями `open()` и `read()`. Такая техника не является переносимой, однако и формат элементов каталога меняется с операционной системой.

Связи и информационные узлы. В QNX на файл данных можно ссылаться более чем по одному имени. Каждое имя файла называется связью. Для того чтобы поддержать связи для каждого файла, имена файлов отделяются от всей другой информации, описывающей файл. Эта информация размещается в структуре, называемой inode - информационным узлом. Если файл имеет только одну связь (т.е. одно имя файла), то его inode-информация размещается в элементе каталога для этого файла. Если файл имеет более чем одну связь, inode размещается в виде записи в специальном файле, называемом `/.inodes`, а элемент каталога для данного файла.

Канал - это безымянный файл, который служит каналом ввода/вывода между двумя или более взаимосвязанными процессами - один процесс пишет в канал, другой читает из канала.

Менеджер Файловой системы заботится о буферировании данных. Размер буфера определяется как {PIPE_BUF} в файле <limits.h>. Канал удаляется, если оба его конца закрылись. Каналы обычно используются, если два процесса хотят запуститься параллельно с передачей данных от одного процесса другому в одном направлении. Если требуется двунаправленная связь, нужно пользоваться сообщениями. Типичным применением для канала является привязка вывода одной программы к вводу другой.

Выполнение менеджера файловой системы. Менеджер Файловой системы имеет несколько возможностей, которые способствуют доступу к высокопроизводительному диску: установка элеватора; кэш-буфер; мультидоступ (multi-threading); управляемый клиентом приоритет; промежуточные файлы; псевдодиски. Установка элеватора минимизирует общее время установки, требующееся для записи данных на диск. Невыполненные запросы на запись упорядочиваются так, чтобы все они могли быть записаны с одного подвода дисковой головки, от меньших дисковых адресов к большим. Кэш-буфер - это высокоорганизованный буфер между Менеджером Файловой системы и драйвером диска. Кэш-буфер пытается разместить блоки файловой системы в таком порядке, чтобы минимизировать число обращений Менеджера Файловой системы к диску. По умолчанию, размер кэша определяется всей системной памятью, но можно указать другой размер через опции Fsys.

Операции чтения синхронизированы. Операции записи, напротив, обычно не синхронизированы. Когда данные попадают в кэш, Менеджер Файловой системы отвечает процессу-клиенту, чтобы отметить, что данные записаны. Затем данные при первой же возможности переписываются на диск, обычно это происходит менее чем через две секунды. Программное обеспечение кэша обычно отдает предпочтение чтению перед записью.

3.1.3.5. Менеджер устройств

Менеджер Устройств QNX (Dev) является интерфейсом между процессами и внешними устройствами. Эти устройства отыскиваются в пространстве имен ввода/вывода под именами, начинающимися на /dev. Программы QNX получают доступ к внешним устройствам, пользуясь стандартными функциями read(), write(), open(), close(). Внешние устройства предоставляют QNX-процессу двунаправленный поток байтов, который может быть прочитан или записан процессом. Менеджер Устройств регулирует поток данных между приложением и устройством. Dev выполняет некоторую обработку этих данных в соответствии с параметрами в управляющей структуре терминала (называемой termios), которая существует для каждого устройства. Пользователи могут отыскивать и изменять эти параметры, используя утилиту stty; программы могут использовать функции tcgetattr() и tcsetattr(). Параметры termios управляют следующими функциями на нижнем уровне:

- дисциплина контроля линии (включая скорость передачи, паритет, стоп-биты и биты данных);

- "эхо" символов;
- редактирование входной строки;
- распознавание и выполнение "break" и подвешивания;
- управление программным и аппаратным потоками;
- трансляция выходных символов.

Менеджер Устройств обеспечивает также множество дополнительного сервиса, доступного процессам для управления внешними устройствами.

Режим редактируемого ввода. Наиболее важный режим обработки в устройстве управляется битом ICANON в управляющей структуре termios. Если этот бит установлен, то Менеджер Устройств выполняет функции построчного редактирования принятых символов. Таким образом, обрабатываемые данные становятся доступными для прикладных процессов, только тогда, когда строка "введена" - обычно, если принят код возврата каретки (CR). Этот режим оперирования обычно называют редактируемым, каноническим, а иногда режимом приготовления.

Большинство приложений, не занимающих полный экран, запускаются в режиме редактирования. Типичным примером является Shell.

Режим неотредактированного ввода. Если ICANON не установлен, то говорят, что устройство находится в необрабатывающем режиме, не выполняется никакого редактирования при вводе, и любое принятое данное немедленно становится доступным процессам QNX. Программы с полным экраном и программы последовательной связи являются примерами приложений QNX, которые переводят устройства в необрабатывающий режим. При чтении с необрабатывающего устройства приложение способно указывать, при каких условиях запрос на ввод будет удовлетворен. Критерии, используемые для поддержки ввода необрабатываемых данных, основываются на двух членах управляющей структуры `termios`: `MIN` и `TIME`. Приложение может указать квалификатор поддержки ввода данных, когда оно выдает запрос на чтение посредством `dev_read()`.

Драйверы устройств. На рис. 3.2 приведена типичная для QNX структура связи драйверов с процессами. Менеджер Устройств (`Dev`) управляет потоком данных к прикладным процессам QNX и от них. Аппаратным интерфейсом управляют индивидуальные процессы - драйверы. Данные передаются между `Dev` и его драйверами сквозь множество очередей совместно используемой памяти для каждого внешнего устройства. Для каждого устройства используются три очереди. Каждая очередь применяется с использованием механизма "первый пришел, первый ушел". Каждой очереди соответствует управляющая структура. Принятые данные помещаются драйвером в необрабатываемую входную очередь и передаются Менеджером только тогда, когда прикладной процесс запрашивает данные. Обработчики прерываний внутри драйверов обычно вызывают библиотечные программы из Менеджера, чтобы добавлять данные в эту очередь - это гарантирует дисциплину последовательного ввода. Менеджер помещает выводимые данные в выходную очередь. Менеджер вызывает из процесса-драйвера соответствующие программы каждый раз, когда добавляются новые данные. Поскольку используются выходные очереди, Менеджер выполняет полностью фоновую запись для всех внешних устройств. Только в случаях, если выходные буферы заполнены, Менеджер заблокирует процесс на время записи. Каноническая очередь управляется полностью Менеджером и используется при обработке данных в режиме редактирования. Размеры всех очередей конфигурируются администратором системы; единственным ограничением является то, что суммарная длина всех трех очередей не должна превышать 64 К.

Управление устройством. Драйверы устройств просто добавляют принимаемые данные во входную очередь необработанных данных или передают данные из выходной очереди. Менеджер решает, когда приостановить передачу данных, как отображать принимаемые данные, и т.п. Чтобы гарантировать хорошую интерактивную реакцию на события ввода, Менеджер должен быть запущен с разумно высоким приоритетом. Менеджер, как правило, имеет очень мало действительно выполняемой работы после запуска, поэтому он редко мешает общей работе системы.

Управление устройством на нижнем уровне выполняется посредством дальнего перехода на входную точку `ioctl` в каждом драйвере. Общее множество команд `ioctl` поддерживается большинством драйверов, непосредственно используемых Менеджером. Специфичные для устройства команды `ioctl` могут быть также посланы драйверам от процессов QNX через Менеджер (посредством Си-функции `qnx_ioctl()`).

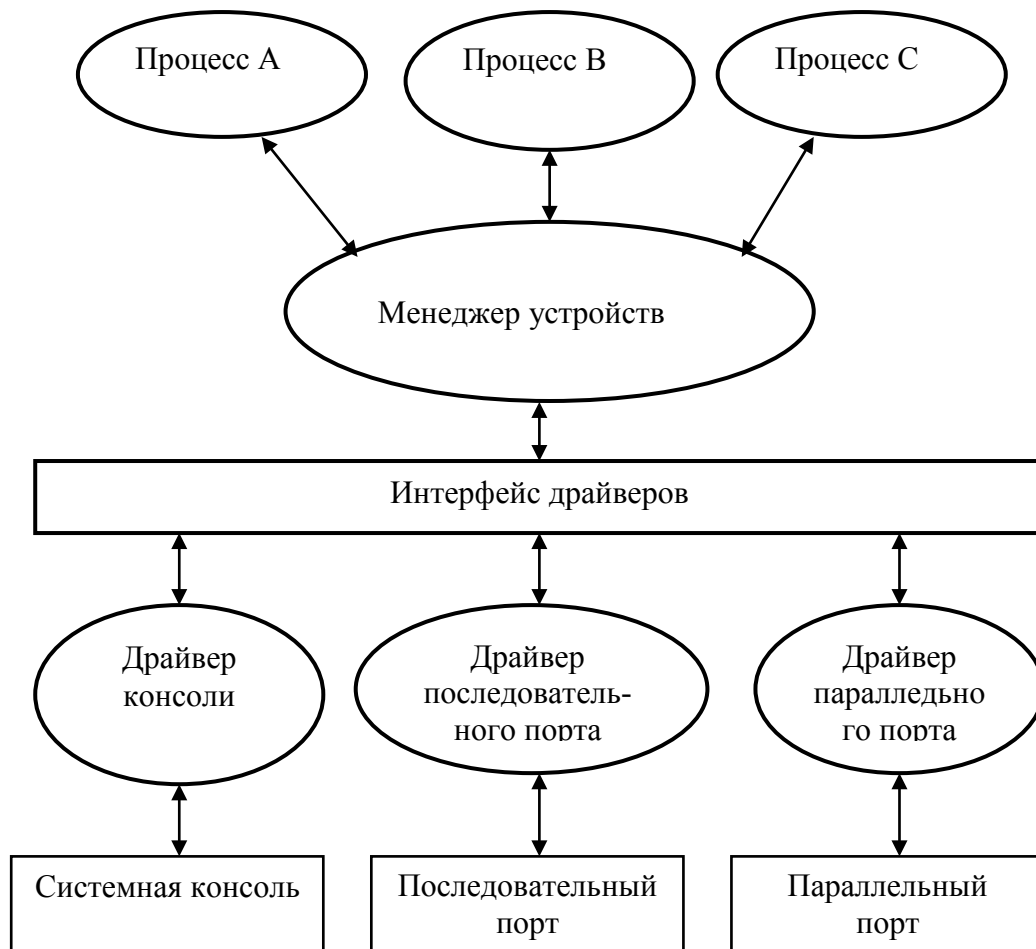


Рис. 3.2. Структура связи драйверов с процессами.

3.1.3.6. Менеджер сети

Менеджер Сети (Net) дает пользователям QNX расширение возможности операционной системы - обмена сообщениями. Связывающийся непосредственно с ядром, Менеджер Сети усиливает связь между процессами в QNX, основанную на передаче сообщений, путем распространения сообщений на удаленные машины. Дополнительно к этому, Менеджер Сети предлагает две дополнительные возможности: увеличивает пропускную способность через балансировку нагрузки и устраняет отказы через дополнительную связь.

Функции менеджера сети. Менеджер Сети управляет обменом сообщениями QNX через локальную область сети. Стандартные примитивы обмена сообщениями, используемые при локальном обмене, без изменений используются при удаленном обмене. Другими словами, не существует специальных "сетевых" Send(), Receive() или Reply(). Менеджер Сети не встроен в операционную систему. Он может запускаться и завершаться в любой момент времени, чтобы обеспечить возможность обмена сообщениями по сети. Когда Менеджер Сети запускается, он регистрируется Менеджером Процессов и ядром. Обмен сообщениями по сети объединяется в самой сути обмена сообщениями и самих примитивов управления процессом обмена. Такая интеграция на самом низком уровне дает QNX сетевую прозрачность и квалифицирует ее как полностью распределенную операционную систему.

Интерфейс микроядро - менеджер сети. Ядро и Менеджер Процессов взаимодействуют с Менеджером Сети через специальную неблокируемую очередь в памяти. Эта очередь является по существу списком передач, которые должны быть выполнены Менеджером Сети. Элементы содержат всю информацию для отдельных операций (например, Send(), Reply(), создание канала связи VC, распространение удаленного сигнала). Другим ресурсом, который использует операционная система для обеспечения обмена сообщениями, является буфер виртуального

канала связи. Буфер виртуального канала связи, выделяемый, когда процесс создает VC, содержит данные, пока полностью не завершится обмен сообщениями с другим узлом. Любые управляющие пакеты, которые получает Менеджер Сети, он немедленно препровождает Менеджеру Процессов через стандартный примитив Send(). Эти управляющие пакеты используются для распространения сигналов и создания VC.

Драйверы сети. Подобно Менеджеру Файловой системы и Менеджеру устройств, Менеджер Сети не содержит кодов, специфичных для аппаратуры. Это функционирование обеспечивается драйверами сетевой карты. Очереди в совместно используемой памяти обеспечивают интерфейс между Менеджером Сети и драйверами. Этот интерфейс разработан для получения максимально возможной производительности. Драйвер определяет подходящий протокол для сетевого носителя. Он отвечает за формирование пакетов, упорядочивание очередности и повторную передачу, если требуются гарантии передачи данных к удаленному физическому узлу. Это подразумевается для всех примитивов обмена сообщениями QNX. Такое решение позволяет QNX поддерживать новое сетевое оборудование и протоколы посредством записи или модификации только сетевых драйверов.

Идентификаторы узла сети. Каждый узел в локальной области сети идентифицируется двумя числами - его физическим и логическим идентификаторами узла. Физический идентификатор узла определяется аппаратурой. Сетевые карты связываются друг с другом путем указания физического идентификатора узла для того узла, с которым они хотят связаться. Каждому узлу QNX дается логический номер. Все процессы QNX имеют дело с логическими идентификаторами узлов. Физические идентификаторы скрыты от процессов, запускаемых в QNX.

Логические идентификаторы узлов упрощают привилегии сети и приложений. Соответствие между физическими и логическими идентификаторами узлов устанавливается в Менеджере Сети. Когда Менеджер Сети выдает запрос на передачу данных к другому узлу, он дает драйверу физический идентификатор узла. В качестве логических идентификаторов узлов обычно назначают последовательные целые числа, начиная с 1. Логический идентификатор узла должен быть уникален для всех узлов, подключаемых к сети QNX. Идентификатор сети идентифицирует отдельную логическую сеть. Логическая сеть - это некоторая аппаратура, позволяющая драйверу сети непосредственно связываться с драйвером сети на другом узле. Пропускная способность сети определяется комбинацией скорости компьютера и скорости сети. Если компьютер может обеспечивать данные быстрее, чем сеть в состоянии поддержать их, то сеть будет ограничивать пропускную способность. Менеджер Сети будет пытаться балансировать нагрузку, выбирая подходящий драйвер сети. Если узлы соединяются двумя или более сетями, существует более чем один путь установления связи. Если карта в одной сети отказала так, что прервала всякую связь по этой сети, то Менеджер Сети будет автоматически повторно передавать данные по другой сети. Это происходит "на лету" без какого-либо привлечения программных приложений, и в результате в прозрачной сети отказ устраняется. К достоинствам системы QNX можно отнести следующие:

- система работает в защищенном режиме процессора, и таким образом обеспечивает высокую надежность, защищенность задач пользователя друг от друга, а также операционной системы от некорректного поведения пользовательских задач;
- имеется встроенная поддержка (в виде менеджеров) нескольких сетевых протоколов;
- данная система является POSIX совместимой, что обеспечивает высокую мобильность системы и переносимость программного обеспечения;
- имеется инструментарий для создания приложений специализированных для QNX. Это WATCOM C++ 10.6, а также PhAB - RAD - система для разработки приложений под графическую оболочку Photon (также разработанной по микроядерному принципу).

Недостатком данной системы является, в первую очередь, высокая цена, как самой системы, так и средств разработки, которые поставляются отдельно и за дополнительную плату. Исходные тексты системы являются собственностью одной компании и не предоставляются разработчикам вообще. Одним из недостатков является небольшое различие между

компилятором WATCOM C++ и компиляторами C, принятыми на других платформах, что немного усложняет перенос программ под QNX, однако использование стандартов POSIX значительно облегчает перенос программ из среды QNX, например, в среду Linux. К недостаткам системы QNX можно отнести также сравнительно небольшой набор драйверов под данную систему, и в связи с этим довольно слабую поддержку аппаратуры.

Система QNX не поддерживает концепцию виртуальной памяти, это связано с тем, что она изначально была разработана как система реального времени, работающая на контроллере, у которого, возможно, отсутствует жесткий диск (загрузка производится из ПЗУ, flash-диска). Она изначально разрабатывалась как система для контроллеров и обладает достаточно высокой надежностью и гибкостью. Развитием ОС QNX является ОС QNX/Neutrino [41]. QNX/Neutrino поддерживает различные аппаратные платформы, в том числе и мультипроцессорные.

3.1.4 Операционная система OS-9

Операционная система OS-9 обладает высокой производительностью и гибкостью. Надежность, малые размеры и модульность OS-9 предопределили ее применение в таких видах и сферах деятельности, как промышленная автоматизация, контрольно-измерительные приборы и телекоммуникация, сбор данных, построение сетей, цифровая обработка изображений, построение сетей реального времени и др. Примеры применений OS-9 можно найти в интеллектуальных продуктах и системах таких фирм, как ABB Automation, Allen-Bradley, Boeing Aerospace, Harris Corporation, Honeywell Inc., Marconi Co., Philips GmbH, Soni, Hitachi, NASA, Texas Instruments Inc., Thomson Semiconductors и многих других. OS-9 - многозадачная, многопользовательская ОС РВ, разработанная фирмой Microware System Corporation (США).

В 1982 г. Microware адаптировала OS-9 для семейства микропроцессоров 68000, создав систему OS-9/680x0 для 16- и 32-разрядных микропроцессоров и микроконтроллеров. Код системы лишь на 20 % был написан на языке высокого уровня, остальная часть с целью достижения максимальной производительности написана на языке АССЕМБЛЕР. Через пять лет, (1987 г). OS-9/680x0 стала признанным промышленным стандартом для операционных систем реального времени и абсолютным лидером по применимости в промышленных приложениях на базе технологии VMEbus.

OS-9000 - переносимая версия OS-9, написанная главным образом (95 %) на языке Си. Оставшиеся, критичные с точки зрения производительности, участки кода написаны на языке АССЕМБЛЕР. Теоретически OS-9000 может быть перенесена на любую современную микропроцессорную архитектуру. Сегодня имеется поддержка со стороны новой версии OS-9000 V2.1 микропроцессоров: 680x0(x>2), Intel 80x86(x>3) и Power PC (40x,50x,60x,8xx), Motorola ColdFire, RISK ARM7100/7500FE, Hitachi SuperH RISK SH-3, MIPS.

Семейство операционных систем реального времени OS-9. Для пользователя OS-9 и OS-9000 - это UNIX - подобная среда. Модель процессоров, межпроцессорная коммуникация, многопользовательская файловая система и большое количество стандартных для UNIX утилит - все это позволяет программисту, знакомому с UNIX, в минимально короткий срок освоить среду реального времени OS-9. Единство архитектурных решений, совместимость приложений на уровне исполняемого кода, технология реализации интегральных средств разработки, а также дизайна и состава системных продуктов - эти факторы объединяет OS-9 и OS-9000 в семейство операционных систем реального времени OS-9.

В множестве приложений (см. табл. 3.1) семейство OS- 9 используется уже 18 лет, а число установленных копий превысило 5 миллионов.

Отрасли применений системы OS-9

Отрасли применений	Количество (в процентах)
Промышленная автоматизация	32
Бытовая электроника	18
Инструментальные и измерительные системы	17
Связь	14
Системы военного и аэрокосмического назначения	12
Другое	7

Система OS-9 является модульной, гибко конфигурируемой, высокопроизводительной, встраиваемой системой РВ. Совокупность требований, исходя из которых проектировалась OS-9, в самом общем случае сводится к обеспечению следующих потребительских свойств создаваемой ОС РВ: минимизации объема (компактность); гибкости (модифицируемость); возможности наращивания функций; доступности по цене (в первую очередь для массовой продукции); охват наиболее широкого спектра задач-приложений реального времени. OS-9 предоставляет собой профессиональный набор программных системных средств РВ, позволяющий в наиболее широком спектре приложений найти оптимальное сочетание производительности целевой системы РВ [42].

Модульность. В OS-9 применен модульный объектно-ориентированный подход в разработке архитектуры системы. Все функциональные компоненты OS-9, включая независимое (автономное) ядро, иерархические файловые менеджеры, систему ввода-вывода и средства разработки, реализованы в виде независимых модулей. Такая система может быть динамически легко реконфигурирована добавлением или удалением отдельных модулей во время работы системы без какой-либо recompilation объектного кода. А так как ОС полностью автономна от приложения, разработчик может выполнять такие операции, как загрузка и исполнение прикладной программы, запуск диагностики, добавление или исключение драйвера или изменение конфигурации системы путем простой команды с клавиатуры или посредством канала удаленной связи.

Ядро OS-9. В самой простейшей конфигурации OS-9 состоит из базового модуля - ядра. Оно выполняет следующие сервисные функции: диспетчеризацию процессов; многоуровневое, приоритетное обслуживание прерываний; обмен информацией между процессами; обработку ошибок, распределение и защиту системных ресурсов. Число базовых функций ядра может быть увеличено за счет написанных пользователем и готовых модулей расширения ядра (для управления FPU, SYSCash, питанием и т.д.), а также модулей управления файловым вводом/выводом.

В процессе работы над новыми версиями системы созданы различные, специфичные для типа процессора, ядра (в отличие от двух ядер общего назначения в OS-9/680x0 V2.4). Эта надстройка позволила оставить в ядре только тот код, который может поддерживаться процессором данного типа. Atomic-ядро OS-9 обеспечивает выполнение только тех функций, которые требуются во встроенной целевой системе, обеспечивая наиболее высокую производительность и детерминированность всех предоставляемых функций. В это же время было решено продолжать развивать стандартное полнофункциональное ядро, включающее все возможные способы улучшения быстродействия без уменьшения функциональных возможностей. Тем самым будет поддерживаться переносимость из системы с микроядром в систему с полнофункциональным ядром, как для прикладных процессов, так и для исполняющегося в системном состоянии ПО (типа обработчиков прерываний). Данное полнофункциональное ядро может поддерживать разработку и отладку как резидентного, так и основного ПО, обеспечивая одновременно адекватную для многих систем времени исполнения эффективность. Пользователи, которые хотят иметь в своей целевой системе микроядро, могут выполнить разработку своего приложения РВ с полнофункциональным ядром, а затем (после

отладки кода) заменить на время исполнения стандартное ядро микроядром. С точки зрения обычного пользователя, стандартное ядро и микроядро почти неотличимы друг от друга.

Надежность системы. Ядро OS-9 обеспечивает диспетчеризацию процессов на основе приоритетов с (необязательным) разделением времени, обслуживание прерываний, обмен информацией между процессами, обработку ошибок, распределение и защиту всех разделяемых системных ресурсов. При реализации каждой из указанных функций предпринимались все меры обеспечения максимальной степени надежности. К примеру, ядро управляет всеми разделяемыми ресурсами (получая и отвечая на запросы исполняющихся процессов) вместо того, чтобы просто позволить различным процессам конкурировать за разделяемый ресурс. При таком подходе гарантируется, что допущенная программистом ошибка при создании прикладной программы не приведет к потере всей системы. Кроме того, защита памяти (выделяемой ядром) выполняется диспетчером памяти. В системе имеются следующие требующие защиты ресурсы: содержимое памяти; регистры портов ввода/вывода; регистры системных часов; процедуры перехвата сигналов в процессах; процедуры обработки ошибок; интервалы приостановки исполнения; приоритеты диспетчеризации; пользовательские и групповые привилегии. При прекращении исполнения процесса (нормальным образом или аварийно) ядро определяет, какие ресурсы были связаны с этим процессом в момент прекращения выполнения, и их высвобождает.

Обслуживание прерываний. Прерывания в OS-9 обрабатываются соответствующими подпрограммами, которые могут быть установлены в ОС либо драйвером устройства, либо прикладным процессом. В большинстве ОС разрешена приостановка исполнения любого кода подпрограммой обработки прерываний, когда бы ни возникло данное прерывание (независимо от того, является ли этот код прикладной программой или подпрограммой обработки системной директивы), при этом должно гарантироваться, что подпрограмма обработки прерывания не будет модифицировать никакие разделяемые системные данные. Это обеспечивается с помощью запрета выдачи подпрограммой обработки прерываний каких-либо системных директив. Однако OS-9 изначально разрабатывалась с учетом того условия, что подпрограмма обработки прерываний может использовать системные директивы так же, как и исполняющиеся процессы. Это обеспечивает дополнительную гибкость разработки и оптимизацию параметров производительности прикладных программных систем РВ. Если некоторое прерывание не требует масштабного обслуживания, то оно может быть обработано соответствующей подпрограммой обработки прерываний (в том числе с применением какого-либо системного сервиса). В противном случае эта подпрограмма может активизировать обслуживающий процесс и вернуть управление. В первом случае, (если подпрограмма может обработать возникшее прерывание без помощи вспомогательного процесса) время отклика на прерывание определяется только временем ее исполнения. Если же подпрограмма активизирует специальный процесс, то время отклика на прерывание включает в себя требующееся для завершения работы данной подпрограммы время, а также время, необходимое для приостановки текущего процесса и возобновления исполнения требуемого процесса.

Системные директивы могут использоваться подпрограммами обработки прерываний для отправки сигналов, объявления событий, сбрасывания семафоров, установки или снятия аварийной ситуации (alarms), распределения и освобождения памяти, передачи информации другому процессу, изменения приоритета процесса. В предыдущих версиях OS-9 весь системный код исполнялся эффективно с максимально возможным приоритетом. Выполнение выданной процессом системной директивы могло быть прервано только подпрограммой обработки прерываний. Это привело к увеличению времени отклика на прерывание, особенно в тех приложениях, в которых одновременно исполнялись как процессы РВ, так и обычные процессы, так как после возврата управления из подпрограммы прерывания возобновляется исполнение системной директивы, ранее выданной прерванным процессом (возможно, с низким приоритетом). В результате этого нужный процесс не сможет отреагировать на возникновение прерывания, пока не завершится выполнение системной директивы. Если ее выполнение занимает много времени, то это существенно снижает реакцию системы на

события РВ. Для решения этой проблемы реализован иной механизм работы ядра: обработка системной директивы ведется с тем же приоритетом, что и выполнение выдавшего эту директиву процесса. При таком подходе любой процесс может вызвать любую функцию ядра, не влияя на время активизации процессов с более высоким приоритетом.

Средства обеспечения межпроцессной коммуникации. В качестве механизмов межпроцессной коммуникации OS-9 используются все имеющиеся и известные UNIX-примитивы: двоичные семафоры, сигналы, события, программные каналы, очереди, разделяемые модули данных.

Подпрограммы перехвата сигналов. Одной из важных особенностей OS-9 является наличие подпрограмм перехвата сигналов. Эта подпрограмма представляет собой необязательную процедуру обработки исключительных ситуаций, которая связывается с данным процессом. Если необходимо, чтобы процесс реагировал особым образом на посылаемые ему сигналы, то в него должна быть включена подпрограмма перехвата сигналов. При посылке сигнала ОС запоминает в состоянии «процесс» соответствующее сигналу целое значение и вызывает процедуру обработки сигнала в момент возобновления исполнения процесса. Исполнение данной подпрограммы во многом похоже на исполнение подпрограммы обработки прерываний. После своего завершения она вызывает специальную системную директиву, в результате которой управление возвращается в ту точку процесса, в которой произошла приостановка его выполнения.

Менеджер ввода-вывода. Унифицированная система ввода-вывода OS-9 имеет пять уровней и использует жестко стандартизованные системные директивы ввода-вывода (такие как Open, Close, Read и Write), тем самым предоставляет стандартный базис для файловых менеджеров и драйверов устройств. Общие для всех системных директив ввода/вывода ресурсы компонуется ОС до того момента, когда системная директива будет передана на обработку файловому менеджеру. Вся эта работа выполняется специально выделенным системным модулем менеджера ввода/вывода In-Out Manager (IOMan). Выделение ядра и функций ввода/вывода в отдельные модули, их многоуровневая организация не только улучшает гибкость системы, но и повышает производительность самого ядра.

Система ввода-вывода. Ограничением традиционных систем РВ является принятая организация ввода/вывода. Спроектированные для работы во встроенных контроллерах, большинство ядер РВ не поддерживают такие возможности расширенного ввода/вывода, как динамическая реконфигурация, многоуровневые директории и перенаправления ввода/вывода. Одна из главных проблем состоит в том, что разработчик должен применять аппаратно зависимые для конкретного встроенного применения вызовы. В большинстве ядер РВ приложения выполняют операции ввода/вывода, делая вызов подпрограммы из библиотеки ввода/вывода. Далее, перед тем как выполнять высокоуровневые операции ввода/вывода, разработчик должен определить механизм и разработать соответствующую задачу взаимодействия оборудования канала ввода/вывода с прикладной задачей. Для повышения гибкости разработки операционные системы, такие как UNIX и OS-9, рассматривают все устройства как файловые объекты и применяют технику, известную как перенаправление ввода/вывода, которая позволяет разработчику выполнять ввод/вывод без детальной спецификации устройства назначения. Конфигурация устройств в OS-9 может быть изменена в период исполнения с автоматическим добавлением соответствующих драйверов, что существенно упрощает реконфигурацию системы с новыми или дополнительными устройствами и процесс тестирования или отладки.

Файловые менеджеры. В OS-9 предоставляется самый широкий набор файловых менеджеров устройств ввода/вывода. Базовые файловые менеджеры OS-9 предназначены для обеспечения обмена информацией между процессами, а также обеспечивают приложениям OS-9 доступ к различным последовательным устройствам типа принтеров и терминалов и к устройствам внешней памяти типа дисков (жестких, гибких, электронных и оптических) и лент. Для поддержки таких сложных приложений, как телекоммуникация, мультимедиа и систем выдачи видеоданных по запросу, разработан ряд дополнительных файловых менеджеров, как:

- файловый менеджер управления стеком протоколов SPF, соответствующий технологии STREAM;

- менеджер сети ISND(LAP-B/LAP-D) с возможностью подключения к сетям коммутации пакетов X.25;

- файловый менеджер для приложений мультимедиа MPFM, способный обрабатывать в реальном масштабе времени поток видеоданных в стандарте MPEG-2, набор системных модулей для доступа к Internet JAVA/BROWSER;

- файловый менеджер работы с PCMCIA и др.

Модульная структура OS-9 позволяет системному разработчику выбирать именно те функциональные блоки, которые требуются данному приложению. Любая из опций OS-9 легко может быть добавлена в систему для поддержки изменившихся требований к системе.

Вопросы переносимости. Один из подходов к достижению переносимости - применения стандартизованных открытых решений. В OS-9 последовательно реализованы соответствующие средства, отвечающие POSIX, ANSI, C/C++, NFS Server & Client, X Windows Server & Client и Open Software Foundation. В OS-9 реализованы все функции стандарта POSIX.. Поддержка функций стандарта содержится в библиотеке POSIX.1 для компилятора Ultra C/C++.

3.1.5. Операционная система ОС5000

Операционная система ОС5000 [43] является одной из первых ОС РВ, разработанной в НПО «Импульс» для архитектуры Intel 8086. Основными областями применения ОС5000 являются автоматизированные системы управления технологическими процессами, функционирующие в реальном масштабе времени.

В ОС5000 в качестве основного режима функционирования реализован режим работы, включающий в себя следующие возможности:

- обеспечение псевдопараллельного выполнения нескольких последовательных программных процессов с использованием дисциплины приоритетов при распределении ресурсов;

- взаимодействие между псевдопараллельно выполняющимися процессами (задачами) с использованием аппаратов синхронизации по событиям, обмена сообщениями, семафоров и других;

- выполнение задач в соответствии с временным расписанием;

- фиксация различного рода событий с указанием времени их возникновения с высокой точностью.

Управление программам ОС передается, во-первых, из обрабатывающих программ по специальным вызовам, называемым в дальнейшем макрооперациями, во-вторых, при возникновении аппаратных прерываний.

3.1.5.1. Управление задачами

В операционной системе допускается псевдопараллельное функционирование до 100 задач. Каждой задаче ставится в соответствие следующие (внешние) атрибуты: имя, представляющее собой последовательность алфавитно-цифровых символов в количестве от 1 до 4; приоритет, представляющий собой целое число в пределах от 1 до 127, определяющее "важность" задачи при предоставлении ей ресурсов, в частности процессорного времени (меньшему числу, соответствует высший приоритет); идентификатор, представляющий собой целое число без знака, используемое для ссылок на задачу в процессе функционирования системы.

Все данные о задаче хранятся в так называемом блоке управления задачей (БУЗ). В БУЗ записаны внешние атрибуты и информация о состоянии выполнения задачи. В нем также находятся различные структуры, связанные с взаимодействием задач (в частности, "почтовые ящики" для приема адресованных задаче сообщений). Предоставление задачам процессорного времени производится в соответствии с их приоритетами. Это означает, что процессор предоставляется той задаче из числа готовых к выполнению, которая имеет наивысший приоритет, переводя ее из состояния "готова к выполнению" в состояние "выполнение". На рис. 3.3 представлена диаграмма состояний задачи. Рядом со стрелками указаны основные макрооперации, в результате выполнения которых изменяется состояние задачи.

Другим задачам, имеющим более низкий приоритет, процессор предоставляется тогда, когда выполняющаяся задача переходит из состояния "выполнение" в состояние "ожидание", "бездействие" или "пауза". Если две или более задачи имеют одинаковый приоритет и они "готовы к выполнению", то выбор задачи, которой должен быть предоставлен процессор, реализуется одним из двух следующих методов: 1 - в соответствии с очередностью создания задач; 2 - путем квантования времени работы. Выбор метода осуществляется на этапе создания конкретного варианта ОС и может изменяться в процессе функционирования.

Взаимодействие между задачами. В ОС реализованы следующие методы взаимодействия между задачами: посредством обмена сообщениями; с использованием аппарата событий; с использованием семафоров; путем организации управления общими ресурсами; с использованием общих буферных пулов; с использованием общих областей оперативной памяти; посредством организации списковых структур. Взаимодействие между задачами

посредством обмена сообщениями предусматривает передачу данных, функционально аналогичную обмену письмами между людьми. А именно, каждой задаче поставлено в соответствие четыре так называемых "почтовых ящика", через которые она может получать различные сообщения-письма от других задач. Сообщение представляет собой массив данных размером 12 байтов. Сообщения передаются через "конверты" - элементы, специально предназначенные для этой цели.

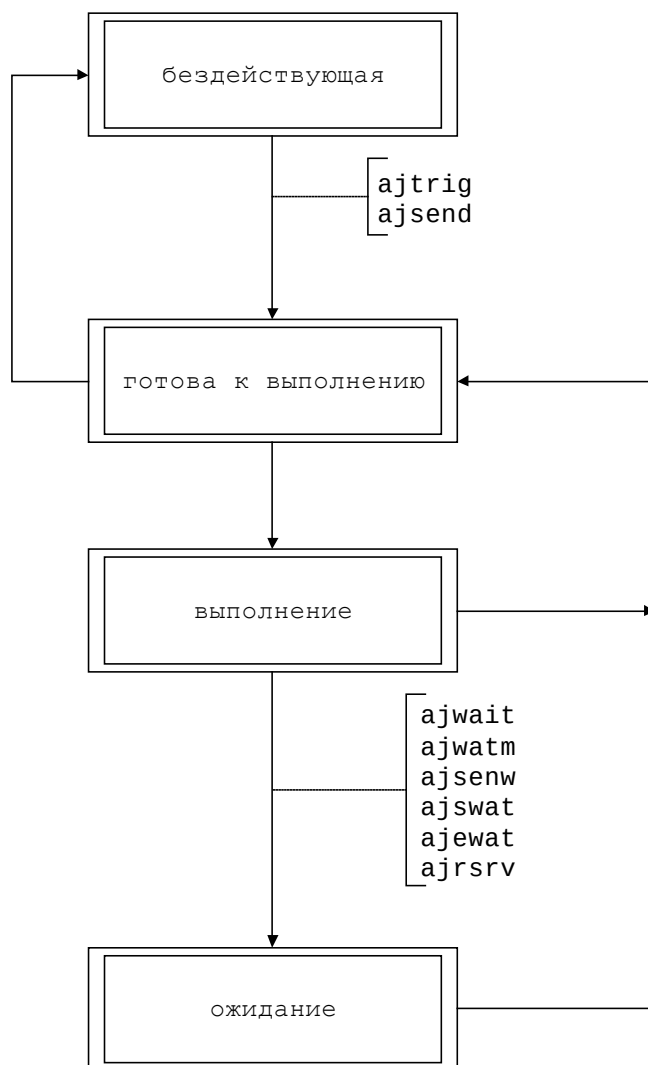


Рис. 3.3 Диаграмма изменения состояния задач.

В ОС предусмотрены различные макрооперации, позволяющие организовать передачу сообщений, в частности, задача может быть переведена в состояние ожидания поступления ей сообщений. При этом можно указать ограничения на время ожидания. Следует заметить, что поступающие задаче сообщения "выстраиваются" в очереди, количество которых равно 4 (по количеству "почтовых ящиков"). "Почтовые ящики" идентифицируются своими номерами - 0, 1, 2 и 3. Номер является не только идентификатором, но и определяет приоритет поступающих к этому "почтовому ящику" сообщений: чем меньше номер, тем выше приоритет.

Аппарат событий предоставляет задачам универсальные способы синхронизации своего выполнения в соответствии с различными происшествиями вне их. Две или более задачи, взаимодействуя между собой с использованием аппарата событий, связываются через так называемую группу событий - 16-разрядное слово, каждому разряду которого ставится в соответствие определенное событие. Динамическое выделение группы событий (с формированием ее идентификатора - числа в пределах от 1 до 127 включительно)

осуществляется по специальной макрооперации. Задача имеет возможность перейти в состояние ожидания обобщенного события, заключающегося в присваивании указанного кода определенным разрядам группы событий или заключающегося в появлении единицы хотя бы в одном из заданных разрядов группы. Кроме описываемого аппарата в ОС предусмотрена возможность переходить в состояние ожидания заранее определенного события, не указывая группу. В этом случае аналогом идентификатора группы событий выступает идентификатор задачи, ожидающей события.

Семафор в ОС5000 - это динамически выделяемый по специальной макрооперации элемент, предназначенный для организации описываемого взаимодействия. Если несколько задач должны совместно использовать общий неделимый ресурс (ресурс с последовательным доступом), то "регулирование" их отношений (захват ресурса и его освобождение) осуществляется с помощью динамически выделяемых по специальным макрооперациям блоков управления ресурсами. Управление памятью в задачах осуществляется как с использованием так называемых буферных пулов, так и с использованием динамического распределения областей памяти.

Для организации различного рода объединений элементов в ОС предусмотрен так называемый кольцевой список, представляющий собой совокупность однобайтных, двухбайтных или четырехбайтных элементов. Для кольцевого списка характерными являются операции включения элемента первым или последним в список, а также исключение первого или последнего элемента из этого списка.

Основные понятия "службы времени" ОС. Организация "службы времени" в ОС основана на аппаратно выполняемых прерываниях, возникающих через строго определенные промежутки времени, называемыми "тиками". Величина тика устанавливается при создании конкретного варианта ОС и является минимальной единицей отсчета различных интервалов. Счет астрономического (календарного) времени суток и даты осуществляется путем наращивания соответствующих величин по прерываниям. Передача в задачу текущих значений времени суток и даты выполняется по специальной макрооперации.

При создании конкретного варианта ОС помимо величины тика устанавливается также величина "кванта", которая кратна величине тика. Квант - это промежуток времени, в течение которого выполняющаяся задача не прерывается другой, имеющей такой же приоритет и готовой к выполнению. В ОС допускается динамическое создание "индивидуальных" интервальных таймеров для отсчета промежутков времени. При создании каждому интервальному таймеру ставится в соответствие блок его управления и идентификатор - число в пределах от 1 до 65535 включительно. В системе допускается одновременное использование до 1024 интервальных таймеров.

Функциональная структура ОС. С функциональной точки зрения ОС состоит из следующих подсистем:

- подсистемы инициализации работы ОС;
- ядра ОС (в свою очередь включающего планировщик задач и программы выполнения макроопераций управления задачами);
- подсистемы обработки прерываний (в состав которой, в частности, входит программа обработки прерываний от системного таймера);
- специальную системную задачу, обеспечивающую работу "службы времени";
- набора программ-администраторов.

Инициализация и завершение работы ОС. Инициализация работы ОС осуществляется сразу после запуска и включает в себя выполнение так называемых "рестарт-процедур". Первой из рестарт-процедур выполняется макрооперация ENTR, в которой присваиваются начальные значения внутренним системным переменным ОС, инициализируются основные процессы и выполняются другие операции, приводящие систему в рабочее состояние. В процессе выполнения ENTR осуществляется вызов рестарт-процедур, определенных на этапе создания ОС (в том числе и рестарт-процедур пользователя). В этих процедурах осуществляется (при необходимости) создание задач, которые автоматически запускаются на выполнение после

запуска ОС, создаются в требуемом количестве интервальные таймеры, присваиваются начальные значения переменным и т.п. Завершение работы ОС осуществляется по вызову макрооперации EXIT. Имеется возможность аварийно завершить работу ОС, вызвав макрооперацию FATH. При этом выполняется ряд макроопераций (называемых процедурами выхода), определенных при создании ОС. В состав процедур выхода могут входить пользовательские процедуры, выполняющие специфические действия по отношению к задачам.

Планировщик задач. Управление планировщику задач передается каждый раз, когда происходят какие-либо события, в результате которых может измениться набор готовых к выполнению задач. В частности, планировщик включается после завершения инициализации. Цель работы планировщика - выбор старшей по приоритету готовой к выполнению задачи и перевод ее в состояние "выполнение". Смена выполняющейся задачи на другую осуществляется путем переключения контекста. Вызов планировщика осуществляется по завершении обработки прерывания (при выполнении макрооперации INX), а также в макрооперациях, изменяющих состояние задач.

Подсистема обработки прерываний. Подсистема обработки прерываний состоит из процедур обработки конкретных прерываний (ПОКП). Управление ПОКП передается непосредственно при возникновении прерывания через таблицу прерываний, содержащую адреса ПОКП. При этом в стеке выполняющейся задачи автоматически сохраняются адрес возврата и содержимое регистра флагов. При этом система прерываний выключена. В начале обработки прерывания запоминается содержимое всех программно доступных регистров. Для этого используется вызов макрооперации INT. В конце ПОКП должно выполняться восстановление запомненного содержимого регистров и возврат к прерванной программе. Так как при обработке прерывания возможно изменение выполняющейся задачи, то следует вызвать планировщик задач. Вызов его осуществляется в макрооперации INX, которая обеспечивает также восстановление содержимого регистров. В процессе обработки прерывания допускается использование любых макроопераций ОС. При этом каждый вызов выполняется от имени прерванной задачи. Частным случаем ПОКП является процедура обработки прерываний от таймера. Когда произошло прерывание от таймера, наращивается текущее время суток и, если необходимо, корректируется дата. Далее, если в системе выполняются процессы, связанные с отсчетом временных интервалов, то специальная системная задача обеспечения работы "службы времени", имеющая наивысший приоритет в ОС, переводится в состояние готовой к выполнению.

Задача обеспечения работы "службы времени". Задача обеспечения работы "службы времени" имеет наивысший приоритет в ОС и предназначена для отслеживания различных процессов, связанных с отсчетом промежутков времени. По исчерпанию промежутка в этой задаче выполняются соответствующие действия по извещению других задач о происшедшем событии. Эта же задача обслуживает интервальные таймеры.

3.1.5.2. Программы-администраторы ОС

Некоторые функции могут включаться или не включаться в состав конкретного варианта ОС в зависимости от требований применения системы. Эти функции реализуются в так называемых программах-администраторах ОС. В общем виде администратор ОС представляет собой совокупность управляющих и обрабатывающих программ, предназначенных для выполнения определенной функциональной группы макроопераций. В состав ОС могут включаться следующие администраторы: управления событиями, семафорами, ресурсами и буферными пулами; динамического распределения памяти; преобразований форматов представления времени и даты (в дальнейшем времени и даты); управления кольцевыми списками; супервизор.

Администратор управления событиями представляет собой совокупность программ, выполняющих макрооперации, связанные с взаимодействием между задачами с использованием аппарата событий. Инициализация работы администратора управления

событиями осуществляется по макрооперации EMRR. Макрооперации EGET и ERLS предназначены для выделения групп событий и возврата их в ОС. С помощью макрооперации EWAT, задачи могут переводиться в состояние ожидания обобщенных событий, а с помощью макрооперации ESIG - устанавливать в группе событий требуемые значения разрядов, обеспечивая тем самым включение (перевод в состояние "готова к выполнению") задач, ожидающих обобщенные события.

Администратор управления семафорами представляет собой совокупность программ, выполняющих макрооперации, связанные с взаимодействием между задачами с использованием семафоров. Семафор представляет собой целочисленную переменную, принимающую положительные, отрицательные и равные нулю значения. Если значение переменной отрицательное или равное нулю, то семафор "закрыт". В этом случае, обращающиеся к нему задачи переводятся в состояние ожидания "открытия" и ставятся в очередь, упорядоченную по приоритету, передаваемому в вызове макрооперации SWAT, по которой обеспечивается разрешение на доступ задачи к "критическому" ресурсу. Если же значение переменной положительно, то по макрооперации SWAT из него вычитается единица и задача не переводится в состояние ожидания и не ставится в очередь. Ей тем самым разрешается работа с "критическим" ресурсом. Действия, выполняемые по макрооперации SWAT, в литературе обычно называются Р-операцией. По завершении работы с "критическим" ресурсом задача обязана выполнить так называемую V-операцию, осуществляемую с помощью вызова макрооперации SSIG. Смысл V-операции заключается в наращивании на единицу значения переменной семафора. Если оно стало положительным и очередь задач, ожидающих "открытие" семафора, не пустая, то первая в ней задача исключается из очереди и вновь выполняется Р-операции, заданная ею в вызове макрооперации SWAT. При вызове макрооперации SWAT указывается приоритет доступа к семафору - число в пределах от 0 до 255 включительно, по возрастанию которого упорядочивается очередь задач, ожидающих "открытие" семафора. В этом же вызове указывается величина максимального промежутка времени ожидания или признак того, что время ожидания не ограничено. Если задача не дождалась "открытия" семафора в течение этого промежутка, она исключается из очереди и переводится в состояние готовой к выполнению, но при этом в качестве результата макрооперации SWAT ей передается код (код завершения), указывающий на то, что семафор для нее не "открыт". Создание семафора производится по макрооперации SCRE. При этом формируется так называемый блок управления семафором - дескриптор, занимающий 8 байтов и используемый в вызовах SWAT и SSIG в качестве идентификатора семафора. В этом блоке содержится и переменная семафора, начальное значение которой указывается в вызове макрооперации SCRE. Кроме того, при вызове SCRE указывается максимальное количество задач, которые одновременно могут ожидать "открытия" семафора. После завершения работы с семафором он может быть уничтожен по макрооперации SDEL.

Администратор управления ресурсами. Представляет собой совокупность программ, выполняющих макрооперации, связанные с распределением между задачами ресурсов с последовательным доступом. Управление последовательным ресурсом базируется на двоичном семафоре, однако в отличие от описанного выше аппарата семафоров задачи, ожидающие предоставления ресурса, упорядочены в соответствии с дисциплиной простой очереди, а также не имеется возможности ограничивать время ожидания. Создание ресурса задач, осуществляется по макрооперации RCRE. Уничтожается ресурс по макрооперации RDEL. Макрооперация RSRV служит для захвата ресурса. По окончании работы с ресурсом его освобождение осуществляется по макрооперации RELS. По макрооперации RMRR, являющейся рестарт-процедурой, инициализируется работа администратора управления ресурсами.

Администратор управления буферными пулами представляет собой совокупность программ, выполняющих макрооперации, связанные с ведением пулов буферов (создание пула, выделение и возврат буферов и т.п.). Администратором обеспечивается быстрый и эффективный доступ к нескольким пулам, каждый из которых содержит буферы

фиксированной длины, что позволяет устранить фрагментацию памяти. Предусмотрено разделение буферов между задачами и совместное владение несколькими задачами одним буфером. Создание буферного пула осуществляется по макрооперации BCRE. Максимальное число одновременно созданных пулов (не превышающее 4095) указывается в процессе получения конкретного варианта ОС. Размер буферов в пуле может варьироваться от 2 до 65535 байтов. Все буферы пула должны располагаться в одном сегменте. Помимо размера буферов в вызове BCRE указывается количество буферов в пуле и адрес (указатель) области памяти, отведенной для буферов. Каждому выделенному из пула буферу ставится в соответствие счетчик использования. Этот счетчик наращивается всякий раз, когда буфер начинает использоваться задачей, и уменьшается по окончании работы с ним (действия по изменению счетчика использования на произвольное число выполняются макрооперацией BAU; при выделении буфера из пула по макрооперации BGB значение счетчика устанавливается равным 1, а при возврате буфера в пул по макрооперации BRB, он уменьшается на 1). Возврат буфера в пул выполняется только тогда, когда значение счетчика его использования стало равным 0. По окончании работы с пулом он может быть уничтожен. Уничтожение производится по макрооперации BDEL. Буферные пулы (один или несколько) могут быть определены на этапе создания ОС. В этом случае с помощью макроопераций BIA и BIP должны быть выполнены процедуры инициализации работы с ними. В состав администратора управления буферными пулами входит вспомогательная макрооперация BGS, с помощью которой можно получить размер буферов (в байтах) в указанном в вызове макрооперации буферном пуле. Инициализация работы администратора управления буферными пулами осуществляется с помощью макрооперации BMRR, являющейся соответствующей рестарт-процедурой. Следует заметить, что использование буферных пулов в качестве средств динамического управления распределением памяти характерно для систем реального времени, так как другие аппараты аналогичного назначения существенно более медленные.

Администратор динамического распределения памяти представляет собой совокупность программ, выполняющих макрооперации, и связанные с обслуживанием динамически распределяемой памяти: MAU, MFRE, MGEN, MGET, MGSZ, MHAN, MSET. Эти процедуры используются тогда, когда требуется работа с областями оперативной памяти, длины которых варьируются в широких пределах (в том числе могут превышать 64 Кбайт), что не позволяет эффективно использовать буферные пулы. Для выделения области памяти, определяемой блоком памяти, задаче следует обратиться к администратору, выполнив макрооперацию MGET. При этом для выделенного блока устанавливается равным единице так называемый счетчик использования, имеющий аналогичное со счетчиком использования буфера назначение, связанное с параллельной работой нескольких задач с выделенным блоком. После завершения работы возврат блока системе осуществляется по макрооперации MFRE. При этом счетчик использования блока уменьшается на 1 и собственно возврат выполняется только в том случае, когда его значение стало равным 0. Изменение счетчика использования выделенного блока на произвольное число осуществляется по макрооперации MAU.

Выделение области динамически распределяемой памяти осуществляется на этапе создания ОС. Однако в процессе работы допускается создание частных областей динамически распределяемой памяти, называемых разделами. Эти области создаются в выделенных блоках памяти с помощью макроопераций MHAN. При этом формируется блок управления разделом, используемый в качестве его идентификатора. Выделение блока памяти из раздела осуществляется по макрооперации MGEN, в вызове которой (в отличие от вызова макрооперации MGET) указывается блок управления разделом.

В состав администратора динамического распределения памяти входят две вспомогательные макрооперации MGSZ и MSET. С помощью первой из них можно получить размер выделенного блока памяти в байтах, а с помощью второй - записать в произвольную область памяти заданный код.

Администратор времени и даты. По этой макрооперации выполняется установка текущих значений времени суток и даты для MS DOS равными текущим времени и дате в ОС.

В состав администратора времени и даты входят программы, предназначенные для реализации макроопераций TDEX, TDF, TDG, TDPC, TDRR и TDS. Эти макрооперации в основном используются для получения задачами текущих времени суток и даты в требуемом формате, а также для первоначальной установки их значений.

Администратор управления кольцевыми списками представляет собой совокупность программ, выполняющих макрооперации, обеспечивающие формирование и ведение кольцевых списков различного назначения. В состав администратора входят программы, реализующие макрооперации ABL, ATL, RBL, RSTL и RTL. Инициализация работы администратора времени и даты осуществляется по макрооперации TDRR или по макрооперации TDPC. Обе макрооперации являются рестарт-процедурами. Кольцевой список создается по макрооперации RSTL. При этом указывается количество мест в списке (число n , варьируемое в пределах от 1 до 16280 включительно), размер элемента списка в байтах. По макрооперации TDPC исходные значения времени и даты формируются равными текущим их значениям в операционной системе MS DOS в момент выполнения TDPC (число s , равное 1, 2 или 4) и область памяти для организации списка размером $(n*s+8)$ байтов. Вся область должна располагаться в одном сегменте. Установка и изменение времени суток и даты осуществляются с помощью макрооперации TDS. Включение элемента в список первым осуществляется по макрооперации ATL, а последним - по макрооперации ABL. Исключение элемента из списка осуществляется по макрооперациям RTL (исключается последний элемент) и RBL (исключается первый элемент). Получение задачами текущих значений времени суток и даты осуществляется с помощью макрооперации TDF или макрооперации TDG.

Администратор управления кольцевыми списками автоматически включается в состав любого варианта ОС.

3.1.5.3. Супервизор

Супервизор представляет собой набор средств ОС, обеспечивающих работу последней в среде MS-DOS. А именно, в состав супервизора входят средства, предназначенные для санкционированного использования следующих аппаратных средств вычислительного комплекса: клавиатуры, телевизионного монитора, устройства печати, каналов асинхронной связи (с интерфейсом RS-232C), накопителей на жестких магнитных дисках, накопителей на гибких магнитных дисках, таймера, устройства звуковой сигнализации. В состав супервизора входят средства, обеспечивающие распределение между задачами устройств с последовательным доступом (клавиатур, каналов асинхронной связи, устройств печати) и управление работой перечисленных устройств. Вызовы операций MS-DOS из задач (в том числе и обращения к базовой системе ввода-вывода - BIOS) осуществляются так же, как это делается из программ, предназначенных для работы под управлением MS-DOS. В частности, большинство запросов, адресованных к BIOS, перехватывается супервизором и обрабатывается по правилам ОС, но при этом сохраняются все интерфейсы и форматы данных.

Супервизором перехватываются также вызовы прерываний "INT 20H" и "INT 27H" (завершение работы программы в MS-DOS) и заменяются на вызов макрооперации END. Аналогичные действия выполняются и при обращении к функциям MS-DOS, предназначенным для обработки завершения программ. Следует заметить, что в отличие от "обычных" программ модули обработки команд оператора MS-DOS не должны оставаться резидентными в памяти после завершения. Для обработки ненормальных ситуаций, возникающих при использовании вызовов MS-DOS, предусмотрена возможность указания специфической процедуры, которой передается управление при выявлении соответствующих ошибок (в частности, по ошибкам ввода-вывода). ОС предоставляет три уровня выполнения операций ввода-вывода:

- с использованием макроопераций и процедур супервизора;
- с использованием вызовов функций MS-DOS;
- с использованием вызовов процедур BIOS.

В супервизор включены средства, обеспечивающие операции ввода/вывода с клавиатурой, таймером и устройством выдачи звукового сигнала. При работе с этими средствами супервизором обеспечивается псевдопараллельное выполнение операций. В частности, в состав супервизора включены макрооперации работы с клавиатурой (KBGC и KBSS) и выдачи звукового сигнала (BEEP, SPOF и SPON).

При использовании вызовов функций MS-DOS, обеспечивающих выполнения операций ввода/вывода, следует помнить, что подпрограммы, реализующие эти операции, не являются реентерабельными (допускающими повторный вход). Поэтому задачи, использующие вызовы функций MS-DOS, должны обеспечивать последовательное выполнение операций. Соответствующие средства для этого имеются в составе супервизора. А именно, задача перед выполнением одной или нескольких операций ввода/вывода должна осуществить захват MS-DOS как ресурса (с помощью макрооперации PDRP или PDRS), а после их выполнения - освободить MS-DOS (по макрооперации PDRL). В отличие от первого и третьего методов выполнения операций в данном случае происходит существенное замедление задач, связанных с работой устройств ввода-вывода. Например, если задача А выполняет какие-либо действия, используя диалог с человеком-оператором, то задача В, которой требуется вывести на печать некоторое сообщение, будет приостановлена до завершения работы с устройствами в задаче А (до освобождения MS-DOS как ресурса).

При обращении задач к устройствам ввода/вывода через вызовы функций BIOS рекомендуется для каждого устройства ввода/вывода создать специальную задачу, предназначенную для выполнения операций на устройстве по запросам от остальных задач. Такие задачи могут выполняться псевдопараллельно, и поэтому данная методика проектирования системы является наиболее предпочтительной. Выполнение операций доступа к дисковым файлам следует осуществлять, используя соответствующие вызовы функций MS-DOS, захватывая ее как ресурс перед выполнением операции и освобождая после ее выполнения. Следует помнить, что с каждым конкретным устройством можно работать либо с использованием вызовов MS-DOS, либо с использованием вызовов BIOS, либо по макрооперациям супервизора ОС. Параллельное использование разных методов в одной или в разных задачах может привести к непредсказуемым последствиям. Если в системе имеется всего одна задача, выполняющая операции ввода/вывода и операции доступа к файлам, то захват и освобождение MS-DOS как ресурса не требуются. Тем самым обеспечивается возможность выполнения как задач ОС обрабатывающих программ, рассчитанных на работу в среде MS-DOS.

Основные недостатки системы ОС 5000. Так как основным режимом данной системы является "реальный" режим процессора Intel 8086, то ей присущи следующие недостатки:

- оперативная память ограничена 640 KB, что накладывает ограничения на объем исполняемых в системе задач;
- все программы и операционная система исполняются в одном адресном пространстве, что может привести к тому, что любая программа своими некорректными действиями может изменять данные и код у другой задачи и у операционной системы, что в конечном итоге приведет к краху системы, а также затрудняет отладку приложений для данной системы;
- в ОС и поставляемой с ней менеджерах - администраторах отсутствуют средства сетевой поддержки, что приводит к необходимости разрабатывать каждый раз заново драйверы вновь появляющихся сетевых устройств;
- ОС 5000 не поддерживает никакой из известных стандартов на API (пользовательский программный интерфейс), поэтому ее приложения (задачи) несовместимы с другими ОС, что затрудняет перенос программ на другие платформы (программные и аппаратные);
- в данной системе отсутствуют какой-либо инструментарий для системного программирования, а именно: для написания программ, драйверов и подсистем используется кросскомпиляция, т.е. программы разрабатываются под MS DOS и транслируются под целевую платформу. В качестве трансляторов применяются Borland C++ 3.1 и Turbo Assembler. В связи с

широким использованием Ассемблера система и все программное обеспечение является сильно машинно-зависимым. Кросскомпиляция также осложняет отладку программ;

- программа имеет возможность запретить прерывания, что может привести к потере работоспособности системы;

- фирмы производители современных аппаратных средств не поддерживают данную операционную систему, что приводит к необходимости заново разрабатывать драйвера для новых устройств;

- система ОС 5000 не поддерживает концепцию виртуальной памяти;

- малая известность системы.

Таким образом, можно сделать выводы, что данная система, хотя и обладает некоторыми достоинствами, такими как небольшой требуемый объем оперативной и внешней памяти, микроядерный способ построения и др., но уже морально устарела.

3.1.6 Операционная система WINDOWS NT

Windows NT - многозадачная операционная система разработки фирмы Microsoft. Основными достоинствами Windows NT по сравнению с Windows 95 и Windows 98 является повышенная надежность и переносимость на другие аппаратные платформы. Windows NT может использоваться на верхнем уровне АСУ ТП - в операторских станциях с функциями SCADA – систем, функционирующих в «мягком реальном времени».

3.1.6.1. Система управление памятью

Система управления памятью (встроенная в ядро Windows NT [44]), является одной из самых сложных и совершенных. Сложность системы управления памятью обусловлена тем, что она использует все возможности современных процессоров, в том числе механизм страничной адресации памяти. В основе системы управления памятью Windows NT лежит система виртуальной памяти, встроенная в ядро ОС. Каждое приложение может использовать до 2 GB виртуальной памяти. Причем для каждого приложения выделяется отдельное адресное пространство виртуальной памяти. Для создания виртуальной памяти используются дисковые устройства. Система виртуальной памяти Windows NT позволяет создать до 16 отдельных файлов страниц, расположенных на разных дисковых устройствах, установленных в компьютере. ОС Windows NT использует все возможности защищенного режима процессора, в частности - переключение задач и страничную организацию. Используя 32-разрядное смещение, приложение может работать с памятью объемом 4 GB без применения сегментных регистров процессора. Сегментные регистры процессора, хранящие селекторы, заполняются операционной системой и приложение не должно их изменять.

Несегментированная модель памяти называется сплошной моделью памяти FLAT. При программировании в этой модели памяти не требуются ключевые слова, так как все объекты, расположенные в памяти, адресуются с помощью только одного смещения. Таким образом, каждому приложению (или процессу) предоставляется линейное адресное пространство размером 4 GB. Область размером 2 GB предоставлена приложению, другие 2 GB адресуемого пространства зарезервированы для использования ОС. Разделение адресных пространств выполняется с помощью назначения приложениям индивидуальных наборов таблиц страниц виртуальной памяти. В результате для каждого приложения выполняется отображение линейных адресов в собственный набор страниц виртуальной памяти, не пересекающийся с набором страниц других приложений.

Полное изолирование адресных пространств создает трудности при необходимости организации обмена данными между различными приложениями. Для организации обменов применяются дескрипторы прототипа PTE (Prototype Page Table Entry). Дескрипторы PTE создаются ОС для совместного использования страниц, содержащих исполняемый код, а также

для работы с файлами, отображаемыми на память. Есть также способ организации общей памяти при помощи библиотек динамической компоновки DDL. Таблица страниц содержит дескрипторы, описывающие отдельные страницы памяти. Эти дескрипторы содержат физические адреса страниц, а также другую информацию. Формат дескриптора страницы приведен на рис. 3.4.

Защита	Физический адрес страницы	Файл страницы	Состояние

Рис. 3.4. Формат дескриптора страницы.

Физический адрес страницы имеет 20 разрядов. Для получения 32-разрядного физического адреса байта внутри страницы к нему добавляется 12 *byte* смещения, взятые из линейного адреса. Устанавливая соответствующим образом биты защиты, операционная система может отметить страницу как: доступную для чтения и записи; доступную только для чтения; недоступную. При попытке выполнить обращение для выполнения неразрешенной операции возникает прерывание. Биты с 3-го по 6-й содержат номер файла страниц, в котором находится страница, соответствующая данному дескриптору. Биты с нулевого по второй описывают состояние страницы памяти. Страница может быть отмечена флагами Т (находится в переходном состоянии), D (обновленная, но не сохраненная в файле страниц) и Р (присутствующая в памяти). Если приложение выполняет попытку обращения к странице памяти, которой нет в памяти, возникает прерывание и нужная страница автоматически читается из соответствующего файла страниц в физическую оперативную память. После этого работа приложения продолжается. В дополнение к 3 битам состояния страниц, хранящегося в дескрипторе страниц, система управления виртуальной памятью хранит состояние страниц в специальной базе данных страниц. В этой базе данных страница может быть помечена как имеющая одно из состояний, указанных в табл.3.3.

Таблица 3.3

Состояния страницы	
Состояние страницы	Описание
Свободная	Страница доступна для использования после ее заполнения нулями
Заполненная нулями	Свободная страница, заполненная нулями и доступная для использования приложениями
Правильная	Страница используется активным процессом
Измененная	Содержимое страницы было изменено, однако страница не была еще сохранена на диске в файле страниц
Запасная	Страница удалена из рабочего набора страниц процесса
Плохая	При обращении к этой странице возникла аппаратная ошибка

Неисправные страницы будут отмечены в базе данных страниц как плохие, и к ним не будет выполняться обращение.

Функции для работы с виртуальной памятью. В программном интерфейсе ОС Windows NT имеются средства для работы с виртуальной памятью. Они работают на уровне страниц памяти, имеющих размер 4096 В, поэтому обычно нет смысла использовать эти функции только для того, чтобы заказать в программе буфер размером в несколько десятков байт. У приложения есть две возможности заказать страницы виртуальной памяти. Первая возможность заключается в резервировании заданного диапазона адресов в адресном пространстве приложения, а вторая - в фактическом получении в пользование страниц виртуальной памяти, к которым можно выполнять обращение. При резервировании приложение только отмечает область памяти, лежащую в заданном диапазоне адресов, как зарезервированную. Если приложение получает страницы памяти для непосредственного использования, эти страницы физически создаются в виртуальной памяти и заполняются нулями. При этом может

происходить запись в файлы страниц. Такой процесс продолжительнее, чем резервирование. Для того чтобы зарезервировать страницы виртуальной памяти, приложение должно воспользоваться функцией VirtualAlloc с указанием необходимых параметров. Освобождение полученной ранее виртуальной памяти осуществляется с помощью функции VirtualFree.

Для получения области памяти, гарантированно размещенной в физической оперативной памяти, существует функция VirtualLock. Эта память может быть необходима, например, для размещения драйвера периферийного устройства.

В программном интерфейсе ОС Windows NT есть средства для получения справочной информации об использовании процессами виртуальной памяти. Процесс может исследовать как собственное адресное пространство, так и пространство других процессов, определяя такие характеристики областей виртуальной памяти, как размеры, тип доступа, состояние и т. д.

Мультизадачность. ОС Windows NT работает в режиме вытесняющей мультизадачности, когда все приложения гарантированно получают кванты времени по прерыванию от таймера. В ОС Windows NT существует два понятия, имеющие отношение к мультизадачности. Это процессы и задачи. Процесс создается, когда программа загружается в память для выполнения. Сразу после запуска процесса создается задача (или поток). При необходимости задача может запускать другие задачи. Все задачи имеют доступ к памяти, выделенной запустившему их процессу. В ОС Windows NT могут работать одновременно несколько процессов, а в рамках каждого процесса могут параллельно работать несколько задач.

В ОС Windows NT используется приоритетное планирование, когда и процессы и задачи имеют свои уровни приоритета. ОС Windows NT устанавливает уровень приоритета задач от 1 до 31, причем значение 31 соответствует максимальному приоритету. В процессе планирования кванты времени выделяются задачам с максимальным приоритетом. Менее приоритетные задачи получают управление лишь в том случае, если более приоритетные задачи переходят в состояние ожидания. В ОС Windows NT используется двухступенчатая система установки приоритетов и задач. При запуске процесса ему можно назначить один из четырех классов приоритета: процесс реального времени, высокоприоритетный процесс, обычный процесс, низкоприоритетный процесс. Когда приоритет процесса не указывается, то по умолчанию он получает приоритет класса обычного процесса.

Класс низкоприоритетных процессов используется для фоновых приложений, которые не должны тормозить работу других. Когда приложение должно немедленно отзываться на действия пользователя, ему следует назначить класс высокоприоритетных процессов. Класс приоритета процессов реального времени должен использоваться только системными программами или драйверами. В рамках одного процесса может быть запущено несколько задач. Процесс устанавливает относительный приоритет задачи, который может быть ниже или выше приоритета процесса. ОС может автоматически изменять приоритет задач, повышая его, когда задача начинает взаимодействовать с пользователем, а затем постепенно его уменьшая. Приоритет задач, находящихся в состоянии ожидания, также уменьшается. Процесс может запустить задачу, а затем увеличить ее приоритет. В этом случае главная задача процесса, запустившая более приоритетную задачу, будет временно приостановлена. Если же процесс запустит задачу и уменьшит ее приоритет таким образом, что он станет меньше приоритета главной задачи процесса, то будет приостановлена запущенная задача.

Для организации взаимодействия задач в ОС Windows NT предусмотрены объекты синхронизации - средства организации последовательного использования ресурсов (mutex), семафоры и события.

3.1.6.2. Файловая система

ОС Windows NT имеет развитые средства работы с файлами. Она способна выполнять операции над файлами, расположенными в нескольких файловых системах, таких, как FAT (MS-DOS), HPFS (IBM OS/2), NTFS (ОС Windows NT). Файловая система NTFS допускает использование длинных имен, имеет высокие характеристики производительности и

надежности, средства разграничения доступа, мощные средства поиска файлов в каталогах, основанные на использовании *B*-деревьев, и не требует дефрагментации диска. Использование *B*-деревьев при поиске имен в каталогах вместо обычного последовательного перебора значительно сокращает время открытия файла, особенно, если каталог содержит очень много файлов (сотни файлов). Файловая система NTFS содержит специальные средства, предназначенные для повышения устойчивости к различным аварийным ситуациям (например, внезапное отключение питания или аварийный останов ОС). Эти средства используют логику обработки транзакций, в результате чего операции, которые не успели завершиться, будут отменены с восстановлением исходного состояния файловой системы. Файловая система NTFS практически не имеет ограничений на максимальный размер файла. Файловая система NTFS позволяет выполнять динамическую компрессию файлов.

3.1.6.3. Применение Windows NT в АСУ ТП

Применение Windows NT в АСУ ТП обусловлено в значительной степени появлением ряда программных продуктов, которые позволяют использовать ее в качестве платформы для создания приложений в системах реального времени, а также во встраиваемых конфигурациях. Наиболее известными расширениями реального времени для Windows NT являются продукты компаний VenturCom, Nematron, RadiSys. Решения фирмы VenturCom стали стандартом для создания приложений жесткого реального времени на платформе Windows NT. При разработке интерфейса для приложений реального времени разработчики пошли по пути модификации модуля Windows NT - слоя аппаратных абстракций (HAL - Hardware Abstraction Layer), отвечающего за выработку высокоприоритетных системных прерываний, мешающих задаче осуществлять управление в жестком реальном времени. Программный продукт Component Integrator компании VenturCom является средством ускоренной разработки и внедрения приложений реального времени для Windows NT. Он поставляется в виде интегрированного пакета, состоящего из инструментов для создания встраиваемых приложений (ECK - Embedded Component Kit) и собственно расширений реального времени (RTX 4.1), позволяющих приложениям, создаваемым для работы под Windows NT, работать в режиме реального времени.

Компания RadiSys применила другой подход к разработке расширений реального времени. Windows NT загружается как низкоприоритетная задача под хорошо проверенной и известной более 20 лет операционной системой реального времени iRMX. Все функции обработки и управления реального времени выполняются как высокоприоритетные задачи под iRMX, изолированные в памяти от приложений и драйверов Windows NT механизмом защиты процессора. Данный подход имеет то преимущество по сравнению с решением VenturCom, что задача реального времени не зависит от работы Windows NT: в случае сбоя или катастрофической системной ошибки в работе Windows NT управляющая задача реального времени будет продолжать работать. Это решение позволяет информировать основную задачу о проблемах, возникших в работе NT, и оставлять только за ней право продолжения работы или останова всей системы.

3.1.7 Операционная система LINUX

LINUX - это ОС для IBM-совместимых компьютеров на базе процессоров I386 и более поздних моделей. Она выполняет многие из функций DOS и WINDOWS [45]. LINUX является бесплатной версией широко распространенной многозадачной ОС UNIX. UNIX в течение десятилетий используется в разнообразных вычислительных системах, начиная от мини-ЭВМ и заканчивая мультипроцессорными и многомашинными системами с производительностью в несколько Гигафлопс. В настоящее время существует несколько версий системы UNIX, которые соответствуют стандарту POSIX. LINUX разработана в соответствии с нормами POSIX.

Операционная система LINUX разрабатывалась десятками программистов, общавшихся между собою через Интернет. LINUX не является собственностью каких-либо фирм и организаций. Разработка осуществляется чаще всего по инициативе пользователей LINUX, решивших реализовать совместный проект. После завершения разработки созданное программное обеспечение отправляется в Интернет на ftp-узел, откуда его может переписать любой пользователь LINUX.

LINUX можно разделить на 4 основных компонента: ядро, SHELL, файловую структуру и утилиты. Ядро - это базовая программа, которая управляет выполнением других программ и аппаратными средствами, например, дисковыми и принтерами. Интерпретатор команд SHELL обеспечивает функционирование интерфейса пользователя. Он принимает от пользователя команды и посылает их в ядро для использования. Файловая структура представляет собой систему хранения файлов на диске. Файлы организованы в каталоги. Любой каталог может содержать произвольное число подкаталогов, в каждом из которых хранятся файлы.

В совокупности ядро, Shell и файловая структура образуют базовую структуру операционной системы. Кроме того, в состав LINUX входят вспомогательные программы или утилиты, являющиеся для системы стандартными. Shell - это интерфейс, обеспечивающий взаимодействие между ядром и пользователем. Он является интерпретатором команд, вводимых пользователем. В качестве альтернативы командной строки в LINUX применяется графический пользовательский интерфейс (GUI) X WINDOW. В этом интерфейсе имеется несколько программ управления окнами, т.е. менеджеров окон. Менеджер окон работает почти так же, как в системе WINDOWS. Он позволяет пользователю управлять с помощью мыши окнами, пиктограммами и меню. Менеджер окон является препроцессором для Shell. Он просто пропускает принимаемые команды в Shell, который их интерпретирует и посылает в ядро. В LINUX имеется несколько менеджеров файлов и программ. За последние годы разработано несколько разновидностей Shell.

3.1.7.1. Файловая структура

В ОС LINUX файлы организованы в каталоги, также как в DOS. Каталоги организованы в иерархическую древовидную структуру, началом которой является корневой каталог. Все остальные каталоги являются подкаталогами этого каталога. В LINUX предусмотрено назначение для каталогов и файлов прав доступа, за счет чего можно разрешить доступ к ним одним пользователям и запретить другим. С одними и теми же файлами могут работать несколько пользователей.

Корневой каталог - это особый каталог, который используется при первоначальной настройке системы. Файлы самой ОС размещаются в специальных каталогах, являющихся, подкаталогами корневого каталога. Иногда их называют системными каталогами. К системным каталогам относятся каталоги: **man**, **bin**, **usr**. Существует и другие системные каталоги. Для того чтобы получить доступ к файлам, находящимся на каком-либо устройстве, необходимо присоединить его файловую систему к определенному каталогу. Эта операция называется монтированием файловой системы. Для каждого файла и каталога в ОС LINUX задаются права доступа, которые определяют, кто и какие операции может производить над данным файлом. Эти права позволяют ограничить доступ к файлу одним из трех способов: 1 - можно предоставить доступ только самому себе; 2 - можно предоставить доступ определенной группе и 3 - доступ может быть предоставлен всем пользователям системы.

Для файла или каталога может быть установлено право на чтение, запись и выполнение. При создании файла автоматически устанавливаются разрешения на чтение и запись для владельца. Существуют три категории пользователей, которые могут иметь доступ к файлу или каталогу: «владелец», «группа» и «прочие». Владелец - это пользователь, создавший файл. Группа - определенный круг пользователей. Прочие - остальные пользователи системы. Для каждой категории пользователей существует отдельный набор прав доступа на чтение, запись и

выполнение. Эти три набора прав доступа для трех категорий образуют в совокупности девять типов разрешений на действия с файлом.

Утилиты. LINUX содержит большое количество утилит. Основные категории утилит: редакторы, фильтры и коммуникационные программы. Существует несколько стандартных редакторов, присутствующих во всех версиях UNIX, в том числе и в LINUX: Ed, Ex, Vi, Emacs. Ed и Ex - это строковые редакторы, а Vi и Emacs - экранные. Утилиты - фильтры считывают входную информацию, поступившую от пользователя, либо из файла, либо из какого-нибудь иного источника, изучают, обрабатывают и затем выдают результат. Существует несколько разных типов фильтров. Одни обеспечивают вывод на экран отредактированные версии файла с помощью команд построчного редактирования. Другие ищут в файлах заданный образец и выводят только часть данных, содержащих этот образец. Третьи выполняют операции подготовки текстов, выявляя в файле команды форматирования и выводя его отформатированную версию. Фильтры можно соединять между собой, подавая выходные данные одного фильтра на вход другого, и т.д.

В LINUX имеется набор утилит, которые позволяют связываться с другими пользователями системы и других систем; реализована система электронной почты; имеется набор инструментальных средств, предназначенных для работы в ИНТЕРНЕТ. Эта сеть создавалась и развивалась на основе UNIX- систем. Существует сетевая операционная система Network Desktop Linux фирмы Caldera, которая свободно стыкуется с серверами Novell и другими сетевыми серверами и позволяет связывать LINUX- системы в локальные сети.

3.1.7.2. Принципы работы с ОС LINUX

Для каждого пользователя LINUX создает виртуальный компьютер, который, по сути, является индивидуальным пользовательским интерфейсом. Эти пользовательские интерфейсы еще называют бюджетами. ОС LINUX можно рассматривать как систему, работающую на трех уровнях, причем каждый уровень работает поверх другого. При включении компьютера ОС загружается и начинает работать. Однако взаимодействовать с ней нельзя до момента регистрации. После запуска LINUX выдает приглашение зарегистрироваться. Для регистрации нужно ввести пользовательское имя и пароль. Процесс регистрации и использования возможностей LINUX можно считать следующим уровнем. После первоначальной регистрации взаимодействие пользователя с системой осуществляется через простой интерфейс командной строки. Для того чтобы система выполняла те или иные действия, нужно вводить команды и нажимать клавишу ENTER. На этом уровне разрешается использование всевозможных утилит и программ, например, редакторов, компиляторов. Третьим уровнем работы системы является взаимодействие с помощью графического интерфейса пользователя X WINDOW. Запуск X WINDOW осуществляется командой "startx". Менеджер окон X WINDOW работает поверх интерфейса командной строки.

Диалоговое руководство **man** позволяет получить информацию о любой команде и программе ОС LINUX. Все обычные файлы и файлы устройств, а также входная и выходная информация команд в ОС имеют одинаковую структуру - байтовый поток. Входные данные для команды направляются в поток данных, называемый стандартным вводом, а вся выходная информация направляется в поток данных, называемый стандартным выводом. Стандартные ввод и вывод имеют такую же структуру, как и файлы, поэтому они свободно сопрягаются между собой. С помощью операторов переадресации стандартные ввод и вывод можно переадресовывать из файла в файл. Выходные данные одной команды можно использовать как входные для другой. В одной командной строке можно связывать несколько команд, формируя тем самым конвейер.

С помощью редактора можно создавать файлы, содержащие команды **Shell** и определения переменных. Такие файлы образуют сценарии. В сценарии могут встречаться переменные, которые используют аргументы, вводимые в командной строке. Установив для файла сценария право на выполнение, можно трактовать его имя так, как будто это обычная команда.

Операционная система Linux может использоваться как операционная система общего назначения. Однако это вовсе не ограничивает область применения данной системы, так как существуют надстройки для данной системы, позволяющие использовать ее в системах реального времени (RT-Linux, Kurt, UTIME). RT-Linux обеспечивает жесткое реальное время, тогда как Kurt - мягкое. ОС Linux позволяет одновременно исполнять достаточное количество задач, ограничивающееся только объемом памяти и временными соотношениями между ними, предоставляет широкий набор IPC - сигналы, сообщения, семафоры, разделяемая память, ввод/вывод с отображением на память, fifo - файлы, сетевые гнезда (сокеты).

К достоинствам данной системы, по отношению к ОС 5000, можно отнести следующие:

- ОС работает в защищенном режиме процессора Intel и использует все его возможности;
- для пользовательского процесса предоставляется до 3 GB виртуальной памяти;
- ОС и все программы работают каждый в своем виртуальном адресном пространстве и, таким образом, не имеют возможности несанкционированно или при ошибке изменять код или данные другого процесса, или ОС;
- ОС Linux предоставляет унифицированный интерфейс для сетевых взаимодействий, благодаря чему Linux становится все более популярным для сетевых технологий;
- ОС Linux соответствует стандарту POSIX (Portable Operating System Interface - Интерфейс переносимых операционных систем), а также совместима на уровне исходных кодов с UNIX SVR4 и BSD 4.4 - что предоставляет возможность быстрого переноса программного обеспечения под другие программные (совместимые с UNIX и POSIX) и аппаратные платформы (Alpha и Sparc);
- для системного программирования используется язык программирования C, что дает преимущества использования языка высокого уровня. Для Linux разработан широкий набор инструментальных средств для системного программирования, которые бесплатно поставляются с системой;
- ОС Linux широко известна и динамично развивается. Множество программ переносятся сейчас под данную платформу, что позволяет использовать ОС не только на нижнем уровне, в контроллерах, но и на верхнем, на пультах операторов, для систем отображения;
- свободно распространяются исходные тексты ядра системы, что позволяют модифицировать систему в соответствии с предъявляемыми требованиями с помощью специального конфигулятора для настройки ядра и предоставляемых ядром сервисов;
- надстройка RT-Linux обеспечивает жесткое реальное время, со временем реакции 40 μ s на Intel 80386 процессоре, что является необходимым для создания систем реального времени в управлении технологическими процессами;
- поддержка многопоточного программирования (threads), что может повысить скорость реакции системы;
- система Linux поддерживает реализацию виртуальной памяти, что еще больше расширяет возможности ее использования. Однако при необходимости можно отменить использование данного механизма при помощи специальных утилит настройки системы;
- существует большой набор программных инструментальных средств, позволяющий создавать не только сами приложения, но и графическую оболочку. Для этих целей существует ряд коммерческих Motif и некоммерческих Lesstif и Mootif библиотек, а так же специальный язык и библиотека к нему, являющиеся широко используемым инструментальным средством создания приложений для оконной системы X Windows System - tcl/tk.

Недостатком Linux, как встроенной системы, является сравнительно большие требования к процессору (Intel 80386 и выше) и памяти (4 MB), хотя этот недостаток себя исчерпал, т.к. уже широко используются контроллеры с Intel 80486 и Intel Pentium процессорами и памятью 8 – 16 MB.

Операционная система Linux находит все более широкое применение, как операционная система общего назначения, так и как операционная система для встраиваемых устройств - контроллеров. Система Linux является наиболее динамично развивающейся системой, для которой каждый день разрабатываются или переносятся с других программных платформ все

новые технологии, драйвера и подсистемы. Все расширения реального времени для операционной системы Linux являются бесплатными, что создает большой интерес для производителей такого рода систем.

3.2 SCADA - системы

Развитие информационных технологий, повышение степени автоматизации и перераспределение функций между человеком и аппаратурой обострило проблему взаимодействия человека-оператора с системой управления.

Расследование и анализ большинства аварий и происшествий в авиации, наземном и водном транспорте, промышленности и энергетике, часть из которых привела к катастрофическим последствиям, показали, что, если в 60-х годах ошибка человека являлась первоначальной причиной лишь 20 % инцидентов (80 %, соответственно, за технологическими неисправностями и отказами), то в 90-х годах доля «человеческого фактора» возросла до 80 %. Причем, в связи с постоянным совершенствованием технологий и повышением надежности электронного оборудования и машин, доля эта может еще возрасти. Основной причиной таких тенденций является старый традиционный подход к построению сложных АСУ ТП, при котором применяются новейшие технические достижения с целью повышения степени автоматизации и, в то же время, недооценивается необходимость построения эффективного человеко-машинного интерфейса (HMI - Human-Machine Interface), т.е. интерфейса, ориентированного на пользователя (оператора).

Изучение материалов по проблемам построения эффективных и надежных ПТК для АСУ ТП показало необходимость применения нового подхода при разработке - «human-centered design» (или top-down, сверху-вниз), т.е. ориентация в первую очередь на человека-оператора (диспетчера) и его задачи, вместо традиционного применявшегося «hardware-centered» (или bottom-up, снизу-вверх), в котором при построении ПТК основное внимание уделялось выбору и разработке технических средств. Применение нового подхода подтвердили его эффективность, позволив увеличить производительность операторов, на порядок уменьшить процедурные ошибки и свести практически к нулю некорректируемые ошибки операторов.

В широком смысле SCADA (Supervisory Control And Data Acquisition System) - это процесс сбора информации в реальном времени с удаленных точек (объектов), ее обработка, анализ и возможное управление удаленными объектами. В широком смысле понятие «SCADA-система» включает следующие структурные компоненты (рис. 3.5): удаленный терминал; диспетчерский пункт управления (главный терминал); коммуникационная система (каналы связи); специализированное программное обеспечение.

SCADA-системы функционируют в «мягком реальном времени». Под мягким реальным временем понимается в данном случае периодичность контроля порядка долей секунды (в отличие от «жесткого реального времени» с периодом контроля до миллисекунд и быстрее, выполняемого аппаратными средствами, и гарантированным временем реакции на любое событие в системе). Участие оператора в процессе управления предполагает, что допустимы достаточно длительные задержки (порядка секунды и более). Особенности процесса управления на основе SCADA - системы:

- процесс SCADA применяется в системах, в которых обязательно наличие человека (оператора, диспетчера), неправильное воздействие которого может привести к отказу (потере) объекта управления или даже катастрофическим последствиям;
- оператор несет, как правило, общую ответственность за управление системой, которая, при нормальных условиях, только изредка требует подстройки параметров для достижения оптимальной производительности;
- активное участие оператора в процессе управления происходит нечасто и в непредсказуемые моменты времени, обычно в случае наступления критических событий (отказы, нештатные ситуации и пр.);

- действия оператора в критических ситуациях могут быть жестко ограничены по времени (несколькими минутами или даже секундами).

Прообразом современных систем SCADA на ранних стадиях развития автоматизированных систем управления являлись системы телеметрии и сигнализации.

Удаленным терминалом могут быть как примитивные датчики, осуществляющие съем информации с объекта, так и специализированные многопроцессорные отказоустойчивые вычислительные комплексы, осуществляющие обработку информации и управление в режиме жесткого реального времени. Конкретная его реализация определяется конкретным применением. Использование устройств обработки информации на нижнем уровне позволяет снизить требования к пропускной способности каналов связи с центральным диспетчерским пунктом.

Диспетчерский пункт управления (главный терминал) осуществляет обработку данных и управление высокого уровня, как правило, в режиме мягкого реального времени. Одна из его основных функций - обеспечение интерфейса между человеком-оператором и системой (HMI, MMI). В зависимости от конкретной системы главный терминал может быть реализован в самом разнообразном виде - от одиночного компьютера с дополнительными устройствами подключения к каналам связи до больших вычислительных систем (мэйнфреймов) и/или объединенных в локальную сеть рабочих станций и серверов.

Коммуникационная система необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на удаленный терминал (или удаленный объект - в зависимости от конкретного исполнения системы).

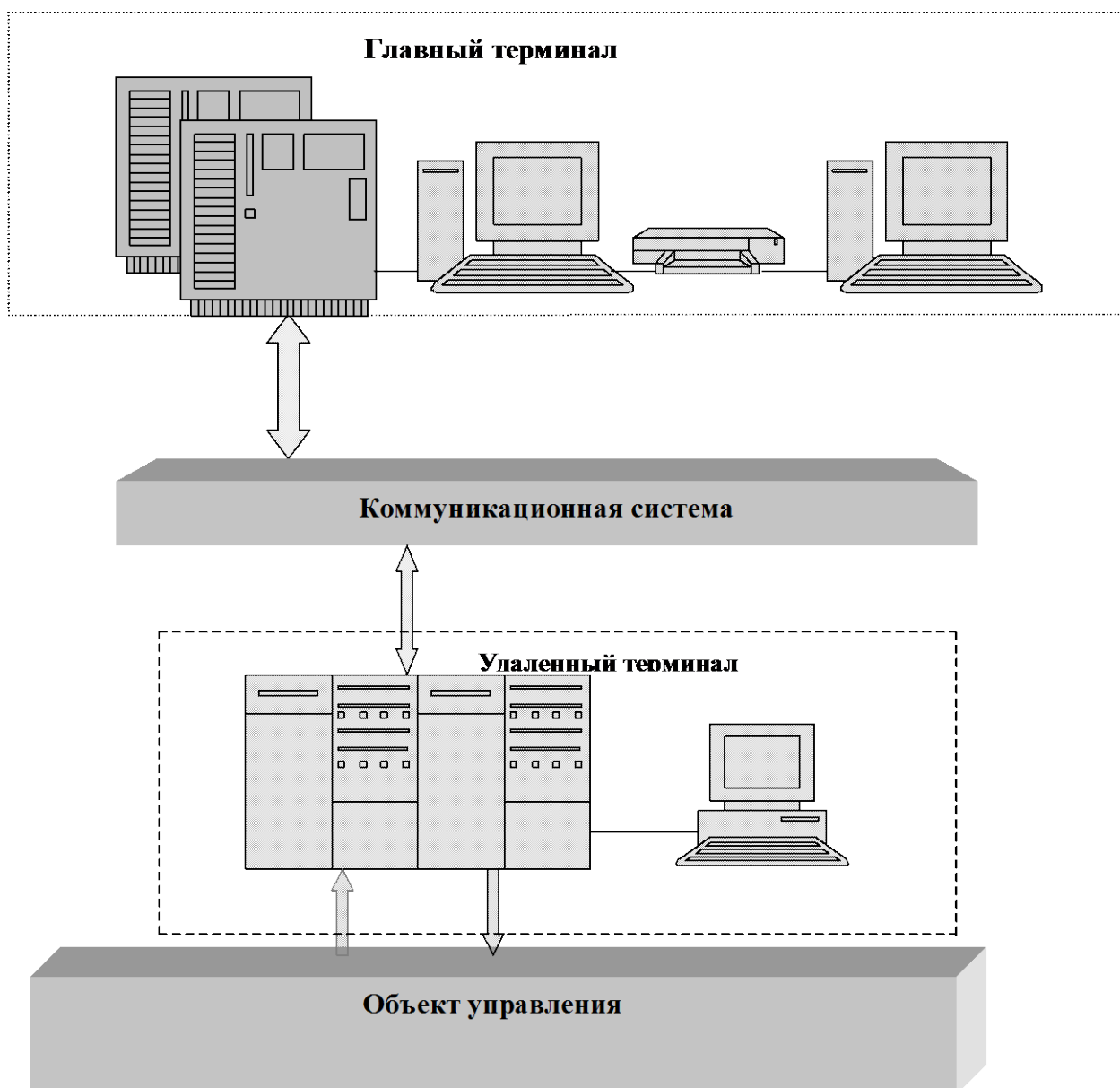


Рис.3.5. Основные структурные компоненты SCADA-системы.

Специализированное программное обеспечение - пакет программ, предназначенный для отображения хода технологического процесса в виде мнемосхем, сигнализации об аварийных ситуациях, ведения системного журнала, обеспечения общего управления процессом со стороны оператора и т. п.

В узком смысле в понятие «SCADA-система» (или «SCADA-пакет») включают только последний компонент: специализированное программное обеспечение, ориентированное на визуализацию технологических процессов и коммуникацию с внешним миром. В дальнейшем будет рассматриваться именно это понятие. На верхнем уровне современных АСУ ТП [46] существует четыре основных функциональных компонента - человек-оператор, компьютер взаимодействия с человеком, компьютер взаимодействия с задачей (объектом), задача (объект управления). Основные функции человека-оператора: планирует, какие следующие действия необходимо выполнить; обучает (программирует) компьютерную систему на последующие действия; отслеживает результаты автоматической работы системы; вмешивается в процесс в случае критических событий, когда автоматика не может справиться, либо при необходимости подстройки (регулировки) параметров процесса; обучается в процессе работы (получает опыт).

Данное представление явилось основой для разработки современных методологий построения эффективных SCADA-систем. Основными функциями SCADA-систем являются

мониторинг процессов, диспетчерское управление, сигнализация, документирование процесса и архивирование данных. Современный мониторинг предполагает применение наглядного дизайна в виде, максимально приближенном к реальному. При этом облегчается восприятие информации, снижается утомляемость и ослабляются требования к квалификации диспетчерского персонала. При отображении используются графическое многооконное представление объектов с применением изменения цвета и движения, параметры отображаются в различных формах (в том числе и в виде графиков).

SCADA-системы функционируют на цеховом уровне автоматизации, связанном, прежде всего, с получением и визуализацией информации от программируемых контроллеров, распределенных систем управления. Поставляемая на данный уровень информация, как правило, недоступна верхнему уровню - системе управления производством. Основные возможности и средства, присущие всем SCADA-системам и различающиеся только техническими особенностями реализации, а именно: автоматизированная разработка, позволяющая создавать ПО системы автоматизации без реального программирования; средства сбора первичной информации от устройств нижнего уровня; средства управления и регистрации сигналов об аварийных ситуациях; средства хранения информации с возможностью ее последующей обработки; средства обработки первичной информации; возможность работы прикладной системы с наборами параметров, рассматриваемых как единое целое. Системы класса SCADA обычно содержат следующие средства:

- база данных (БД) параметров и диалоговый редактор БД;
- средства сбора и отображения данных по любому параметру БД с учетом предыстории;
- графический редактор статических изображений, динамических изображений и символов, позволяющий создавать библиотеки типовых подпрограмм, используемых графическими редакторами статических и динамических изображений;
- генератор отчетов, использующий данные реального времени и предыстории;
- средства отображения и обработки событий реального времени (сигнализация на мнемосхемах, тренды динамики и тренды предыстории, сводки событий и тревог, звуковая и речевая сигнализация);
- парольная защита и средства разграничения доступа по уровням прав операторов.

Перечисленные средства присутствуют во всех системах класса SCADA, хотя могут называться по-другому. В целом технология проектирования системы автоматизации на основе SCADA-систем заключается в следующем:

- разработке архитектуры всей системы автоматизации (на этом этапе определяется функциональное назначение каждого узла системы автоматизации);
- решении вопросов, связанных с возможной поддержкой распределенной архитектуры, необходимостью введения узлов с горячим резервированием и т.п.;
- создании прикладной системы управления для каждого узла, где специалист в области автоматизируемых процессов наполняет узлы архитектуры алгоритмами, совокупность которых позволяет решить задачи автоматизации;
- приведении параметров прикладной системы в соответствие с информацией, поступающей от датчиков в устройства нижнего уровня;
- отладке созданной прикладной программы в режиме эмуляции и реальном режиме.

Перечисленные возможности SCADA-систем в значительной мере определяют стоимость и сроки создания ПО, а также сроки ее окупаемости. Подавляющее большинство SCADA-систем реализовано на MS Windows-платформах. Главные причины этого - быстрое развитие OPC-технологий (OLE for Process Control), низкие цены аппаратного обеспечения, распространенность Windows NT на офисных рынках, хорошие технические характеристики.

Одной из основных черт современных систем автоматизации является высокая степень интеграции. В любой из них могут быть задействованы объекты управления, исполнительные механизмы, аппаратура, регистрирующая и обрабатывающая информацию, рабочие места операторов, серверы БД и т.п. Очевидно, что для эффективного функционирования в этой разнородной среде SCADA-система должна обеспечивать высокий уровень сетевого сервиса.

Необходимо, чтобы она поддерживала работу в стандартных сетевых средах (Arcnet, Ethernet и т.д.) с использованием стандартных протоколов (Netbios, TCP/IP и др.), а также обеспечивала поддержку наиболее популярных сетевых стандартов из класса промышленных интерфейсов (Profibus, Canbus, LON, Modbus и т.д.).

Главной тенденцией развития SCADA-систем является их реализация на основе архитектуры «клиент-сервер», состоящей из четырех функциональных компонентов:

1. User (Operator) Interface (интерфейс пользователя-оператора) - важная составляющая систем SCADA, для которой характерны: стандартизация интерфейса пользователя вокруг нескольких платформ; все более возрастающее влияние Windows NT; использование стандартного графического интерфейса пользователя (GUT); технологии объектно-ориентированного программирования: DDE, OLE, Active X, OPC, DCOM; стандартные средства разработки приложений, среди которых наиболее популярные - Visual Basic for Applications (VBA), Visual C++; появление коммерческих вариантов программного обеспечения класса SCADA/MMI для широкого спектра задач. Объектная независимость позволяет интерфейсу пользователя представлять виртуальные объекты, созданные другими системами. Результатом является расширение возможностей по оптимизации HMI-интерфейса.

2. Data Management (управление данными) - отход от узкоспециализированных баз данных в сторону поддержки большинства корпоративных реляционных баз данных (Microsoft SQL, Oracle). Функции управления данными и генерации отчетов осуществляются стандартными средствами SQL, 4GL. Эта независимость данных изолирует функции доступа и управления данными от целевых задач SCADA, что позволяет легко разрабатывать дополнительные приложения по анализу и управлению данными.

3. Networking Services (сети и службы) – переход к использованию стандартных сетевых технологий и протоколов. Службы сетевого управления, защиты и управления доступом, мониторинга транзакций, передачи почтовых сообщений, сканирования доступных ресурсов (процессов) могут выполняться независимо от кода целевой программы SCADA.

4. Real-Time Services (службы реального времени). Данные службы представляют собой быстродействующие процессы, которые управляют обменом информацией с главным терминалом и SCADA-процессами, осуществляют управление резидентной частью базы данных, оповещение о событиях, выполняют передачу информации о событиях на интерфейс пользователя (оператора).

Некоторые фирмы разрабатывают тесно интегрированные со SCADA-пакетами системы управления производством и обеспечивают обмен между этими уровнями, тем самым заметно расширяя сервисные возможности своих продуктов для реализации комплексного подхода к автоматизации промышленного предприятия в целом. Исходя из требований, которые предъявляются к SCADA-системам, спектр их функциональных возможностей определен и реализован практически во всех известных пакетах. Известны такие SCADA-системы, как Genesis (Iconics), FIX (Intellution), InTouch (WonderWare), Trace Mode (AdAstra), Simplicity (GE Fanuc Automation) [46] и др. Практически все SCADA-системы предлагают дружелюбный интерфейс для разработки приложений, ориентированный не на профессионалов-программистов, а на технологов. Поэтому приложение в SCADA-системе может быть реализовано без реального программирования. Предусмотрены формализованные средства сбора и хранения первичной информации от устройств нижнего уровня, средства хранения информации с возможностью ее пост-обработки (как правило, реализуется через интерфейсы к наиболее популярным базам данных), средства обработки первичной информации.

Основу большинства SCADA-систем составляют несколько программных компонентов (база данных реального времени, ввода/вывода, предыстории, аварийных ситуаций) и администраторов (доступа, управления, сообщений). Технология проектирования систем автоматизации на основе различных SCADA-систем во многом схожа с проектированием приложений в системах управления реального времени. Возможны два подхода к проектированию:

1. Разрабатываются независимые приложения для каждого узла, а затем создается коммуникационное программное обеспечение для распределенного приложения в целом;

2. Собственно распределенное приложение состоит из множества реализованных на каждом узле частей, а коммуникация их в единое приложение осуществляется автоматически.

Графические интерфейсы SCADA-пакетов создаются с помощью графических объектно-ориентированных редакторов с набором анимационных функций. Векторная графика дает возможность осуществлять широкий набор операций над выбранным объектом, а также быстро обновлять изображение на экране. Большинство SCADA-пакетов являются открытыми, т.е. для них описаны форматы данных и интерфейсы для подключения независимо разработанных компонентов, таких как: собственно программные модули; драйверы ввода-вывода; компоненты, реализованные в соответствии с OPC, ActiveX.

3.3 Системы технологического программирования

Системы технологического программирования обеспечивают реализацию всех задач автоматического регулирования, логического управления и защит, а также задач обработки информации путем представления их в виде блок-схем пошагового логического управления или структурных схем преобразования информации. Современные системы технологического программирования фактически являются CASE-системами (Computer Aided Software Engineering). CASE-система ориентирована на разработку программ управления технологическими объектами и процессами. В основном используется на нижнем уровне управления – в программируемых контроллерах. Системы технологического программирования имеют модульную иерархическую структуру, позволяющую описывать и манипулировать объектами различной степени сложности.

Для реализации программ логического управления предусматриваются унифицированные средства: управления программами; реализации шаговых программ; реализации логического и других преобразований. Библиотека стандартных алгоритмических модулей включает в себя следующие группы алгоритмов:

- ПИД (П, ПИ) законов регулирования аналогового и импульсного типов различных модификаций;
- алгебраических и других преобразований;
- логических преобразований и связанных с ними преобразований;
- стандартных алгоритмов управления приводами, механизмами, регуляторами, группами оборудования;
- первичной и статистической обработки информации.

Средства технологического программирования должны быть доступны не только программисту, а технологу или специалисту КИП, хорошо знающим специфику объектов управления и технологических процессов. Для их описания обычно используют языки релейно-контактных схем, функциональных блоков и другие, теоретические основы которых взяты из методов автоматического управления. Накопленный опыт был специфицирован Международной Электротехнической Комиссией в виде стандарта IEC-1131-3. Стандарт определяет 5 языков программирования для систем автоматизации технологических процессов: **Sequential Function Chart (SFC)** - язык последовательных функциональных блоков; **Function Block Diagram (FBD)** - язык функциональных блоковых диаграмм; **Ladder Diagrams (LD)** - язык релейных диаграмм; **Structured Text (ST)** - язык структурированного текста, **Instruction List (IL)** - язык инструкций. Использование стандартных средств и языков программирования позволяет существенно снизить затраты на разработку прикладного программного обеспечения.

Язык SFC находится над всеми остальными языками и дает возможность описать логику программы на уровне чередующихся функциональных блоков и условных переходов, но предполагает конечную реализацию программы на других языках. Функциональные блоки представляют собой действия, которые должны быть исполнены, в том числе и параллельно, а

условные переходы задают условия, которые необходимо выполнять для перехода к следующему функциональному блоку.

Язык LD используется для описания различных логических выражений и реализует такие элементы, как открытый и закрытый контакты. Кроме этого, реализованы функции и функциональные блоки.

Язык FBD позволяет строить комплексную процедуру, состоящую из различных функциональных блоков.

Язык ST относится к классу языков высокого уровня и по мнемонике похож на PASCAL. Он служит для создания процедур со сложной логикой.

Язык IL принадлежит к классу языков низкого уровня и позволяет создавать высокоэффективные функции. Этот язык имеет смысл применять для написания наиболее критических мест программы.

Использование данного стандарта соответствует концепции открытых систем, а именно: делает программу для контроллера не критичной к конкретному оборудованию. Программа в стандарте IEC не зависит от типа процессора, ОС и блоков ввода/вывода. В настоящее время десятки фирм обеспечивают поддержку этого стандарта. Наиболее известны следующие реализации этого стандарта: пакет ISaGRAF (CJ International, Франция), InConirol (Wonderware, США), Paradym 31 (Intellution, США), Ultralogic (Ultralogic, г. Москва). Эти пакеты имеют открытую архитектуру систем контроля в реальном времени, предназначены для разработки, проверки и исполнения программ управления дискретными и непрерывными процессами. Разработка приложений в последних версиях этих продуктов стала носить принципиально новый характер. Раньше разработка представляла собой алгоритмизацию работы для каждого узла с реализацией коммуникационных связей, обеспечивающих согласованное функционирование узлов для решения некоторой задачи. Сейчас в рамках инструментальной системы описываются отдельные узлы как ресурсы с правилами их функционирования по определенному алгоритму по отношению друг к другу. Реализация распределенного приложения встраивается в сам процесс его разработки в рамках систем управления в реальном времени. В продуктах различных фирм распределенность встраивается в процедуру разработки приложения по-разному. В системах, ориентированных на Windows NT, таких как **InControl**, используются механизмы COM/DCOM, в многоплатформных системах типа **ISaGRAF Pro** - чаще всего, протокол TCP/IP. Более подробно остановимся на реализации указанных систем на Windows NT/NTE. Последние версии этих продуктов предлагают более широкие возможности и лучшую производительность благодаря поддержке новых технологий, перечисленных ниже.

ActiveX-объекты типа органов управления для ПИД-регулирования, управления интегрированным движением и с нечеткой логикой. Можно создавать свои собственные алгоритмы на языке C++ и обращаться к ним из CASE-систем типа InControl как к ActiveX-объектам. Объекты типа *Fuzzy Logic* (с нечеткой логикой) позволяют повысить точность и стабильность сложных процессов.

Распределенное управление на базе COM. Механизм взаимодействия между исполняющими подсистемами вышеуказанных компонентов, ответственных за управление контроллерным оборудованием и процессами, основан на технологии COM/DCOM.

Новые редакторы. Можно использовать для создания своей собственной системы управления самые различные графические и текстовые языки программирования. Поддерживается прямое взаимодействие с самыми разными устройствами ввода/вывода, двигателями, датчиками и прочим заводским оборудованием, а также все широко распространенные и новейшие открытые интерфейсы с устройствами. Следует отметить ту особенность, что многие распределенные приложения должны быть интегрированы в уже существующие на производстве сетевые инфраструктуры. Требование специфического сетевого протокола повлекло бы обновление всех уже существующих компонентов. Таким образом, пакеты комплексной автоматизации, использующие DCOM-технологию, индифферентны к сетевым протоколам и поэтому позволяют с небольшими затратами наращивать существующие системы.

При разработке распределенных приложений, реализованных на различных платформах (клиент - на одной, сервер - на другой) у разработчиков, естественно, возникают трудности, связанные с различной философией пользовательских интерфейсов, различными системными службами и даже набором имеющихся сетевых протоколов. Один из популярных подходов в решении этой проблемы состоит в использовании только самых общих свойств, присущих этим платформам, в том числе сетевых протоколов. Но он препятствует использованию развитого сервиса определенной ОС, характеризуется бедной визуальной интеграцией, скромными средствами просмотра состояния приложения в целом. DCOM открыт для всех подходов. Эта модель не препятствует использованию зависящих от ОС служб или оптимизаций, позволяет интегрировать компоненты на различных платформах в одно распределенное приложение. В целом, использование CASE-средств заменяет различные самостоятельные разработки и характеризуется более дешевой архитектурой управления с интегрированными возможностями по подключению, мощными средствами обработки данных и простотой расширения.

3.3.1 Система ULTRALOGIC

Система ULTRALOGIC предназначена для разработки программ промышленных контроллеров с помощью простых инструментальных средств с использованием в качестве языка программирования язык функциональных блок-диаграмм [47]. «Простота» в данном случае достигается применением методов объектного визуального программирования. При этом исключаются «ошибки пальца», широко распространенные в лингвистических языках программирования, опасные операторы циклов, проблемы с захватом памяти и т.п. Система максимально ориентирована на то, чтобы инженер в области автоматизации работал с понятным ему технологическим контроллером, а не компьютером с его сложной системой команд, памятью и прерываниями. ULTRALOGIC рассматривает контроллер как «черный ящик», связанный с объектом управления посредством формальных устройств аналогового и дискретного ввода/вывода. Подобный подход вовсе не предполагает тривиальности системы.

ULTRALOGIC имеет механизм вызова внешних процедур, написанных на других языках, таких как АССЕМБЛЕР, С, PASCAL. Ориентированная на IBM PC совместимые контроллеры, система ULTRALOGIC фактически является независимой по отношению к аппаратной платформе целевого контроллера. Для этого в системе есть специальный инвариантный компилятор, который использует подготовленную ранее информацию об аппаратной платформе контроллера и его конфигурации. Информация готовится пользователем в диалоговом режиме. Система предоставляет пользователю возможность создания и накопления собственных функциональных блоков. Этот механизм является мощным инструментом для облегчения разработки и улучшения читабельности программы, когда один или несколько функциональных блоков полностью описывают управление тем или иным технологическим процессом. ULTRALOGIC представляет собой интегрированный комплекс программ в операционной среде DOS или Windows и включает в себя графические средства, компиляторы, средства интерактивного диалога, настройки и отладки проектов. Он функционирует на IBM PC совместимом компьютере, с помощью которого может производиться и отладка программы на объекте.

С помощью ULTRALOGIC были реализованы проекты по автоматизации инженерных сооружений больших зданий, процессов технологических испытаний ряда объектов, научного эксперимента.

3.4 Современные технологии для разработки базового ПО

Практически на любом уровне ПТК возможно применение технологий, предложенных фирмой Microsoft - COM, объекты ActiveX, OPC.

Технология COM (Component Object Model). Для систем управления, работающих на платформе Windows, фирма Microsoft предложила архитектуру компонентных объектов. Для ее реализации в Windows предусмотрен код, который облегчает СОМ-программирование.

Традиционное приложение состояло из отдельных файлов, модулей или классов, которые компилировались и компоновались вместе. Разработка приложений из компонентов - так называемых приложений компонентной архитектуры - происходит иначе. Компонент подобен мини-приложению. Он поставляется пользователю как двоичный код, скомпилированный, компонованный и готовый к использованию. Модификация или расширение приложения сводится к замене одного из составляющих его компонентов новой версией. Один из наиболее многообещающих аспектов компонентной архитектуры - это быстрая разработка и развитие приложений. Из накапливаемого набора компонентов в библиотеках можно будет выбирать, как из деталей конструктора, требуемые цельные приложения.

Распределенные компоненты. С возрастанием производительности и общего значения сетей потребность в приложениях, распределенных по различным узлам сети, увеличивается. Компонентная архитектура позволяет упростить процесс разработки подобных распределенных приложений. Шагом в сторону компонентной архитектуры являются приложения «клиент-сервер», разделенные на 2 части: клиентскую и серверную.

Создать из обычного приложения распределенное легче, если оно состоит из компонентов. Во-первых, оно уже разделено на функциональные части, которые могут размещаться на расстоянии друг от друга. Во-вторых, поскольку компоненты заменяемы, вместо некоторого компонента можно подставить другой, единственная задача которого - обеспечение связи с удаленным компонентом. Так, если некоторые компоненты А и В переносятся с локальной машины на удаленные, то на локальной машине вместо компонентов А и В появляются переадресовщики, перенаправляющие запросы к соответствующим компонентам по сети. При наличии подходящих переадресующих компонентов приложение может совершенно игнорировать фактическое местоположение своих частей. Основное преимущество применения компонентов - их способность подключаться к приложению и отключаться от него. Для этого компоненты должны удовлетворять двум требованиям. Во-первых, они должны компоноваться динамически, а во-вторых, - должны скрывать (или инкапсулировать) детали своей реализации. Программа или компонент, использующий другой компонент, называется *клиентом* (client). Клиент подключается к компоненту через *интерфейс*. Если компонент изменяется без изменения интерфейса, то изменений в клиенте не требуется. Аналогично, если клиент изменяется без изменения интерфейса, то нет необходимости изменять компонент. Чем надежнее интерфейс изолирован от реализации, тем менее вероятно, что он изменится при модификации клиента или компонента. Таким образом, СОМ - это спецификация, указывающая, как создавать динамически взаимозаменяемые компоненты. СОМ определяет правила, по которым должны создаваться компоненты и клиенты, чтобы гарантировать возможность совместной их работы [48]. Компоненты СОМ состоят из исполняемого кода, распространяемого в виде динамически компоновываемых библиотек (DLL) или EXE-файлов Win32. Но сама по себе динамическая компоновка не обеспечивает компонентной архитектуры. Компоненты СОМ объявляют о своем присутствии стандартным способом. Используя схему объявлений СОМ, клиенты могут динамически находить нужные компоненты. Интерфейс СОМ состоит из набора функций, которые реализуются компонентами и используются клиентами. Интерфейсом в СОМ является определенная структура в памяти, содержащая массив указателей на функции.

Методы межпроцессной коммуникации. Существуют такие методы межпроцессной коммуникации как: DDE, именованные каналы, разделяемая память. Однако СОМ использует локальный вызов процедуры (local procedure call, LPC). LPC - это средство связи между разными процессами на одной и той же машине, построенное на основе RPC (remote procedure call) - удаленного вызова процедур. С помощью разнообразных сетевых протоколов RPC обеспечивает коммуникацию между процессами на разных машинах. Распределенная модель СОМ (DCOM - Distributed COM) использует RPC для связи по сети. LPC работает, используя

вызовы операционной системы. ОС известны физические адреса, соответствующие логическому адресному пространству каждого процесса. Следовательно, ОС может вызывать функцию внутри любого процесса. Но вызвать соответствующую функцию еще недостаточно. Необходимо передать параметры функции из адресного пространства клиента в адресное пространство компонента. Этот процесс называется маршалингом (от marshaling; marshal - располагать, размещать или устанавливать в определенном порядке). Если оба процесса находятся на одной машине, то данные одного процесса копируются в адресное пространство другого. Если процессы находятся на разных машинах, то данные необходимо преобразовать в стандартный формат, учитывающий межмашинные различия (например, порядок следования байтов в слове).

Клиент, которому необходимо взаимодействовать с компонентом в другом процессе, не может обращаться непосредственно к этому компоненту. Для этого существует форма межпроцессного взаимодействия, поддерживаемого ОС. COM обеспечивает эту коммуникацию в прозрачном режиме. Он перехватывает вызовы от клиента и направляет их в компонент другого процесса. Стандарт RPC определен фондом OSF (OPEN Software Foundation) в спецификации DCE (Distributed Computing Environment) RPC. Когда клиент и компонент находятся на разных машинах, DCOM заменяет локальную межпроцессную коммуникацию сетевым протоколом. Ни клиент, ни компонент «не знают» о сети. Динамические (run time) COM-библиотеки обеспечивают объектно-ориентированный сервис клиенту и компоненту и используют RPC и Security provider, чтобы генерировать стандартные сетевые пакеты, которые соответствуют DCOM - сетевым протокольным стандартам.

Приложения компонентной архитектуры не ограничиваются более рамками одной машины. Причем прозрачный механизм расширения приложений компонентной архитектуры распространяется на Internet. DCOM встраивается в ОС Windows NT 4.0. Приложение, компоненты которого взаимодействуют по принципу «n-tier» - равный с равным - ответственно за управление технологическим процессом в реальном времени, визуализацию и выборку информации. Вышеуказанные операции реализуются на разных машинах, соединенных в сеть, причем в узлах размещаются исполняющие системы, ответственные за взаимодействие с контроллерами, за визуализацию информации, за регистрацию данных, за систему управления ресурсами предприятия.

ActiveX-объекты. Одна из реализаций интерфейсов COM/DCOM - создание управляющих компонентов ActiveX. Для разработки приложений и SCADA-систем на языках Visual Basic, C, C++, Java. могут использоваться управляющие элементы ActiveX. Технология ActiveX предназначена для ускорения разработки приложений путем компоновки их из модулей, разработанных на разных языках. Три основных понятия, определяющие этот несложный интерфейс, - Properties (данные), Methods (функции) и Events (события). Наличие высокоуровневых программных средств для разработки профессионалами-программистами и простые возможности встраивания ActiveX-объектов в разрабатываемые приложения делают технологию ActiveX популярной.

3.4.1 OPC-стандарт

До недавнего времени в качестве механизма разделения данных между прикладными системами и устройствами типа программируемых контроллеров применялся протокол DDE. Применение технологии OLE/DCOM в разработке программных стандартов обмена информацией между устройствами привело к появлению спецификаций OPC (OLE for Process Control). Спецификация OPC описывает объекты OPC COM и их *интерфейсы*, реализованные в OPC-серверах. OPC является наиболее общим способом организации взаимодействия между различными источниками данных, такими как устройства, базы данных, SCADA-системы и т.д. [49-51]. OPC обеспечивает интерфейс между приложениями-клиентами и серверами путем реализации стандартного механизма связи между источниками данных (серверами) и

получателями данных (клиентами). OPC-клиенты могут связываться с одним или несколькими OPC-серверами, разработанными разными производителями.

Основная цель стандарта OPC заключается в определении механизма доступа к данным с любого устройства из приложений и, в частности, обеспечении совместной работы и взаимозаменяемости (совместимости) промышленных устройств разных поставщиков. OPC позволяет производителям оборудования поставлять программные компоненты, которые **стандартным способом** обеспечат клиентов данными программируемых контроллеров. Имея утвержденный в стандарте набор интерфейсов, конечный пользователь сможет организовать взаимодействие и обмен данными между любыми распределенными компонентами системы. Применение OPC-стандарта обеспечивает следующие преимущества:

- позволяет определять на уровне объектов различные системы управления и контроля, работающие в распределенной неоднородной среде;

- устраняет необходимость использования различного нестандартного оборудования и соответствующих коммуникационных программных драйверов.

Широкому распространению OPC-компонентов способствует наличие инструментальных средств, которые позволяют экономить время разработки, поскольку нет необходимости вникать в детали OPC-стандарта и не требуют отслеживать различия в версиях стандарта (по сравнению с предыдущим).

OPC обеспечивает интерфейс между приложениями-клиентами и серверами путем реализации стандартного механизма связи между источниками данных (серверами) и получателями данных (клиентами). Иными словами, OPC является аналогом технологии Plug-n-Play для программного обеспечения, выполняющего функции человеко-машинного интерфейса в сфере промышленной автоматизации. В традиционной идеологии клиент-сервер различные серверы или устройства имеют различные интерфейсы или драйверы для каждого приложения-клиента. Поскольку аппаратные средства разных производителей имеют различные и притом фиксированные протоколы обмена, архитектура приложений-клиентов также является уникальной в каждом конкретном случае. Это приводит к увеличению времени разработки и стоимости АСУ ТП, а любое изменение, внесенное производителем в устройство или протокол обмена, требует внесения изменений в функционирующую систему. Архитектура клиент-сервер, основанная на технологии OPC, позволяет решить данную проблему. В данном случае устройство каждого производителя имеет единственный стандартный драйвер, совместимый с OPC (OPC-сервер). Приложения, соответствующие спецификации, выработанной для клиента OPC (OPC-клиенты), могут при этом обмениваться данными с устройствами любого производителя при наличии OPC-совместимых драйверов для указанных устройств.

Приложения, соответствующие спецификации клиента OPC, исполняющиеся на рабочих станциях, которые объединены в локальную вычислительную сеть, могут иметь доступ к каналам ввода/вывода аппаратуры, обслуживаемой серверами OPC, которые исполняются на любых узлах сети. Одно из уникальных качеств, присущих данной технологии, состоит в том, что клиенты OPC имеют возможность получения данных от удаленных серверов OPC через глобальную сеть Интернет.

Описание технологии OPC. OPC-технология основана на модели распределенных компонентных объектов Microsoft DCOM и устанавливает требования к классам объектов доступа к данным и их специализированным интерфейсам для использования разработчиками клиентских и серверных приложений. Структура взаимодействия между приложениями-клиентами и OPC-серверами различных производителей показана на рис. 3.6. OPC-сервер состоит из следующих основных объектов: сервер (server), группа (group), и элемент (item). Объект “OPC-сервер” содержит информацию о себе и является контейнером для объектов “OPC-группа”.

Объект “OPC-группа” содержит информацию о себе и обеспечивает механизм управления и логической организации объектов “OPC-элемент” при обращении к ним со стороны клиентов. Например, группа может представлять элементы, подлежащие отображению на экране монитора или представления в отчете. При этом могут выполняться операции записи и

чтения данных. Кроме того, между клиентом и элементами, входящими в группу, может быть организован обмен данными, основанный на механизме исключений, когда создание объектов типа “ОПС-элемент” в группе выполняется только при необходимости передачи данных между клиентом и элементами группы. ОПС-клиент также может установить для различных групп элементов различные условия, при выполнении которых сервер будет извещать клиентов об изменениях связанных с элементами данных. Каждая группа содержит один или более элементов. Взаимосвязь групп и элементов показана на рис. 3.7.

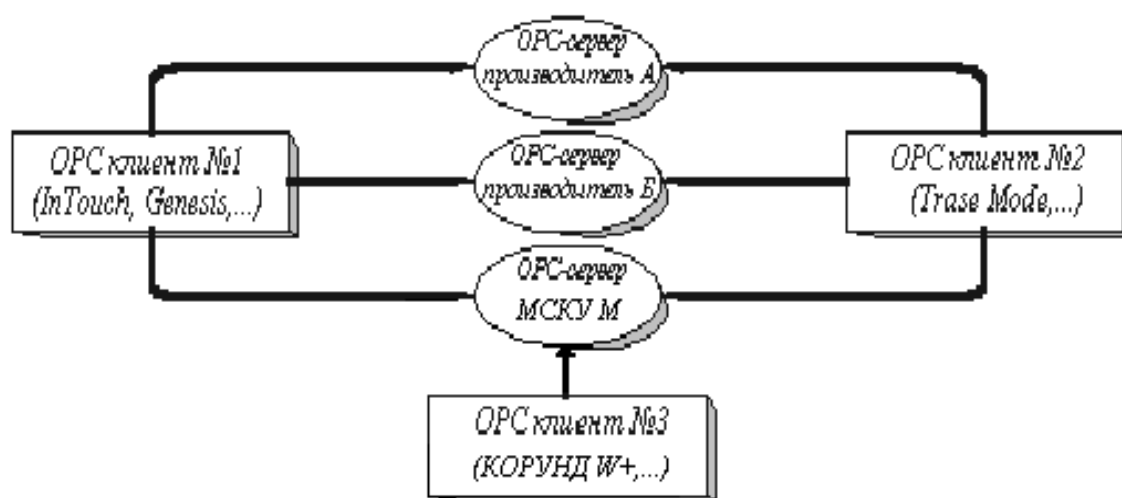


Рис 3.6 Структура взаимодействия между приложениями-клиентами и ОПС-серверами.

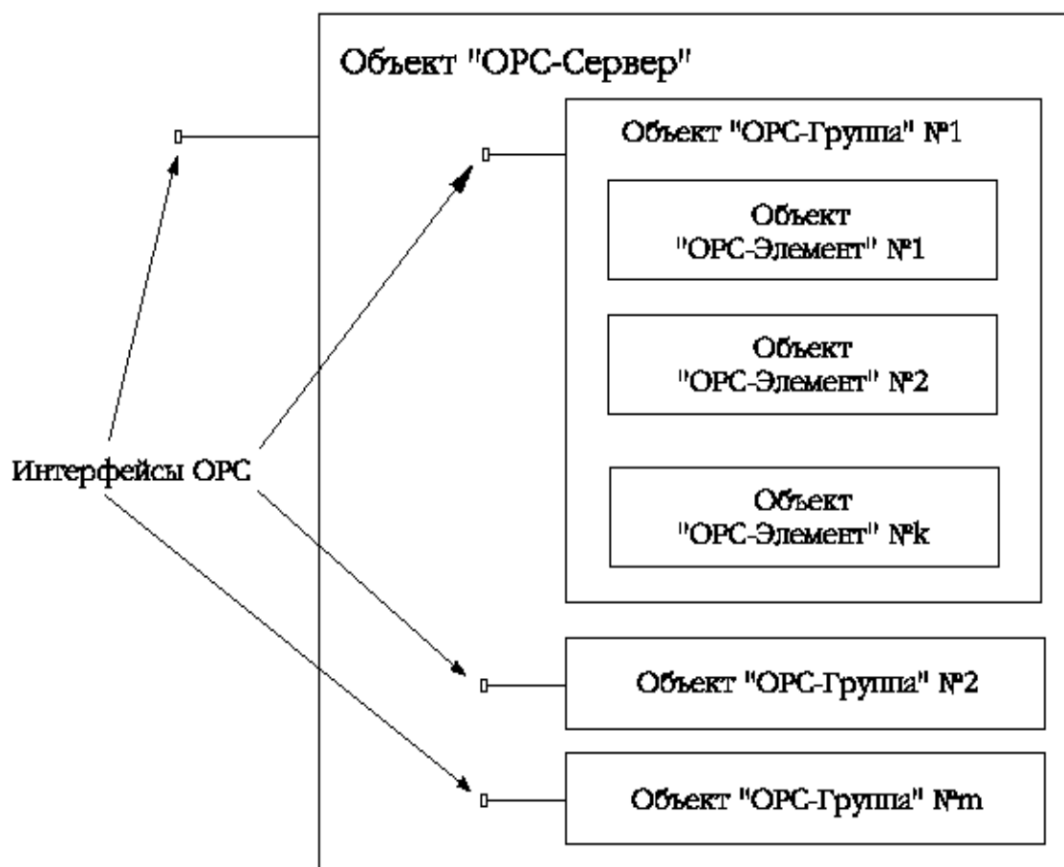


Рис 3.7. Структура взаимосвязи групп и элементов.

Элемент представляет логическое соединение с источниками данных внутри сервера. ОПС-элемент, с точки зрения специализированного интерфейса, не доступен в качестве объекта ОПС-клиенту. Таким образом, для ОПС-элемента, не определено никаких внешних интерфейсов. Любая операция доступа к ОПС-элементам со стороны клиента осуществляется через объект “ОПС-группа”, который включает в себя элементы (является контейнером для ОПС-элементов). Иными словами, доступ к объектам “ОПС-элемент” осуществляется через интерфейсы объектов, в которых они определены. С каждым ОПС-элементом связаны следующие атрибуты: Значение (Value), Качество сигнала или Признак достоверности (Quality) и Метка времени (Time Stamp). Значение представляется в форме VARIANT. Признак достоверности представляет состояние текущего значения, связанного с ОПС-элементом. Последние не являются собственно источниками данных, а представляют только логические связи или ссылки на реальные источники данных. ОПС-элементы должны рассматриваться в качестве адресов памяти, через которые ОПС-клиенты получают доступ к физическим источникам данных.

ОПС-сервер является программным модулем, который содержит код, реализующий методы специализированного настраиваемого интерфейса ОПС, обеспечивающие приложениям-клиентам доступ к данным аппаратуры (устройств ввода/вывода), которые применяются в системах промышленной автоматизации. Следует обратить внимание на тот факт, что понятие “Объект ОПС-сервер” и “Приложение ”Сервер ОПС” не являются равнозначными. Объект класса ”ОПС-сервер” создается при запуске приложения ”Сервер ОПС”. Серверы могут быть внутрizaдачными (in-process), локальными и удаленными.

Внутрizaдачный сервер создается в адресном пространстве приложения-клиента. Код, реализующий методы внутрizaдачного сервера, размещается в библиотеке динамической

компоновки (DLL). Доступ к данным внутризадачного сервера является наиболее быстрым, поскольку вызов его методов сводится к простому вызову функций DLL, однако внутризадачный сервер может быть доступен только одному приложению-клиенту и притом на одной физической машине.

Локальный сервер создается в отдельном процессе, исполняющемся на той же физической машине, что и приложение-клиент. Код, реализующий методы локального сервера, размещается в исполняемом файле (EXE). Локальный сервер позволяет нескольким приложениям-клиентам, исполняющимся на одной физической машине, получать доступ к обслуживаемым указанным сервером источникам данных.

Удаленный сервер создается на физической машине, находящейся в одной сети (локальной или глобальной) с физической машиной, на которой исполняется приложение-клиент. Код, реализующий методы удаленного сервера, размещается в исполняемом файле (EXE). Удаленный сервер позволяет нескольким приложениям-клиентам, исполняющимся на разных физических машинах, подключенных к локальной и/или глобальной вычислительной сети, получать доступ к обслуживаемым указанным сервером источникам данных.

Применение современных технологий в процессе генерации конечного приложения все более напоминает не традиционный процесс *разработки* прикладного программного обеспечения, а процесс компонентной *сборки*. Соответственно, качественно меняется характер труда прикладного программиста.

3.5 Программное обеспечение МСКУ М

Функции и структура программного обеспечения МСКУ М ориентированы на определенную методику проектирования и программирования задач управления технологическими процессами на базе МСКУ. В этой методике предусмотрены не только средства сбора и обработки информации от объектов АСУ ТП, но также необходимые программные средства отображения информации, средства взаимодействия между абонентами сети МАПС, средства подготовки и отладки программ, измерительные и тестовые средства и т.д. [52]. Типовая методика проектирования задач управления технологическими объектами, как правило, включает следующие основные этапы.

1. *Постановка задачи создания системы управления*, функционирующей в МСКУ. На этом этапе формулируется задача проектирования; определяются состав и характеристики входных и выходных каналов связи с объектом, а также операции первичной обработки сигналов, получаемых через входные каналы. Определяется состав информации, передаваемой из данного МСКУ другим абонентам системы и из других абонентов системы в данный МСКУ, а также регламент выполнения передач. На основании этих данных осуществляется выбор требуемой операционной среды и генерация соответствующей ОС для МСКУ.

2. *Разработка алгоритмов функционирования задач управления*. На этом этапе, во-первых, определяются состав и типы задач. Каждой задаче присваивается имя и устанавливается регламент ее выполнения. Во-вторых, разрабатываются алгоритмы функционирования задач в терминах технологического процесса. Выполняется разбиение задач на относительно независимые функционально законченные модули, определяются интерфейсы между ними. Если на этом этапе возникает необходимость внесения корректив в планы генерации ОС и (или) в базу данных, эти коррективы вносятся.

3. *Разработка программ*. На этом этапе выполняется разработка программ для всех модулей, определенных на этапе 2. При этом может использоваться любой из вошедших в средства технологического программирования языков.

4. *Автономная отладка программ*. На этом этапе выполняется автономная отладка всех разработанных программ, для которых она применима. Инструментальными средствами выполнения автономной отладки являются соответствующие отладчики, функционирующие на персональных ЭВМ.

5. *Сборка загрузочного модуля*. На этом этапе выполняется генерация управляющей системы, формирование базы данных МСКУ и сборка всех программ в единый загрузочный модуль, предназначенный для работы в МСКУ.

6. *Комплексная отладка системы*. На этом этапе осуществляется проверка правильности функционирования программ созданной системы в режиме, близком к реальному режиму ее использования. Инструментальными средствами, которыми пользуются в процессе комплексной отладки, являются отладчики, работающие в режиме непосредственной связи с МСКУ, а также программный комплекс инженерной панели МСКУ.

7. *Организация рабочего места оператора-технолога*. На этом этапе осуществляется выбор средств для организации операторской станции, ее настройка, подготовка статической и динамической информации для отображения параметров системы, мнемосхем, таблиц и т.п.

8. *Разработка эксплуатационной документации и документации сопровождения*. На этом этапе выполняется разработка всех текстовых документов, предназначенных для эксплуатационного персонала системы, а также для сопровождения ее в процессе эксплуатации. В частности, все разработанные программы должны сопровождаться текстами, в которые внесены необходимые комментарии. Кроме того, должны быть разработаны спецификации на межпрограммные интерфейсы.

9. *Проведение испытаний и внедрение системы*. Работы, выполняемые на данном этапе, специфичны для каждой системы. Они завершаются получением откорректированной (по результатам испытаний и опытной эксплуатации) документации.

На рис. 3.8 приведена общая структура программного обеспечения микропроцессорной системы контроля и управления нового поколения - МСКУ 2М. Компоненты ПО МСКУ 2М разработаны на платформах операционных систем MS DOS, ОС 5000, QNX, LINUX, WINDOWS. Состав операционных систем МСКУ 2М приведен на рис. 3.9.

3.5.1. Системное ПО МСКУ 2М

Структура системного ПО МСКУ 2 приведена на рис. 3.10.

Управляющая система МСКУ 2 функционирует под управлением управляющей системы (УС МСКУ 2), находящейся на FLASH-диске контроллера МСКУ 2 [53]. УС МСКУ 2 включает: стартовое программное обеспечение контроллера (Стартовое ПО) и исполнительную систему МСКУ 2 (ИС МСКУ 2).

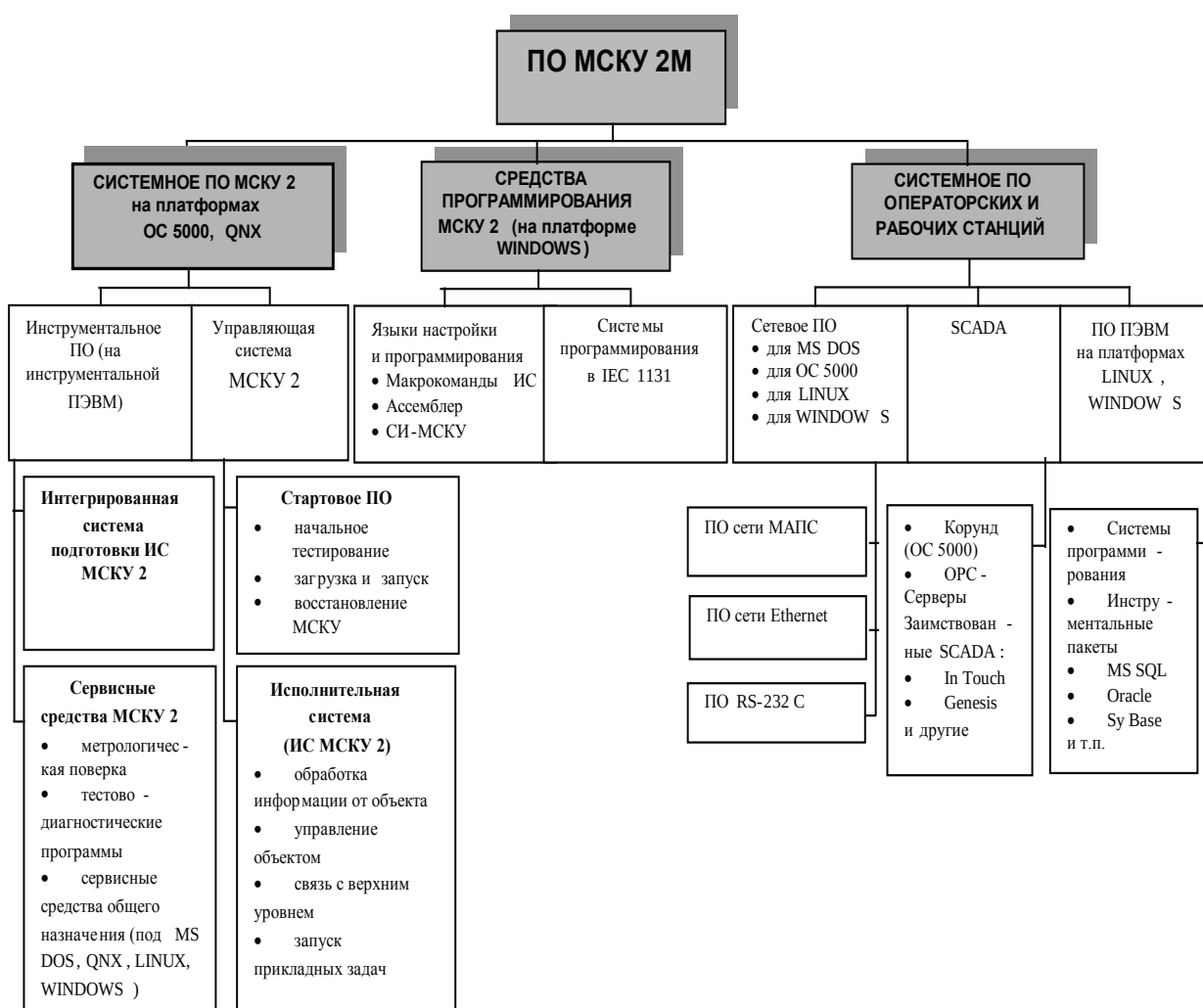


Рис.3.8. Структура программного обеспечения МСКУ 2М.

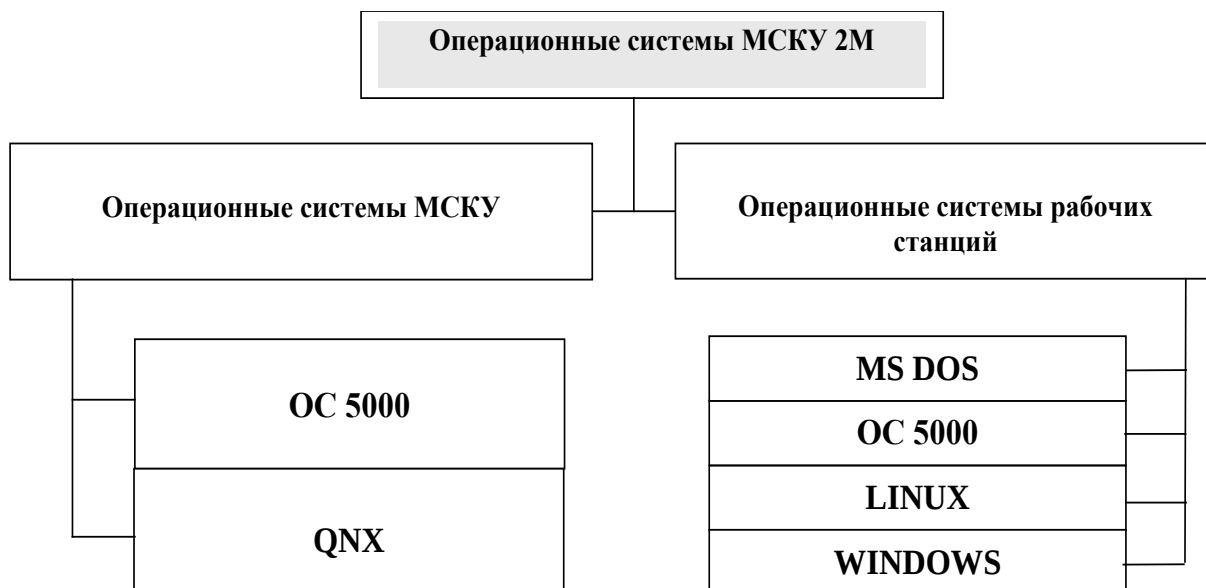


Рис. 3.9 Состав операционных систем МСКУ 2М.



Рис. 3.10. Структура системного ПО МСКУ 2.

Стартовое ПО включает следующие функции: начальное тестирование и запуск контроллера; запуск ИС МСКУ; запуск МСКУ по включению питания; восстановление функций МСКУ (после замены отказавшего контроллера в резервированном МСКУ). Стартовое ПО недоступно пользователю.

Исполнительная система МСКУ 2 (ИС МСКУ 2) – это совокупность управляющих и прикладных программ, организующих работу МСКУ 2 в соответствии с определенными при генерации режимами. ИС создается под каждую конкретную конфигурацию и набор функций МСКУ 2 в виде единого загрузочного модуля путем объединения системных библиотек и задач пользователя. С МСКУ 2 загружается в ОЗУ процессора из FLASH-диска. Состав исполнительной системы МСКУ 2 показан на рис. 3.11.

Функции системных управляющих программ:

- поддержка функционирования как нерезервированных, так и резервированных конфигураций МСКУ;
- обмен информацией между МСКУ и внешними абонентами МСКУ (другими МСКУ, рабочими станциями ПС 5110, ПЭВМ) по локальным сетям или интерфейсам ИРПС, RS-232C; ввод и обработка информации от каналов связи с объектом (аналоговых, дискретных, число-импульсного типа);
- формирование и вывод управляющих воздействий на объект (через формирователи аналоговых, дискретных, и импульсных сигналов);
- автоматическое логическое отключение отказавших сменных блоков, не влияющих на общую работоспособность МСКУ. Автоматическое восстановление функций ИС МСКУ в резервированных МСКУ при замене отказавшего сменного блока на исправный (без выключения МСКУ);
- поддержание единого времени в контроллерах резервированных МСКУ, а также единого времени с другими компонентами системы управления;
- запуск прикладных задач (по временному расписанию, по командам внешних абонентов);
- контроль и защита от несанкционированного доступа к МСКУ; периодический и непрерывный контроль работоспособности технических и программных средств МСКУ в процессе функционирования, а также выполнение тестово-диагностических операций при техническом обслуживании МСКУ и др.

Получаемые ИС результаты обработки помещаются в операционную базу данных (ОБД) и становятся доступными для других компонентов ПО МСКУ 2. Оперативная база данных является главным инструментом взаимодействия между функциональными частями системы управления.

Конфигурация технических средств МСКУ 2 и необходимый состав функций в ИС МСКУ задаются в программе генерации на специальном языке макрокоманд.

Монитор ИС управляет такими процедурами: ввода/вывода; обработки; синхронизации и выравнивания данных; контроля оборудования и задачами пользователя.

Модули ядра операционной системы

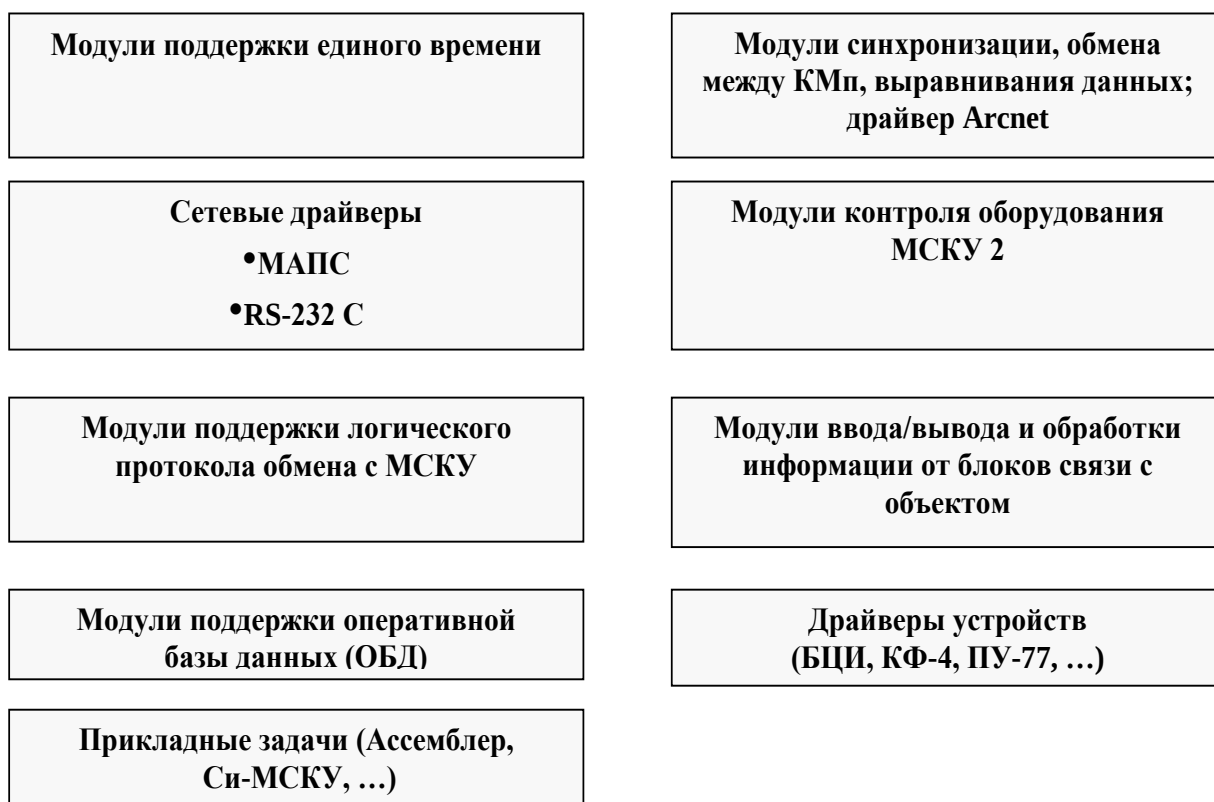


Рис. 3.11. Состав исполнительной системы МСКУ 2.

Монитор запускает процедуры по времени и по событию. К запускаемым по времени относятся процедуры "инициативного" цикла (период цикла от 5 до 50 *ms*), основного цикла (до трех циклов с периодами свыше 50 *ms*) и фоновые. К запускаемым по событию относятся процедуры типа "запрос" (по запросу от других абонентов, от задач пользователя), обработки инициативы от модулей связи с объектом (при наличии изменений состояния дискретных входов), обработки сообщений от устройств, подключенных к МСКУ через модуль последовательной связи МПСв16-1. Основной цикл работы исполнительной системы МСКУ 2 приведен на рис. 3.12.

В цикле любого типа производится запуск процедур в следующей последовательности: синхронизация контроллеров (в троированных МСКУ); ввод и обработка данных от блоков связи с объектом, относящихся к данному циклу; обмен данными между контроллерами и выравнивание данных (в троированных МСКУ); выполнение задач пользователя, относящихся к данному циклу; синхронизация контроллеров (в троированных МСКУ); вывод управляющих воздействий на блоки связи с объектом, относящиеся к данному циклу. Последовательность запуска процедур основного цикла определяется последовательностью описания этих процедур в программе генерации и текущим циклом.

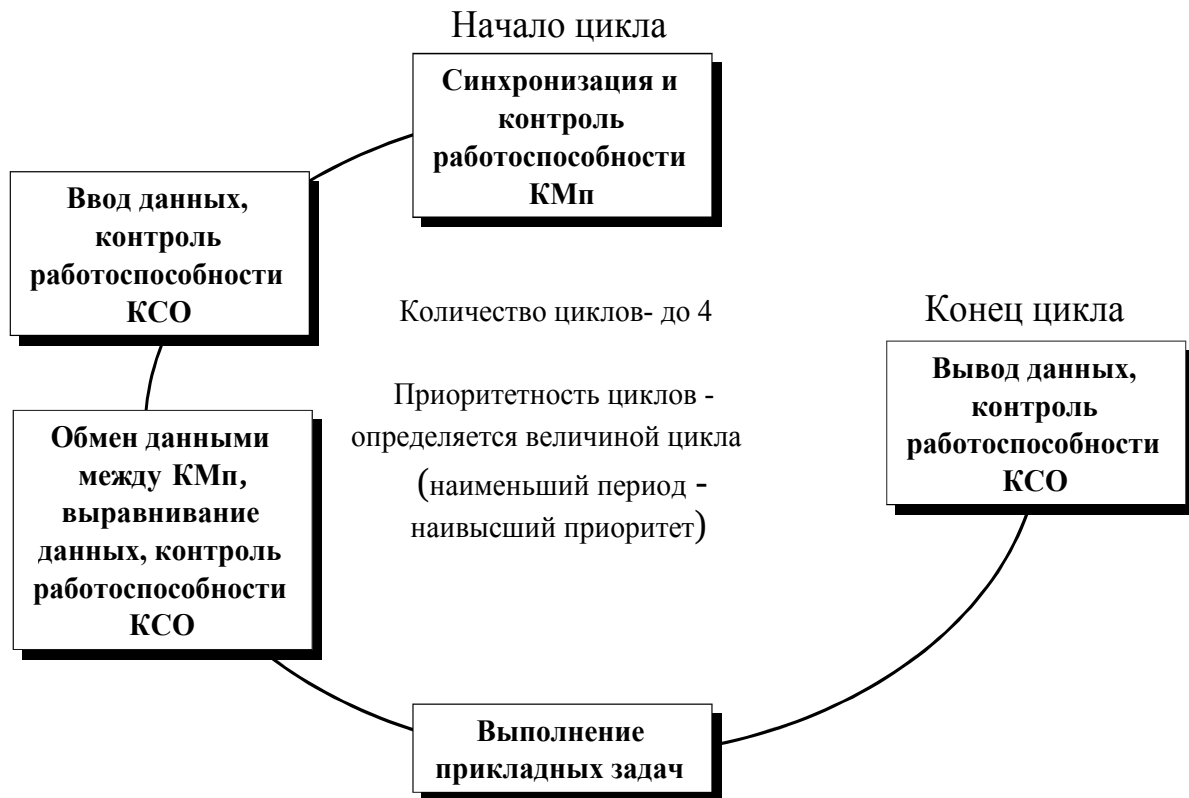


Рис. 3.12. Основной цикл работы исполнительной системы МСКУ 2: КСО – каналы связи с объектом.

Задачи пользователя, запускаемые циклически, могут вызываться в каждом цикле или раз в несколько циклов, что определяется при генерации ИС. Очередная задача пользователя в пределах цикла одного типа запускается на выполнение только после завершения предыдущей. Процедуры "инициативного" цикла запускаются либо в каждом "инициативном" цикле, либо при наличии изменений состояния соответствующих каналов ввода дискретных сигналов, что определяется при генерации исполнительной системы. Процедуры "инициативного цикла" более приоритетны, чем процедуры основного цикла, т.е. прерывают их выполнение. Процедуры ввода/вывода выполняются в каждом цикле, за которым закреплены соответствующие блоки связи с объектом. При выполнении процедур ввода/вывода ИС выполняет обработку следующей информации по заданным алгоритмам для каналов связи с объектом: цифровую коррекцию измеренных значений, фильтрацию, линеаризацию, масштабирование, технологический контроль и др.

Процедуры ввода и вывода, а также задачи пользователя могут выполняться по запросам от других абонентов комплекса (режим "прозрачного ввода-вывода"). Задачи пользователя, запускаемые по запросу от внешних абонентов МСКУ, запускаются на выполнение сразу при поступлении запроса, после завершения выполнения процедур «инициативного» цикла. Аналогичным образом выполняются системные операции по запросам внешних абонентов. Фоновые процедуры выполняются только при наличии резерва времени от всех других типов процедур.

Подсистемы связи с объектом. Каналы функциональных блоков, входящих в состав МСКУ, логически подразделяются на подсистемы, которым присваиваются номера, которые могут изменяться в пределах от 2 до 256. В подсистему могут объединяться до 256 каналов (групп). Адрес канала или группы может изменяться в пределах от 0 до $(n-1) \cdot k$, где n - число каналов (групп) в подсистеме, k - число измеряемых параметров для каждого канала (как правило, $k=1$). Подсистема с номером 2 включает каналы, используемые для контроля оборудования (источники питания, блок вентиляторов и т.д.). Данная подсистема включается в

состав управляющей системы при ее генерации. Если она не включена в управляющую систему, то контроль оборудования МСКУ не выполняется. Состав подсистем от 3 до 256 определяется по усмотрению пользователя.

Объединение каналов в подсистемы обеспечивает возможность организовать необходимый режим ввода/вывода и обработки данных, а также логический уровень адресации каналов ввода/вывода без знания физических адресов и других физических параметров. В одну подсистему объединяются каналы в соответствии со следующими правилами:

- 1) однотипность блоков, которым принадлежат данные каналы;
- 2) единство правил ввода/вывода данных для этих каналов;
- 3) единство алгоритма выравнивания данных, вводимых контроллерами от одних и тех же каналов;
- 4) общность функционального назначения с точки зрения внутренней обработки и обработки в программах пользователя.

Распределение каналов ввода/вывода по подсистемам осуществляется в процессе генерации управляющей системы (создание загрузочного модуля). По типам выполняемых операций и способам их запуска подсистемы подразделяются на следующие типы: подсистемы ввода и обработки в основном, либо в одном из дополнительных циклов управляющей системы; вывода и обработки в основном, либо в одном из дополнительных циклов управляющей системы; ввода инициативных сообщений по прерываниям от блоков; ввода и обработки "инициативных" сигналов; ввода сигналов контроля оборудования в основном, либо в одном из дополнительных циклов управляющей системы; управления вводом данных от ПКИ-2; обработки в основном, либо в одном из дополнительных циклов управляющей системы.

Каналы преобразователей ПКИ-2 включаются в подсистемы следующих типов:

- подсистема управления вводом от преобразователя. Цикличность запуска определяется пользователем, которая обеспечивает ввод последних измеренных значений токовых сигналов в массивы памяти МСКУ, вычисление действительных значений сигналов и запоминание их в памяти как измеренных каналов блока с адресами 41-47;

- подсистема ввода вспомогательных сигналов от преобразователя (каналы ввода фоновой составляющей сигналов, проводимости изоляции и коэффициентов передачи по каналам ввода сигналов). Состав каналов, включаемых в подсистему, способы обработки задаются пользователем;

- подсистема обработки информации. Обработке подвергаются действительные значения токовых сигналов, сформированные в памяти МСКУ подсистемой управления вводом основной информации. Для каналов данной подсистемы дополнительные способы обработки задаются по усмотрению пользователя из числа допустимых для сигналов аналогового типа.

Каналы преобразователей ПКИ-3 включаются в подсистемы ввода информации от термоэлектрических преобразователей и ввода информации от термопреобразователей сопротивления. Количество подсистем, распределение каналов по подсистемам и способы обработки задаются по усмотрению пользователя. Все типы нормализаторов дискретных сигналов включаются в состав подсистем ввода дискретных сигналов и подсистем ввода дискретных "инициативных" сигналов. Для подсистем ввода дискретных "инициативных" сигналов обнаружение изменения события выполняется в каждом "инициативном" цикле, текущие результаты ввода от блоков сравниваются с результатами предыдущего цикла и любое изменение состояния сигналов рассматривается как факт инициативы. Таким образом, будут обнаружены только те изменения сигналов, длительность которых превышает величину "инициативного" цикла.

Подсистема ввода дискретных "инициативных" сигналов может быть настроена на фиксацию переключения сигнала из 0 в 1, либо из 1 в 0, либо на оба переключения. Требуемая настройка указывается в программе генерации. Регистрация изменения состояний осуществляется каждым контроллером МСКУ независимо. Выравнивание результатов в МСКУ не выполняется. Сравнение результатов ввода и их выравнивание возлагается на задачу пользователя, если по факту инициативы необходимо принимать какие-либо действия в МСКУ,

которые могут изменить алгоритм функционирования системы. Если подсистема предназначена только для регистрации процесса переключения, то операция сравнения и выравнивания может быть выполнена на верхнем уровне управления.

Каналы формирователей входят только в подсистемы вывода и обработки. Для формирователей дискретных сигналов допустимы также и операции ввода, которые являются вспомогательными и используются только для контроля их работоспособности по каналам обратной связи.

Блоки сопряжения БСп-14, БСп-15 включаются в подсистему ввода-вывода (с выполнением операций по запросам прикладных программ) и используются для управления цифровыми измерительными приборами: БСп-14 - для управления приборами, выходящими на интерфейс КОП (канал общего пользования); БСп-15 - для управления калибратором ПЗ20 и вольтметром ЦЗ1. Указанные выше приборы используются при проверке метрологических характеристик измерительных каналов ввода/вывода аналоговых сигналов и на время проверки устанавливаются в свободное зарезервированное для них место, а в управляющей системе МСКУ предусматривается подсистема, описывающая эти блоки.

Особенностью подсистем, включающих данные блоки, является то, что они не включаются в циклический режим и операции на них могут быть запущены только по запросу от внешних абонентов МСКУ или программ пользователя текущего МСКУ. Команды управления блоками принимаются только с одного, первого направления ИР и поэтому они доступны только при работоспособности первого контроллера МСКУ.

Учитывая, что алгоритм выполнения операций ввода и вывода на БСп-14 не отличается от аналогичных операций для первого канала БСп-15, управление БСп-14 может осуществляться через подсистему, описывающую БСп-15. Это особенно удобно, когда одно интерфейсное место в монтажном каркасе используется для поочередной установки того и другого блока, как это имеет место при проверке метрологических характеристик каналов ввода и вывода аналоговых сигналов.

Форматы вводимых и выводимых данных. Минимальной адресуемой единицей информации при обмене данными с подсистемами является 16-разрядное слово. Формат представления данных, используемых в операциях "прозрачного" ввода-вывода, определяется типом канала и различен для каналов разного типа. Операции "прозрачного" ввода-вывода предназначены, в основном, для использования в тестово-диагностической системе.

Основными операциями ввода-вывода в МСКУ являются операции циклического ввода-вывода и инициативного ввода, поддерживающие обмен данными между задачами пользователя и каналами связи с объектом. Формат представления данных, используемых в операциях циклического ввода-вывода, может быть сохранен таким же, как в операциях "прозрачного" ввода-вывода, но может быть изменен путем включения процедур обработки (линеаризация и масштабирование данных). Использование процедур обработки обеспечивает приведение представления кодов аналоговых сигналов к единому диапазону от минус 32752 до 32752, что упрощает их обработку в задачах пользователя. Для каналов преобразователей ПКИ-2 операции "прозрачного" ввода-вывода допустимы только для подсистем ввода вспомогательных сигналов. Для подсистем управления вводом основной информации и ввода вспомогательных сигналов от ПКИ-2 формат представления данных в операциях циклического ввода сохраняется таким же, как в операциях "прозрачного" ввода-вывода, а действительные значения сигналов, вычисляемые в процессе ввода и поступающие на обработку в подсистемы обработки сигналов, представляются в диапазоне от минус 32752 до 32752.

Для подсистем ввода аналоговых сигналов среднего и низкого уровня соответствия между диапазонами изменений входного сигнала и пределами выходного кода для каждого блока приведены в табл. 3.2.

Для каналов приема частотных сигналов ПЧ(РТ)-8 текущее значение представляет собой целое число, равное количеству импульсов, накопленных в течение интервала времени преобразования и занимает 12 младших разрядов слова данных (от 0 до 4095) (время преобразования устанавливается перемычкой на блоке). Для каналов приема время-импульсных

сигналов ПВИ(РТ)-8 текущее значение, равное количеству импульсов тактового генератора, накопленных в течение измеряемого промежутка времени (длительности импульса, периода следования импульсов, интервала времени между двумя импульсами) в пределах от 0 до 65535 (режим работы и частота тактового генератора устанавливаются с помощью переключателей на блоке). Для каналов ввода дискретных сигналов текущее значение представляет собой 16-разрядный код, каждый бит которого равен значению двухпозиционного датчика, подключенного к соответствующему входу блока. Первому входу соответствует старший бит слова (15 разряд), второму - 14 разряд слова и т.д. Для блока НД(РК)-28 каналы с адресами 28-31 второго слова являются контрольными. Для каналов формирования аналоговых сигналов текущее выдаваемое значение представляет собой целое число в пределах от 0 до 32752, что соответствует диапазону выходного сигнала. Значащими являются 12 старших разряда кода, 15 разряд - знаковый всегда равен 0. Для каналов формирования дискретных сигналов текущее значение представляет собой 16-разрядный код, каждый бит которого определяет состояние соответствующего выхода: старший бит (15 разряд) - состояние 1-го выхода, 14-й разряд - состояние 2-го выхода и т.д.

Табл. 3.2 Соответствие между входными сигналами и выходными кодами

Тип блока	Пределы изменения входного сигнала	Пределы выходного кода
ПНК(И)-30, ПНК(Р)-8	$\pm 2,5 \text{ V}$; $\pm 5,0 \text{ V}$; $\pm 10 \text{ V}$; $\pm 5 \text{ mA}$; $\pm 0 \text{ mA}$	От минус 32000 до 32000
ПНК(Р)-1	То же	От минус 32000 до 32000
ПТП-4	$\pm 10 \text{ mV}$; $\pm 20 \text{ mV}$; $\pm 30 \text{ mV}$; $\pm 40 \text{ mV}$; $\pm 50 \text{ mV}$; $\pm 80 \text{ mV}$; $\pm 100 \text{ mV}$	От минус 32000 до 32000
ПТП(П)-8	$\pm 10 \text{ mV}$; $\pm 15 \text{ mV}$; $\pm 20 \text{ mV}$; $\pm 30 \text{ mV}$; $\pm 40 \text{ mV}$; $\pm 50 \text{ mV}$; $\pm 80 \text{ mV}$	От минус 32000 до 32000
ПТП(В)-16	$\pm 10 \text{ mV}$; $\pm 20 \text{ mV}$; $\pm 30 \text{ mV}$; $\pm 40 \text{ mV}$; $\pm 50 \text{ mV}$; $\pm 80 \text{ mV}$; $\pm 100 \text{ mV}$	От минус 2000 до 2000
ПТС-4	0-50-100 Ω ; 0-100-200 Ω 0-50-150 Ω ; 0-100-300 Ω 0-50-250 Ω ; 0-100-500 Ω	От минус 32000 до 32000 От минус 16000 до 32000 От минус 8000 до 32000
ПТС(П)-8	25-50-75 Ω ; 0-50-100 Ω 50-100-150 Ω ; 0-100-200 Ω 100-150-200 Ω	От минус 32000 до 32000
ПТС(В)-16	0-10-200 Ω ; 0-50-100 Ω 0-100-200 Ω 0-10-30 Ω ; 0-50-150 Ω 0-100-300 Ω 0-10-50 Ω ; 0-50-250 Ω 0-100-500 Ω	От минус 2000 до 2000 От минус 1000 до 2000 От минус 500 до 2000
ПКИ-3	$\pm 15 \text{ mV}$; $\pm 30 \text{ mV}$ 30-50-70 Ω ; 10-50-90 Ω	От минус 32000 до 32000

Методы сравнения и выравнивания данных. В резервированных МСКУ обеспечивается синхронный режим выполнения программ процессорами разных контроллеров, т.е. выполняется одинаковый набор операций над одинаковыми данными. Для формирования одинакового массива данных, получаемых в результате ввода от функциональных блоков, контроллеры обмениваются результатами ввода, сравнивают между собой с целью контроля правильности ввода и выравнивают с целью формирования одинаковых для задач пользователя.

Методы сравнения и выравнивания данных зависят от типа данных и могут быть: логические коды - одноразрядные двоичные коды, упакованные в 16-разрядные слова; 16-разрядные числовые коды - числа в пределах от 0 до 65535 включительно; 16-разрядные целые числа - числа в диапазоне от минус 32768 до 32767 включительно.

Логические коды сравниваются на полное совпадение. В резервируемых МСКУ, при условии работоспособности всех трех контроллеров и функциональных блоков на ИР, результатом выравнивания является код, разряды которого сформированы по принципу "два из трех".

Числовые коды и целые числа сравниваются либо на точное совпадение, либо на совпадение с точностью до заданной величины, определяющей допустимую величину расхождения между значениями, полученными разными контроллерами. Данные, отбракованные в результате сравнения, в операции выравнивания не участвуют, по соответствующим каналам и направлениям фиксируется ошибка. Выравнивание может быть выполнено по одному из следующих алгоритмов: выбор минимального значения; выбор максимального значения; вычисление среднего арифметического; выбор срединного значения (медианы).

По результатам сравнения и выравнивания данных формируется значение канала, которое запоминается в оперативной базе данных МСКУ.

Посредством установки маски каналы ввода-вывода могут исключаться из режима ввода-вывода. Для каналов с установленной маской операции ввода-вывода, выравнивания и обработки данных не выполняются. Отключение и включение каналов выполняется по командам оператора. Для отключенных каналов в массивах текущих значений сохраняются последние введенные значения.

Инструментальное ПО МСКУ 2 включает интегрированную систему подготовки исполнительной системы МСКУ 2 и сервисные средства МСКУ 2. Инструментальное ПО реализовано на инструментальной ПЭВМ.

Интегрированная система подготовки программ (ИСПП), предназначена для создания конкретных вариантов ИС МСКУ 2 и ее записи на FLASH-диски контроллеров. ИСПП состоит из программного монитора и совокупности программных компонентов, вызываемых монитором по мере необходимости.

Сервисные средства МСКУ 2 представлены на рис. 3.13.

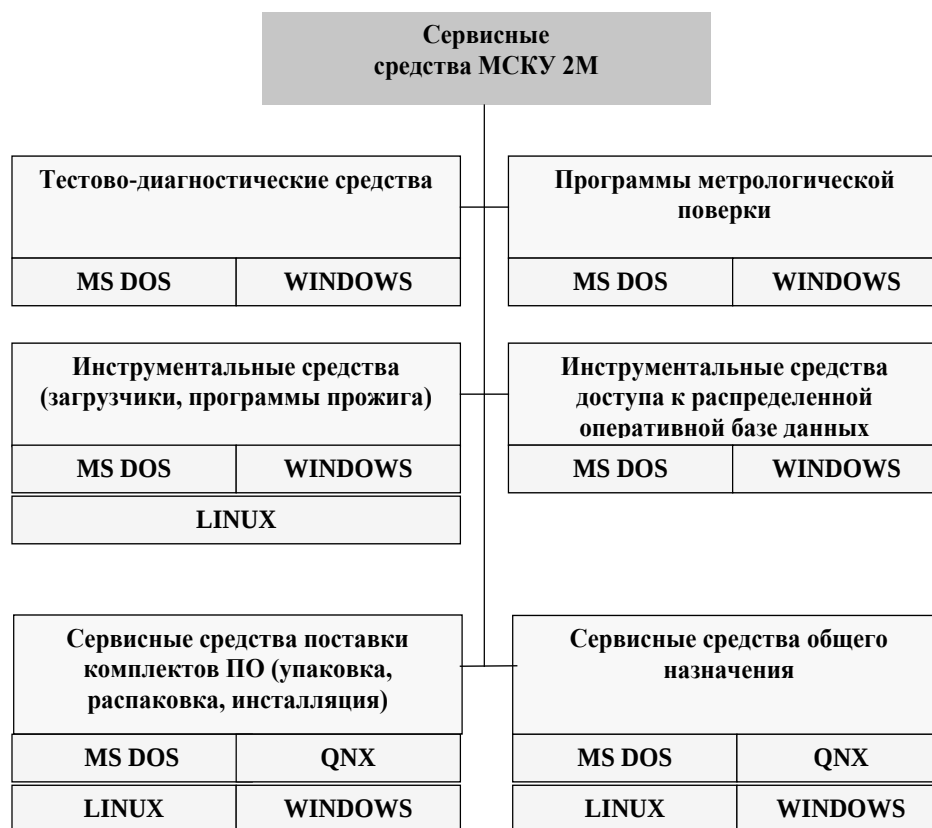


Рис. 3.13 Состав сервисных средств МСКУ.

3.5.2 Средства программирования МСКУ 2

Состав средств программирования МСКУ 2 представлен на рис. 3.14. Система Си-МСКУ включает средства программирования, трансляции, компоновки и отладки программ на языках АСSEMBЛЕР и Си-МСКУ, а также библиотеки поддержки. В основу этих языков программирования положены языки систем Турбо АСSEMBЛЕР и Турбо Си, разработанные фирмой Borland International для IBM PC. Язык АСSEMBЛЕР в основном используется для программирования специфических задач, требующих доступа ко всем ресурсам МСКУ. Отладка программ осуществляется с помощью команд инженерной панели в режиме реального функционирования в соответствии с абсолютными адресами расположения отлаживаемой программы в памяти МСКУ. Основной язык программирования МСКУ - язык Си-МСКУ, являющийся специализированным языком для программирования задач управления технологическими процессами, в котором сохранены все основные возможности языка системы Турбо Си, за исключением библиотек, типа данных double, которые в Си-МСКУ трактуются аналогично типу float, и некоторых других. В то же время в систему Си-МСКУ введены дополнительные функции, ориентированные на программирование задач управления, в том числе и задач шагово-логического управления.

Библиотеки системы Си-МСКУ ориентированы на программирование задач управления технологическими процессами, таких как обработка входных аналоговых, дискретных и импульсных сигналов, регулирование, шагово-логическое управление, формирование и выдача управляющих воздействий на объект, поддержание диалога с оператором через локально подключенные к МСКУ технические средства общения с оператором-технологом, поддержание обмена информацией с внешними абонентами МСКУ через сеть МАПС и др.

Языковые средства, а также встроенные в систему различные препроцессоры позволяют без особого труда связать между собой данные, принадлежащие различным объектам (управляющей системе, задачам, средствам связи с оперативным персоналом, абонентам сети и т.д.) в единую базу данных распределенной системы управления, доступ к которой возможен по именам из различных компонентов комплекса. Отладка программ на языке Си-МСКУ ведется в режиме реального функционирования.

К функциям средств отладки относятся: просмотр текстов программ; выполнение программ до указанного оператора; выполнение Си-программ по операторам АСSEMBЛЕРА; вычисление выражений; чтение/запись локальных и глобальных переменных по идентификаторам, ячеек памяти по адресам; задание/отмена точек останова и точек фиксации; пуск программы в автоматическом режиме и с указанного оператора; прием инициативных сообщений от МСКУ и выдача информации о них на экран и в файл на диске; просмотр файлов, входящих в загрузочный модуль, и т.п.

Интегрированная система подготовки программ. Система Си-МСКУ предоставляет интегрированную среду со специальными инструментальными и технологическими программными средствами, позволяющими быстро и качественно сгенерировать загрузочные модули и запустить их в работу. Эти средства, используя "дружественный" интерфейс с пользователем, обеспечивают создание базы данных, описывающих конфигурации аппаратуры, на основании которых генерируются необходимые управляющие программы, операционные системы, программы пользователя и пакеты. Наличие элементов САПР в этих средствах делает общение человека простым и удобным и, главное, не требует специальной подготовки и высокой квалификации. Последовательно задавая параметры всей сети МАПС, а затем номенклатуру устройств связи с объектом, подключаемых к каждому абоненту, пользователь создает требуемые файлы конфигурации, на базе которых генерируются управляющие системы МСКУ. При генерации управляющих систем пользователю предоставляется возможность методом выбора из списка предлагаемых услуг подобрать для себя необходимые этапы сбора и обработки информации АСУ ТП, включить штатные алгоритмы управления и определить требования для подсистем отображения на рабочих и операторских станциях.

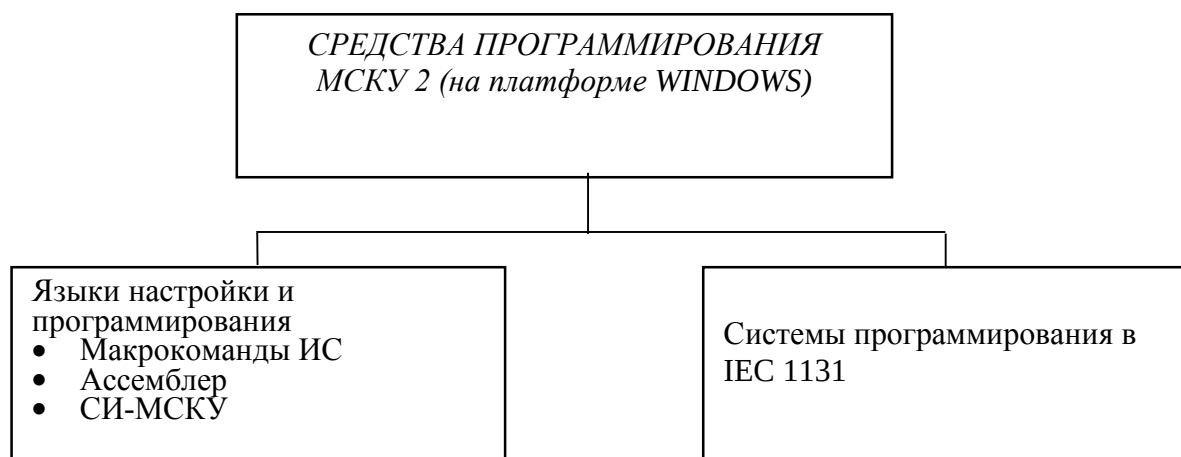


Рис. 3.14. Состав средств программирования МСКУ 2.

Подготовленные загрузочные модули с помощью сервисных технологических программ загружаются в выбранные для этих целей МСКУ, проверяются на правильность функционирования и объявляются работоспособными для ввода в эксплуатацию.

3.5.3 Системное ПО операторских и рабочих станций

Структуры системного и сетевого ПО операторских и рабочих станций приведены на рис. 3.15 и 3.16.

SCADA-система МСКУ М КОПУНД [54] и заимствованные SCADA могут использоваться в рабочих станциях, а также в IBM-совместимых ПЭВМ как на верхнем уровне управления, обеспечивая доступ ко всем МСКУ в системе, так и на локальном уровне обеспечивая доступ к одному. Основным их назначением является выполнение функций сбора, обработки и отображения технологической информации, ведения архива нарушений технологического процесса и архива (журнала) изменений значений параметров технологического процесса, подготовки и выдачи отчетов о ходе технологического процесса, ручного управления объектом. Состав SCADA-систем МСКУ М представлен на рис. 3.17.

OPC-сервер (для ОС Windows NT) обеспечивает взаимосвязь МСКУ 2 и рабочих станций, функционирующих под управлением системного ПО МСКУ 2М, с рабочими станциями, функционирующими под управлением заимствованных SCADA (Genesis, In Touch, Trase Mode и т.п.).

Пакет Корунд. К основным его функциональным возможностям относятся:

- настройка пакета на состав технических средств рабочей станции и состав обслуживаемых МСКУ, подготовка базы данных рабочей станции (контуров сбора, обработки и контроля, контуров ручного управления), подготовка мнемосхем, настройка технологической клавиатуры, сбор, обработка и контроль информации от МСКУ;

- отображение информации в виде мнемосхем (время вызова новой мнемосхемы не превышает 1 s), выдача сообщений о нарушениях, накопление архива нарушений и оперативная работа с ним, обновление базы данных по инициативе МСКУ, накопление архива изменений параметров технологического процесса, передаваемых по инициативе МСКУ, и оперативная работа с ним (архив изменений параметров может формироваться как на диске, так и в памяти рабочей станции);

- подключение до двух функциональных клавиатур; прием инициативных сообщений о нарушениях от МСКУ, накопление архива нарушений и оперативная работа с ним, прием параметров технологического процесса по инициативе МСКУ, их преобразование, обработка по

алгоритмам, заданным пользователем при настройке, и отображение результатов обработки на мнемосхемах;

- контроль протокола обмена между обобщенной базой оперативных данных и локальными базами оперативных данных в МСКУ, регистрация нарушений обмена;

- распараллеливание процесса выдачи сообщений в файлы на диске о нарушениях в работе оборудования и о нарушениях технологического процесса), просмотр по инициативе оператора файлов о нарушениях и выдача выбранных сообщений на печать, подготовка и выдача отчетов о ходе технологического процесса (из обобщенной базы оперативных данных ПЭВМ) по инициативе оператора;

- управление объектом по командам оператора (ручное управление), преобразование данных в формат Lotus 1-2-3, возможность параллельной работы с 16 экранами (количество элементов динамики на мнемосхеме – до 256);

- введение групп технологических сообщений; использование стандартных графических редакторов для подготовки графических изображений (Paintbrush и т.п.).



Рис. 3.15. Структура системного ПО операторских и рабочих станций.

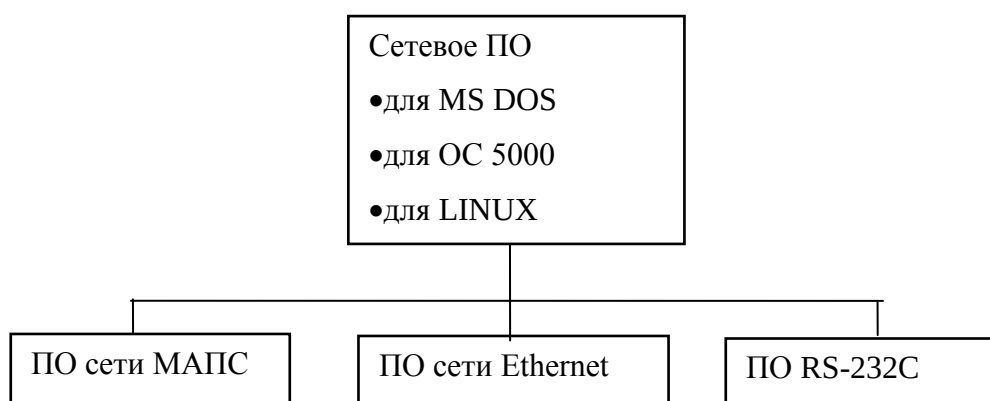


Рис. 3.16. Структура сетевого ПО операторских и рабочих станций.

Корунд обеспечивает подготовку контуров сбора, обработки и контроля технологической информации, контуров ручного управления из стандартного набора блоков, включающего: аналоговый ввод и вывод; дискретный ввод и вывод; логические и арифметические вычисления; линейное преобразование; сбор статистических данных и их контроль; суммирование; формирование тенденции (истории изменения сигнала); программный блок (для реализации нестандартных алгоритмов обработки и управления).

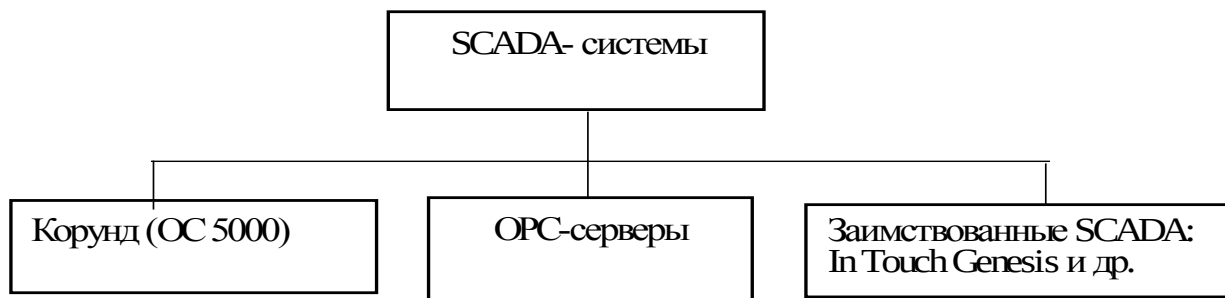


Рис. 3.17. Состав SCADA-систем МСКУ М.

На рис. 3.18 и 3.19 представлены примеры мнемосхем, подготовленных при помощи пакета Корунд для подсистемы визуализации в автоматизированной системе контроля и управления ядерными установками на производственном объединении «Маяк».

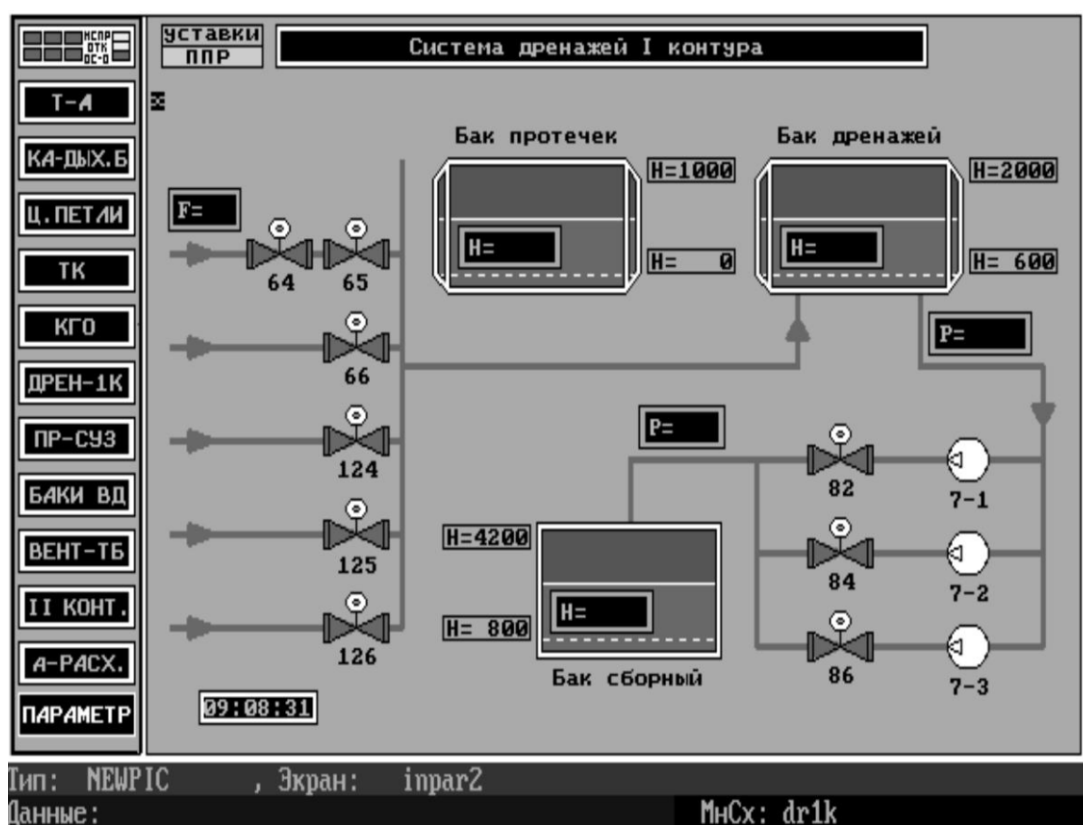


Рисунок 3.18 Мнемосхема «Система дренажей первого контура»

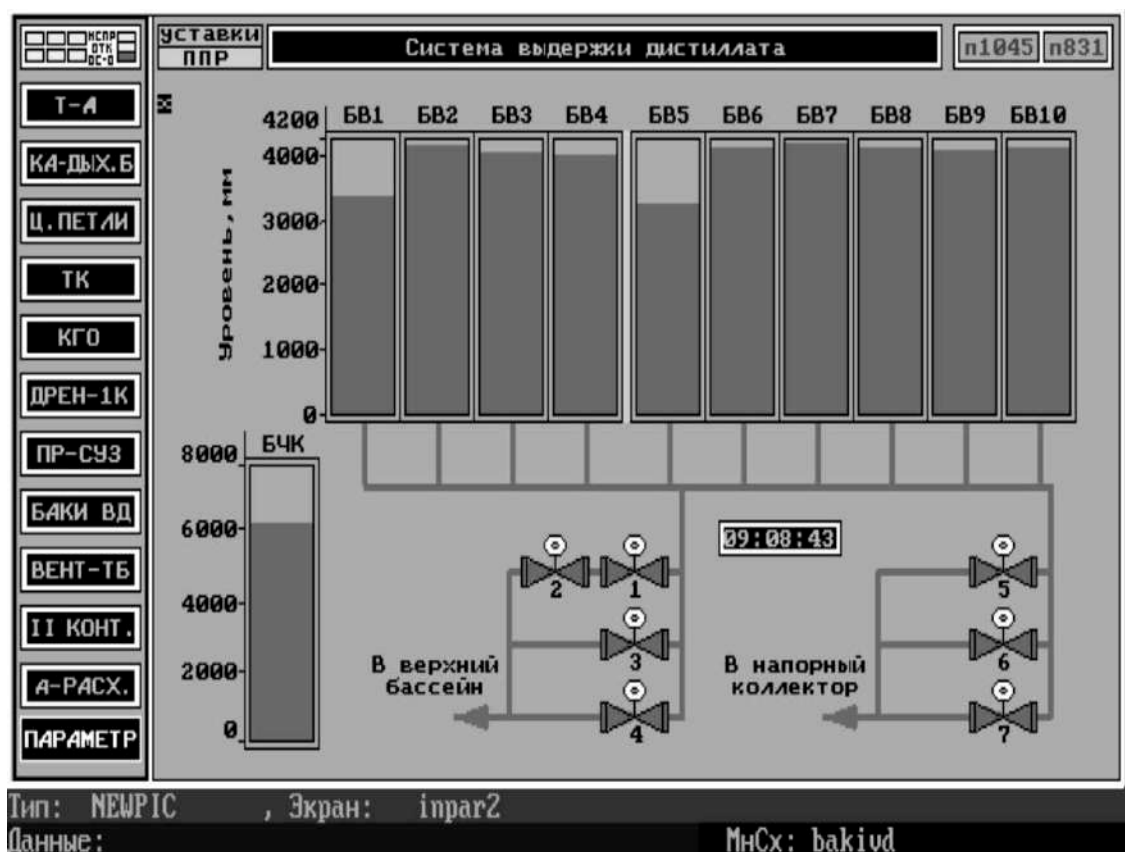


Рис. 3.19. Мнемосхема «Система выдержки дистиллата».

Другие SCADA-системы. Дополнительно к пакетам программ СОНАР-М и КОРУНД система МСКУ М представляет пользователям программные средства, обеспечивающие применение на рабочих станциях пакетов: Trace-Mode фирмы AdAstra, InTouch, Genesis.

ОПС-сервер МСКУ М. Пакет программ для верхнего уровня АСУ ТП - «ОПС-сервер МСКУ М» позволил не только создавать приложения АСУ ТП, работающие под управлением операционных систем семейства Windows и взаимодействующих с компонентами МСКУ М, но и использовать в построении SCADA МСКУ М такие пакеты как GENESIS-32, ICONICS [55]. Приложение «ОПС-сервер МСКУ М» реализовано в виде исполняемого файла и может применяться в качестве локального и удаленного сервера. ОПС-сервер предоставляет стандартный прикладной программный интерфейс (Application Program Interface – API), который позволяет использовать данные процесса управления прикладным программам. ОПС-сервер данных содержит объекты и интерфейсы, которые выполняют следующие функции: поддержка конфигурации ОПС-сервера; обмен данными с аппаратными средствами процесса управления; кэширование данных; навигацию по имеющейся информации; представление данных прикладным процессам согласно ОПС спецификации.

Структура взаимодействия ОПС-сервера МСКУ М с ОПС-клиентами показана на рис.3.20.

На рис. 3.21 отображено место ОПС-сервера в общей структуре ПТК на базе МСКУ М.

Программа настройки ОПС-сервера является графической оболочкой, позволяющей производить настройку ОПС-сервера. В качестве исходных данных при настройке используются файлы, получаемые при конфигурировании оперативной базы данных МСКУ М. Кроме этого, программа включает средства, позволяющие произвести соединение с удаленным или локальным сервером, просмотреть системную конфигурацию, просмотреть статическую информацию о работе сервера.

Интерфейсная динамически подгружаемая библиотека доступа к ОПС-серверу данных предназначена для предоставления прикладным процессам доступа к ОПС-серверу данных в

упрощенном виде. Задачей этой библиотеки является скрытие от задач пользователей всей сложности технологии ОРС.

Драйверы сети МАПС (драйвер контроллера связи КСв-31 и транспортный драйвер сети МАПС) обеспечивают сервис доступа к сети МАПС прикладным процессам, функционирующим в операционной среде Windows NT. В частности, посредством данной сети обеспечивается доступ к технологическим данным, находящимся в МСКУ. Драйверы разработаны в соответствии с требованиями по созданию сетевых драйверов Windows NT. Динамически подгружаемая библиотека ввода-вывода для доступа к драйверу МАПС обеспечивает стандартный прикладной программный интерфейс доступа к сети посредством указанных драйверов. Программа загрузки управляющей программы КСв-31 предназначена для начальной загрузки программы функционирования контроллера связи КСв-31 и изменения параметров, задающих способ подключения к сети (сетям) МАПС.

ОРС-менеджер представляет собой мощную программу ОРС-клиента, которая может быть использована для тестирования, просмотра и записи элементов и свойств ОРС-сервера. На рис. 3.22 представлена структура ОРС-сервера МСКУ М.

Оперативная база данных МСКУ М и ее элементы. Обмен данными между программами управляющей системы, задачами пользователя текущего узла и других узлов в МСКУ М осуществляется через базу данных, формируемую соответствующими программами текущего узла в ходе их выполнения [56]. База данных узлов (рабочие станции, МСКУ) включает данные двух видов: локальные и общесистемные.

Локальные элементы базы данных доступны только в пределах некоторого круга программ текущего узла по специально определенным для каждого вида внутренним правилам в соответствии с их функциональным назначением. Общесистемные элементы базы данных - это элементы, доступные всем задачам текущего узла, а также задачам других узлов системы. Общесистемные элементы могут быть использованы для передачи данных в другие узлы и приема данных от них. Совокупность всех общесистемных элементов называется оперативной базой данных комплекса МСКУ М, а элементы одного узла – оперативной базой данных узла (рабочей станции или конкретного МСКУ).

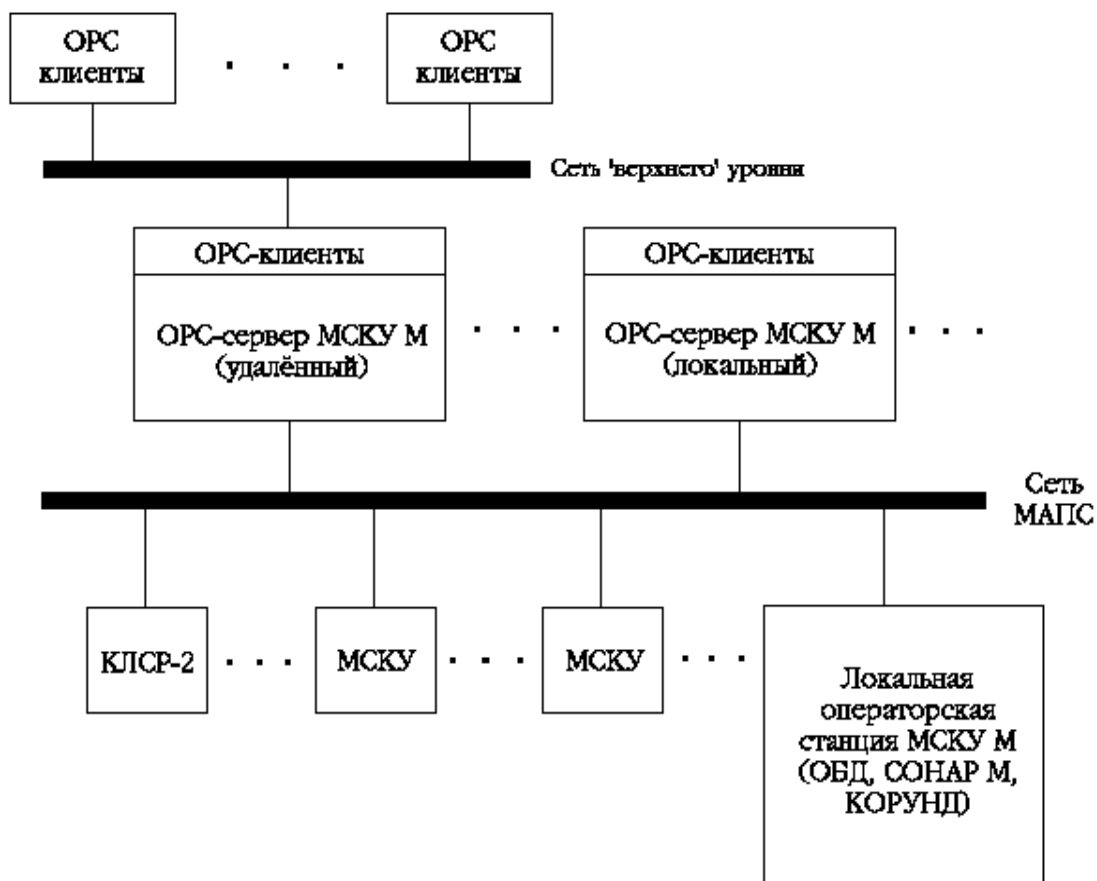


Рис. 3.20 Объектная модель комплексов МСКУ М

Элемент оперативной базы данных представляет собой минимальную единицу информации, доступную программам (измеренное значение сигнала, значение границы контроля и т.д.). Общесистемный объект оперативной базы данных (в дальнейшем - объект базы) - группа (либо один) взаимосвязанных элементов, объединенных по какому-либо функциональному признаку и имеющих общее, объединяющее их имя, называемое именем объекта базы данных. Элементы, принадлежащие одному многоэлементному объекту базы данных, называются еще его атрибутами. Атрибуты одного вида (одинакового функционального назначения и типа) идентифицируются именем, называемым идентификатором вида атрибута. Идентификатор вида атрибута в сочетании с именем объекта используется для доступа к значению атрибута.

Оперативная база данных компонентов комплексов МСКУ М представляет собой набор поименованных блоков следующих видов:

- переменная: одиночное значение типа int, unsigned int, long, unsigned long, char, unsigned char или float;
- одномерный массив: группа однотипных значений, называемых элементами массива, любого из перечисленных выше типов. Каждый элемент внутри массива идентифицируется его номером (целое число от 0 до n-1, где n - размерность массива);
- структура: группа разнотипных элементов, называемых полями структуры, в качестве которых могут использоваться переменные перечисленных выше типов, а также поля бит (длиной от 1 до 16 бит).

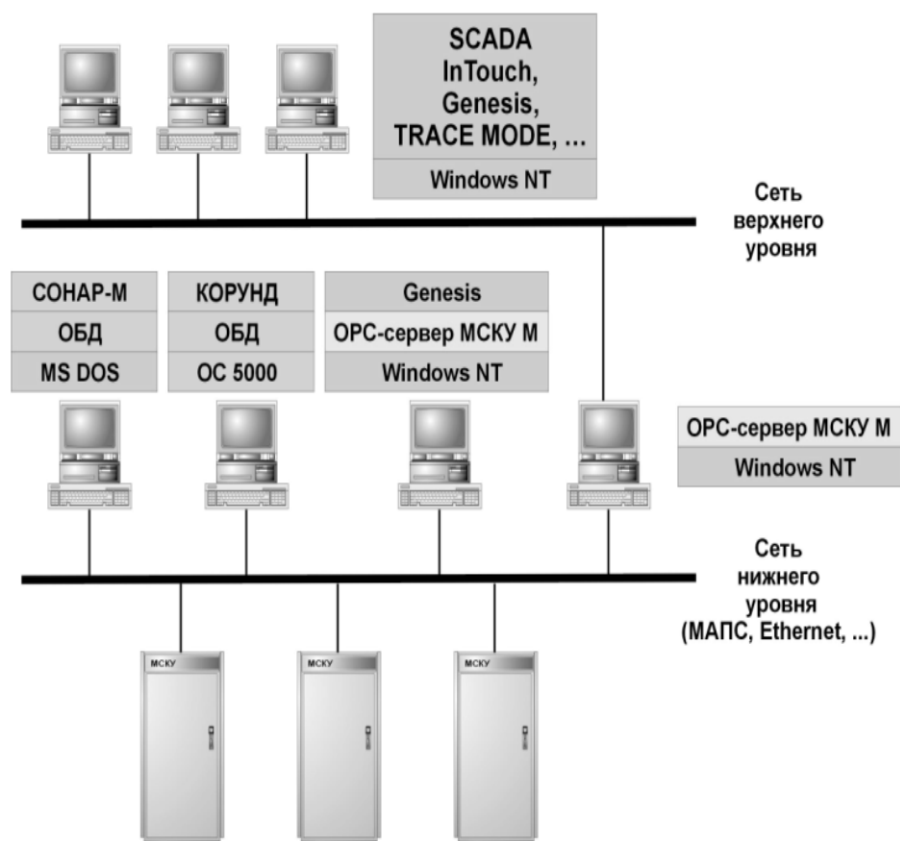


Рис. 3.21 OPC-сервер в общей структуре ПТК на базе МСКУ М

Объекты с одним элементом объявляются в оперативной базе данных как переменные соответствующего типа, идентификатор переменной является именем, а объекты с несколькими однотипными атрибутами объявляются в оперативной базе данных как одномерные массивы, идентификатор массива является именем объекта, а номер элемента в массиве - номером соответствующего атрибута данного объекта. Объекты с несколькими разнотипными (а возможно и однотипными) атрибутами могут быть объявлены в базе данных одним из двух способов: 1 - в виде структуры, полями которой являются атрибуты данного общесистемного объекта. Идентификатором структуры в этом случае является имя объекта, а идентификатором поля структуры – идентификатор вида атрибута; 2 - в виде нескольких структур, каждая из которых содержит атрибуты одного вида всех общесистемных объектов однотипного функционального назначения. Идентификатором каждой из структур в этом случае является идентификатор вида атрибута, а идентификаторами полей структур – имена объектов.

Независимо от способа распределения атрибутов общесистемного объекта по структурам каждому атрибуту присваивается идентификатор вида: <имя_объекта>. <идентификатор_вида_атрибута>. Таким образом, идентификатор элемента оперативной базы данных имеет один из следующих форматов: <идентификатор_переменной>, <идентификатор _ массива> [<номер_элемента>], <имя_объекта>.<идентификатор_вида_атрибута>. Идентификаторы описанного вида используются при создании блоков оперативных данных на этапе генерации управляющей системы МСКУ или оперативной базы данных рабочей станции.

Состав и структура оперативной базы данных рабочей станции определяется пользователем в программе генерации базы данных, в ходе обработки которой(в процессе генерации) создаются файлы для настройки программных средств данного пакета на требуемую структуру базы данных и заданный режим функционирования.

Общесистемные элементы могут быть объявлены непосредственно при генерации базы

данных узла либо посредством файлов объявлений данных элементов. Блоки также идентифицируются номерами, определяемыми в программе генерации.

Структура OPC-сервера МСКУ М

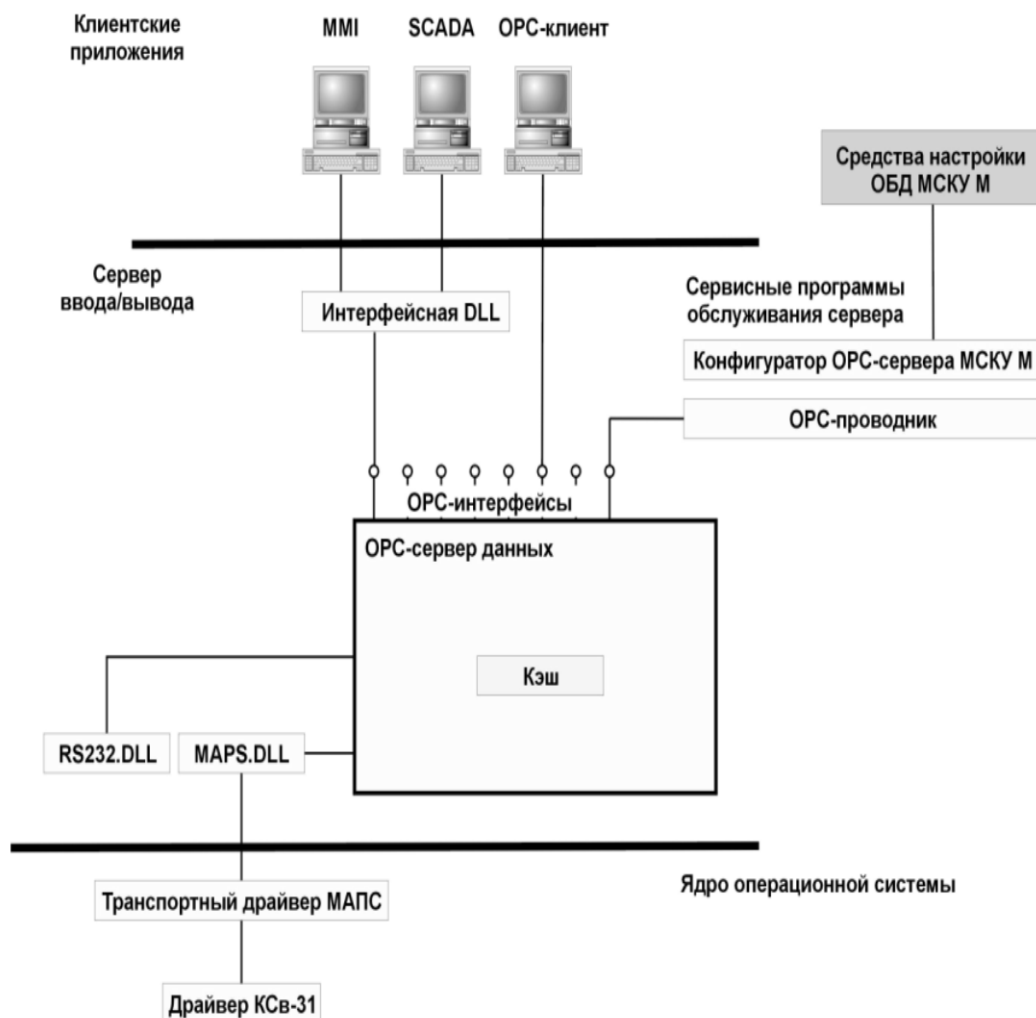


Рис. 3.22. Структура OPC-сервера МСКУ М.

В общем случае, все сведения о структуре базы данных конкретного абонента содержатся в файле-описателе структуры базы данных (с расширением .bn2), созданным на «этапе 2» процесса генерации.

Обмен информацией между узлами. Передача информации из оперативной базы данных текущего узла может осуществляться по запросу от других узлов с указанием состава требуемой информации; периодически или однократно по инициативе программ текущего узла, сгруппированной в блоки оперативных данных.

Состав и порядок информации, включаемой в блоки, определяются статически в программе генерации оперативной базы данных рабочей станции (в случае МСКУ - в программе генерации управляющей системы). Дисциплина передачи определяется статически в программе генерации путем задания периода повторения выдачи блока либо динамически в задачах пользователя путем передачи им заявок на выдачу тех или иных блоков программным средствам доступа к оперативной базе данных текущего узла.

Передача осуществляется циркулярными сообщениями (адресованными всем узлам системы) либо адресными сообщениями (адресованными конкретному узлу). Блоки идентифицируются номерами, определяемыми в программе генерации.

Необходимость приема информации из тех или иных блоков определяется в узлах-приемниках в их программах генерации, при этом дисциплина приема может быть выбрана одной из следующих:

- по запросу от узла-приемника путем чтения им информации из блока оперативных данных узла-источника по его номеру. Таким образом, может осуществляться чтение блоков как не включенных в процесс периодической передачи, так и включенных (например, в момент инициализации соответствующего узла-приемника с целью скорейшего получения необходимых данных);

- путем приема всех или некоторых данных из блоков оперативных данных, выдаваемых узлами-источниками, и размещения их в памяти узла в копиях этих блоков.

Все перечисленные выше способы обмена оперативными данными могут использоваться в любом сочетании по усмотрению пользователя, но наиболее предпочтительным является способ обмена через блоки оперативных данных по следующим причинам:

- такой способ обмена не требует от пользователя усилий по программированию операций обмена, необходима только соответствующая настройка программ доступа к оперативной базе данных комплекса;

- пропускная способность сетевых средств ограничена (для сети МАПС пропускная способность одной магистрали составляет порядка 100 - 150 сообщений в секунду и несущественно зависит от длины сообщения), а потому необходимо стремиться к уменьшению числа передаваемых сообщений. Блоки передаваемых данных позволяют компактно распределить информацию по сообщениям;

- передача информации по инициативе источника этой информации (в данном случае рабочая станция) позволяет повысить общую реактивность системы на изменения состояния объекта.

Прием сообщений об общесистемных объектах, передаваемых по инициативе МСКУ. В ряде случаев обмен между МСКУ и его абонентами, построенный на инициативных сообщениях, передаваемых данным МСКУ, может оказаться более эффективным по сравнению с обменом, построенным на операциях типа «запрос-ответ».

МСКУ может передавать инициативные сообщения с информацией об общесистемных объектах периодически с заданным при генерации циклом обновления; при изменении состояний дискретных параметров; при изменении значений аналоговых параметров на заданную величину; при нарушениях технологического контроля; по инициативе прикладных программ в МСКУ. Инициативные сообщения могут выдаваться МСКУ в виде циркулярных сообщений, адресованных всем центральным узлам системы, либо адресных сообщений, передаваемых конкретным абонентам, что определяется при генерации управляющей системы или при программировании задач пользователя в МСКУ.

Генерация обобщенной базы оперативных данных. Обобщенная база оперативных данных объявляется пользователем посредством текстового языка, операторами которого являются макрокоманды, определяющие состав общесистемных элементов, включаемых в обобщенную базу данных текущей рабочей станции, и требуемый режим обслуживания обобщенной базы данных.

Файл генерации может быть подготовлен любым штатным редактором текстовой информации рабочей станции и может служить исходным заданием для создания конкретной версии обобщенной базы данных. Файл, содержащий программу генерации, должен иметь расширение *.gen.

В файле генерации определяются такие элементы как структуры данных; переменные и массивы; блоки оперативных данных и включаемые в него элементы; состава информации, принимаемой из блоков оперативных данных других узлов; узлы, доступ к которым разрешен из текущего узла. Наименование входных и выходных файлов генерации приведены в табл. 3.5.

Генерация обобщенной базы оперативных данных рабочей станции проводится программой `ms_gen.exe` в два этапа: 1 - создание собственной оперативной базы данных и формирование передающих структур; 2 - создание приемных структур.

Входные и выходные файлы имеют следующие значения: `*.gen` - программа генерации обобщенной базы данных рабочей станции; `*.bn1` - двоичный файл описания базы данных своего узла, содержащий информацию об общесистемных элементах и блоках оперативных данных своего узла, создаваемый на первом этапе генерации; `**bn1` - двоичный файл описания базы данных другого узла, с которым необходим обмен информацией, содержащий информацию об общесистемных элементах и блоках оперативных данных этого узла и создаваемый на первом этапе генерации обобщенной базы данных в случае рабочей станции или на первом этапе генерации управляющей системы в случае МСКУ; `*.bn2` - двоичный файл-описатель обобщенной базы данных своего узла, содержащий информацию об общесистемных элементах и блоках оперативных данных своего узла и об общесистемных переменных и блоках оперативных данных других узлов, доступ к которым необходимо обеспечить из текущей рабочей станции; `*.ini` - файл инициализации обобщенной базы данных рабочей станции; `*.fbn` - текстовый файл, содержащий строки с указанием пути доступа к файлам `**bn1` (`**bn3`); `code_atr.h` - текстовый файл, содержащий информацию о соответствии между идентификаторами видов атрибутов и их кодами.

Таблица 3.5

Входные и выходные файлы программы генерации

Входные данные	Программа обрабатывающая	Выходная информация	Наименование этапа
<code>code_atr.h *.gen</code>	<code>ms_gen.exe</code>	<code>*.bn1</code>	Этап 1
<code>**bn1*.bn1 *.fbn</code>	<code>ms_gen.exe</code>	<code>*.bn2 *.ini</code>	Этап 2

Этапы генерации выполняются под управлением монитора системы генерации `ms_gen.exe` путем последовательного автоматического вызова генераторов `gn.exe`, `slink.exe`, `stage1.exe` и `stage2.exe` с использованием файла соответствия между идентификаторами видов атрибутов и их кодами (`code_atr.h`), поставляемого в составе данного пакета. Затем вызывается программа `sinit.exe`, которая создает файл для инициализации обобщенной базы данных ПЭВМ.

Вопросы

1. Каковы особенности ОС реального времени?
2. Какие типы ОС РВ существуют?
3. Какие методы диспетчеризации параллельных процессов стандартизованы?
4. Что такое семафор-счетчик?
5. Какие существуют механизмы диспетчеризации нитей?
6. Охарактеризуйте примитивы синхронизации нитей.
7. В чем состоит различие понятий «процесс» и «поток»?
8. Каковы основные особенности задач реального времени?
9. Дайте характеристику методов планирования задач реального времени.
10. Для чего необходима синхронизация задач реального времени?
11. Как осуществляется синхронизация задач реального времени с внешними событиями?
12. Для чего нужна синхронизация выполнения задач реального времени с астрономическим временем?
13. Какие основные особенности ОС РВ QNX?
14. Какие основные особенности ОС РВ OS-9?
15. Какие основные особенности ОС РВ OS5000?
16. Какие основные особенности ОС Windows NT?
17. Какие основные особенности ОС LINUX?

18. В чем состоит назначение SCADA-систем?
19. Какие основные функции SCADA-систем?
20. Какие основные особенности SCADA-систем?
21. Какова структура SCADA-системы?
22. Охарактеризуйте технологию проектирования системы автоматизации на основе SCADA.
24. Каковы функции систем технологического программирования?
25. Какой состав библиотеки стандартных алгоритмических модулей?
26. Охарактеризуйте языки программирования стандарта IEC-1131-3.
27. Дайте краткую характеристику системы ULTRALOGIC.
28. В чем сущность технологии COM?
29. Какие существуют методы межпроцессной коммуникации?
30. Где могут использоваться управляющие элементы ActiveX?
31. Для чего может использоваться механизм OPC?
32. В чем состоят достоинства применения OPC-стандарта?
33. Из каких основных объектов состоит OPC-сервер?
34. Какие типы OPC-серверов существуют?
35. Какие основные этапы включает методика проектирования задач управления технологическими объектами?
36. Какова структура программного обеспечения МСКУ 2М?
37. Охарактеризуйте операционные системы МСКУ 2М?
38. Какой состав управляющей системы МСКУ 2?
39. Охарактеризуйте исполнительную систему МСКУ 2?
40. Какие функции исполнительной системы МСКУ 2?
41. Какими процедурами управляет Монитор исполнительной системы?
42. Какой состав инструментального ПО МСКУ 2?
43. Что относится к средствам программирования МСКУ 2?
44. Какой состав системного ПО операторских и рабочих станций?
45. Какие основные особенности SCADA-систем МСКУ 2М?
46. Какова структура и функции OPC-сервера МСКУ 2М?
47. Что составляет оперативную базу данных ПТК на базе МСКУ 2М?

ПРИНЦИПЫ ОРГАНИЗАЦИИ ОТКАЗОУСТОЙЧИВЫХ ПТК

Важным вопросом при создании АСУ ТП является организация отказоустойчивых ПТК. Вопросам построения отказоустойчивых вычислительных систем и автоматики и методам оценки их надежности в литературе уделено достаточно много внимания [57-60]. Под *отказоустойчивостью* вычислительной системы понимается способность устранения последствий неисправностей элементов и программного обеспечения за счет использования аппаратной, информационной и алгоритмической избыточности. В отказоустойчивых системах необходимо различать отказ элементов и системы в целом. Если система обладает свойствами отказоустойчивости, то при отказах элементов она обычно сохраняет работоспособность в отличие от систем, не обладающих свойствами отказоустойчивости. Отказ элементов отказоустойчивой системы может быть необратимым или носить самоустраняющийся характер. Самоустраняющиеся отказы до момента их идентификации называют перемежающимися отказами.

Переход системы в неисправное состояние в результате, например, повреждения, может привести через определенное время к прекращению ее использования по назначению ввиду возникновения отказа системы или перехода системы в предельное состояние. Предельное состояние отказоустойчивой системы может возникнуть при определенных условиях, например, в случае неработоспособности системы, возникновения предаварийных режимов работы или перехода в состояние, при котором дальнейшее использование системы нецелесообразно или недоступно ввиду вредности или опасности при дальнейшей эксплуатации.

Поскольку существует много разных способов обеспечения избыточности в системе, а функции системы ограничены определенным кругом решаемых задач, то часто используется более узкое понятие отказоустойчивых систем. Например, для *отказоустойчивых вычислительных систем* основной задачей является способность их правильно выполнять заданные алгоритмы при наличии отказов в аппаратуре, ошибках в программе и т. д.

Мера, которой оценивают нечувствительность системы к возникающим в ней неисправностям (отказам), называется *свойством отказоустойчивости* или *просто отказоустойчивостью*. В качестве оценки меры часто принимают отношение наибольшего числа неисправных элементов, при котором система способна устранить последствия неисправностей, к общему числу элементов системы.

Особо актуальными есть вопросы обеспечения отказоустойчивости при разработке ПТК для управления непрерывными технологическими процессами. Невыполнение функций ПТК приводит к ухудшению качества продукции, потере мощности агрегатов, установок, в отдельных случаях – к аварийным ситуациям.

4.1 Системные способы обеспечения отказоустойчивости

Отказоустойчивость достигается реализацией комплекса следующих свойств ПТК: избыточностью, наличием средств обнаружения нарушений, диагностирования, устранения неисправностей, возможностью реконфигурации, возможностью восстановления исходных характеристик после устранения неисправности.

Избыточность в отказоустойчивой системе используется для организации процесса идентификации отказов и устранения их влияния на правильность выполняемых системой функций. Избыточность в системе обеспечивается, в частности, информационным, структурным, функциональным и временным резервированием. Прекращение пользования системой при резервировании происходит после исчерпания имеющихся резервов системы или при переходе ее в предельное состояние. Наиболее распространено аппаратное структурное резервирование. Последнее предполагает, что на N идентичных ресурсах ПТК синхронно выполняются N одина-

ковых вычислительных процессов. N - кратность резервирования ($N = 2, 3, 5, \dots$). Способы сравнения результатов вычислений могут быть чисто аппаратными, программными или комбинированными (аппаратно-программными). Аппаратные методы позволяют практически «мгновенно» обнаружить сбой (или отказ) в работе ПТК и локализовать его (при $N = 3, 5, \dots$). Они характеризуются большой избыточностью. Программные методы сравнения базируются на механизме «контрольных точек», когда после завершения вычислительных процессов производится сравнение их результатов. Наиболее распространенными являются комбинированные методы, в которых оптимальным образом сочетаются оба метода.

Основными средствами обнаружения нарушений являются различные виды контроля, как: аппаратный, программный, программно-логический. *Аппаратные* методы контроля базируются на использовании контроля по четности, кодов Хэмминга, циклических кодов, сторожевых таймеров, мажоритарных схем, применением специальной аппаратуры (п. 4.5.3), а *программные* базируются на использовании функциональных и специальных тестов. К *программно-логическим* методам контроля относятся контроль длительности и последовательности выполнения программных процедур, контроль гладкости изменения и выхода за допустимый диапазон измеряемых величин.

Под диагностированием подразумевается процедура локализации неисправности объекта, т.е. установления того, какая часть диагностируемого объекта неисправна. При диагностировании проводится установление неисправности объекта на более низком иерархическом уровне, чем при контроле. Мерой проникновения по иерархии объекта является глубина диагностирования. Принято говорить, что метод диагностирования обладает глубиной на уровне элементов, ТЭЗов, модулей и т. д. Требование большой глубины диагностирования увеличивает сложность, стоимость и продолжительность соответствующей диагностической процедуры. Поэтому необходимо выбрать такой метод диагностики, который обеспечил бы необходимую глубину диагностирования при минимальном расходе времени и средств. Вопрос о выборе глубины диагностирования решается исходя из организации процессов восстановления.

В целях быстрого восстановления системы и при наличии резервных устройств целесообразно ограничиться сначала установлением отказавшего устройства. Резервные устройства могут быть использованы для продолжения работы системы. Далее отказавшее устройство может подвергаться дальнейшему диагностированию в автономном режиме, в то время как система продолжает работать. При диагностировании отказавшего устройства при большей его глубине определяется отказавший ТЭЗ (или их группа). Далее отказавший ТЭЗ может быть заменен и восстановленное устройство возвращено в систему. Отказавший ТЭЗ диагностируется на специальном пульте. Его восстановление производится путем замены отказавших радиокомпонентов.

Реконфигурацией называется перестройка системных средств, вызванная необходимостью скомпенсировать действие отказов и обеспечить необходимыми ресурсами выполнение заданных функций. Процесс реконфигурации может включать в себя совокупность действий системы по физическому переключению линий связи, исключение отказавших блоков, перераспределение системной нагрузки. Стратегия и способы реконфигурации зависят от количественного и качественного состава системы и выбираются исходя из специфики для конкретной системы. Процесс реконфигурации системы, приводящий к ухудшению характеристик производительности и/или надежности называют процессом *деградации*, а состояние системы в таком случае – состоянием деградации. Реконфигурация может выполняться на фоне решения основных задач (динамически) или после снятия всех задач (статически).

Процедуры восстановления различаются либо относительно определенного класса неисправностей, либо исходя из метода обеспечения работоспособного состояния системы после возникновения в ней отказа. В одном случае восстановление определяется как совокупность действий, которые осуществляются системой по получении сигнала о неисправности, в другом случае восстановление – это реконфигурация структуры системы при отказах элементарных ЭВМ или процессоров с целью перераспределения решаемых системой задач между работоспособными модулями, а в третьем случае под восстановлением подразумевается замена отказав-

шего элемента и его ремонт. В дальнейшем, если термин «восстановление» относится к системе (подсистеме), находящейся в состоянии деградации, то его следует понимать как процесс, обратный процессу деградации. В отношении отказавшей системы (или элемента) восстановление следует понимать как ремонт этой системы (или элемента).

Различают восстановление после трех видов неисправностей (ошибок): аппаратных, ошибок программного обеспечения, при управлении вычислительным процессом. Соответственно выделяют три принципиально различных уровня восстановления: аппаратный (физический), программный и уровень управления. Процедуры восстановления разных уровней различаются.

Восстановление функционирования после возникновения неисправности – одна из наиболее трудно реализуемых функций ПО, так как в ПТК ограничена способность программных и аппаратных средств к обнаружению, диагностированию, локализации неисправностей, а при эксплуатации ПТК могут возникать непредвиденные ситуации.

4.2 Методы организации отказоустойчивого программного обеспечения

Особой проблемой является разработка высоконадежного программного обеспечения для отказоустойчивых ПТК. ПО отказоустойчивых ПТК наряду с традиционными функциями ПО должно обеспечивать контроль и диагностирование оборудования, восстановление его после отказа, проводить реконфигурацию структуры ПТК.

При разработке ПО должны решаться две основные задачи:

- создание надежного (безошибочного) базового программного обеспечения;
- проектирование и включение программных средств для обеспечения живучести при программных, информационных и аппаратных отказах [60].

Основными средствами уменьшения ошибок в ПО являются не только рациональный выбор структуры и тщательность проектирования, применения и проверки ПО, но и эффективное использование избыточности в форме жестких структур данных и информации о том, каково ожидаемое поведение программного обеспечения. К обеспечению надежности программного обеспечения прослеживается четыре основных подхода: предупреждение ошибок, обнаружение ошибок, исправление ошибок и обеспечение устойчивости к ошибкам. К первому подходу относятся принципы и методы, позволяющие минимизировать или вообще исключить ошибки. Методы второго подхода сосредотачивают внимание на функциях программного обеспечения, помогающих выявить ошибки. К третьему подходу относятся функции программного обеспечения для исправления ошибок или их последствий. Под устойчивостью подразумевается способность программного обеспечения продолжать функционирование при наличии ошибок. Это свойство программного обеспечения является едва ли не самым основным при создании отказоустойчивых систем.

Рассмотрим методы обеспечения устойчивости ПО к ошибкам. Методы этой группы ставят своей целью обеспечивать функционирование программной системы при наличии в ней ошибок. Г. Майерс разбивает их на три класса: динамическая избыточность, методы отступления и методы изоляции ошибок. Программная избыточность обычно выражается в многократном хранении критических программ и данных; наличии диагностических программ на различных программных и аппаратных уровнях; особых качествах прикладных программ, обеспечивающих программные рестарты и отказоустойчивый аппаратный интерфейс с операционной системой. Избыточное программное обеспечение не должно требовать сложных проверок; не базироваться на диагностике в процессе работы программного обеспечения; содержать независимые процедуры для одних и тех же функций.

Программная динамическая избыточность подобна аппаратной избыточности. В отличие от повторного выполнения операций на резервных устройствах предполагается повторное выполнение вычислений на других версиях программ. Очевидно, что в случае ошибок в одной из версий использование точной копии такой версии не даст никаких результатов. Необходимо создание версии некоторыми другими средствами, например, на другом языке, с использованием

другого алгоритма и т. д. Такой подход называется в литературе *N*-версионным (многовариантным). Этот метод имеет ряд серьезных недостатков: не дает желаемого результата при ограничениях на ресурсы и производительность системы; теряет смысл для задач, имеющих несколько путей решения, так как для них невозможно осуществить проверку – сравнение в контрольных точках; неприменим, если последовательность выдаваемых результатов нельзя упорядочить; в ряде случаев последовательность выходных значений программ является контекстно зависимой и любая ошибка, сдвигающая последовательность, приводит к ошибке при сравнении; метод трудно применим, если нельзя достаточно точно определить величины допустимых расхождений для сравниваемых значений в контрольных точках. На практике этот метод можно использовать если не полностью, то частично, когда результаты выполнения отдельной операции, для которой существует запасная копия, признаны неправильными либо когда выполнение некоторой программы существенно для живучести системы в целом.

Вторая группа методов обеспечения устойчивости к ошибкам – методы отступления или сокращенного обслуживания. Применение их ограничивается теми случаями, когда для системы важно лишь закончить работу, например, обеспечить безаварийное завершение всех процессов, протекающих под управлением системы. Наконец, третья группа – методы изоляции ошибок. В этом случае необходимо обеспечить наименьший выход последствий ошибки за пределы как можно меньшей части системного программного обеспечения, чтобы при возникновении ошибки не вся система оказалась неработоспособной, а отключались бы лишь отдельные функции в системе либо некоторые ее пользователи. Часть таких методов связана с защитой каждой из программ в системе от ошибок других программ, в частности с защитой операционной системы от ошибок прикладных программ. Многие ошибки можно сделать «неощутимыми» для программ. Для этого большинство операционных систем обеспечивает многократность операций ввода-вывода, что ликвидирует перемежающиеся отказы этих устройств.

Важное обстоятельство, касающееся всех методов обнаружения, исправления ошибок и устойчивости к ним, состоит в том, что все методы требуют дополнительных функций от самого программного обеспечения. Это увеличивает сложность системы и может привести к новым ошибкам. Еще одним крупным источником ошибок в ПО является внесение изменений в готовые программы.

Использование современных прогрессивных технологий программирования повышает надежность ПО прежде всего за счет уменьшения числа проектных ошибок. Наибольшее применение получили объектно-ориентированный и структурный подходы. Такие подходы объединяют несколько способов создания ясной, легкой для понимания программы. При отладке написанных программ важно тщательно провести тестирование программного продукта. Программы нельзя считать правильными до тех пор, пока это качество не доказано. Цель тестирования состоит в том, чтобы убедиться, что программа решает действительно ту задачу, для которой предназначена, и выдает правильный ответ при любых условиях. Исчерпывающее тестирование программы не оправдано с экономической точки зрения и обычно неосуществимо на практике. В связи с этим в работе [60] отмечено, что экспериментальное тестирование программ может служить доказательством наличия ошибок, но никогда не докажет их отсутствия.

4.3 Способы построения отказоустойчивых систем

Вопросы построения отказоустойчивых вычислительных систем волновали проектировщиков с момента появления ЭВМ. Попытки обеспечить отказоустойчивость в системах относятся еще к первому поколению ЭВМ, в которых применялись, в основном, различные средства и методы резервирования с небольшой кратностью резерва: резервирование на уровне процессоров, на уровне функциональных плат и узлов, информационное резервирование данных и программ. Эти методы продолжают широко использоваться и в современных системах. Для повышения отказоустойчивости, например, в ранних системах, применялось тройное резервирование модулей процессора и информационное резервирование данных, хранимых в запоминающих устройствах (ЗУ).

Показательной является **электронная система коммутации ESS**, структура которой показана на рис. 4.1. Она включает быстродействующий дублируемый центральный процессор и периферийные устройства, непосредственно управляющие коммутацией телефонных каналов. Резервный процессор обеспечивает управление системой при обнаружении отказов в основном процессоре. При этом работать может только один из них. Каждый процессор ESS первого поколения содержит центральное устройство управления ЦУУ и два устройства памяти: запоминающее устройство программ ЗУП и вызовов ЗУВ. В ЗУП, выполненном в виде постоянного запоминающего устройства (ПЗУ), находятся программы обработки вызовов, обслуживания системы и программы диспетчеризации, а также параметры системы и ряд рабочих программ. Резервирование на уровне процессоров не обеспечивает высокой надежности при малых аппаратных затратах. Поэтому в системе ESS второго поколения, структура которой показана на рис. 4.2, резервирование ведется уже на уровне устройств ЗУП, ЗУВ, ЦУУ, а также шин вызовов, программ и периферийных блоков.

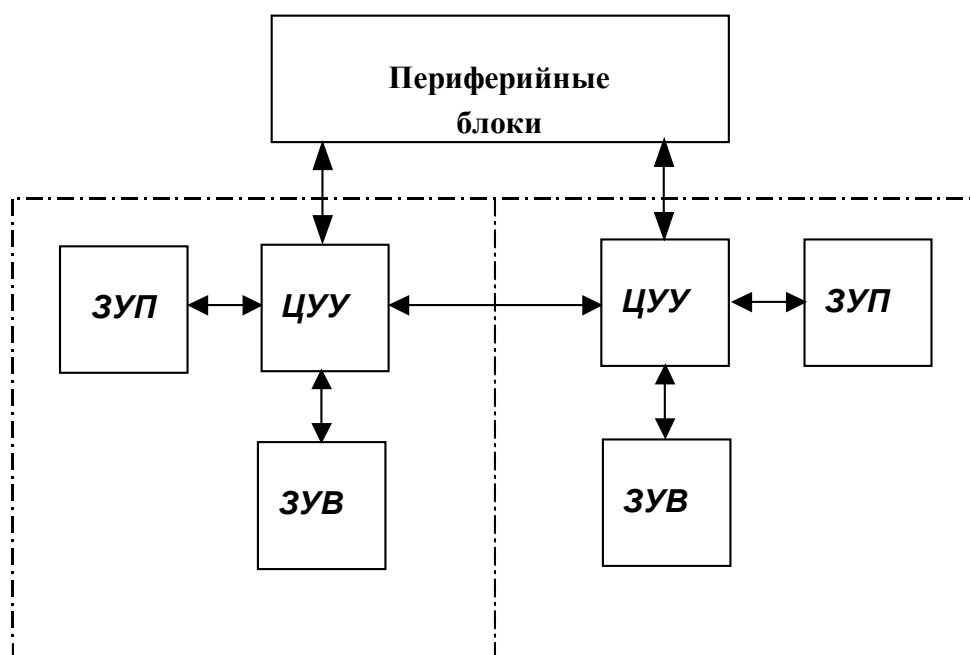


Рис. 4.1. Структурная схема системы ESS первого поколения.

Центр обслуживания, включающий схему сравнения, производит сравнение данных обмена между ЗУП и ЦУУ, выдачу по шинам ошибок команд останова резервного ЦУУ или прогона программ диагностирования рабочего ЦУУ. Локализация неисправностей осуществляется с помощью диагностической программы и встроенной аппаратуры контроля (схем самопроверки), которая контролирует также микропрограммное управление системы. Защита от неисправностей в ЗУВ производится дублированием записей в рабочем и резервном ЗУВ. При возникновении неисправностей в рабочем ЗУВ в работу включается резервное.

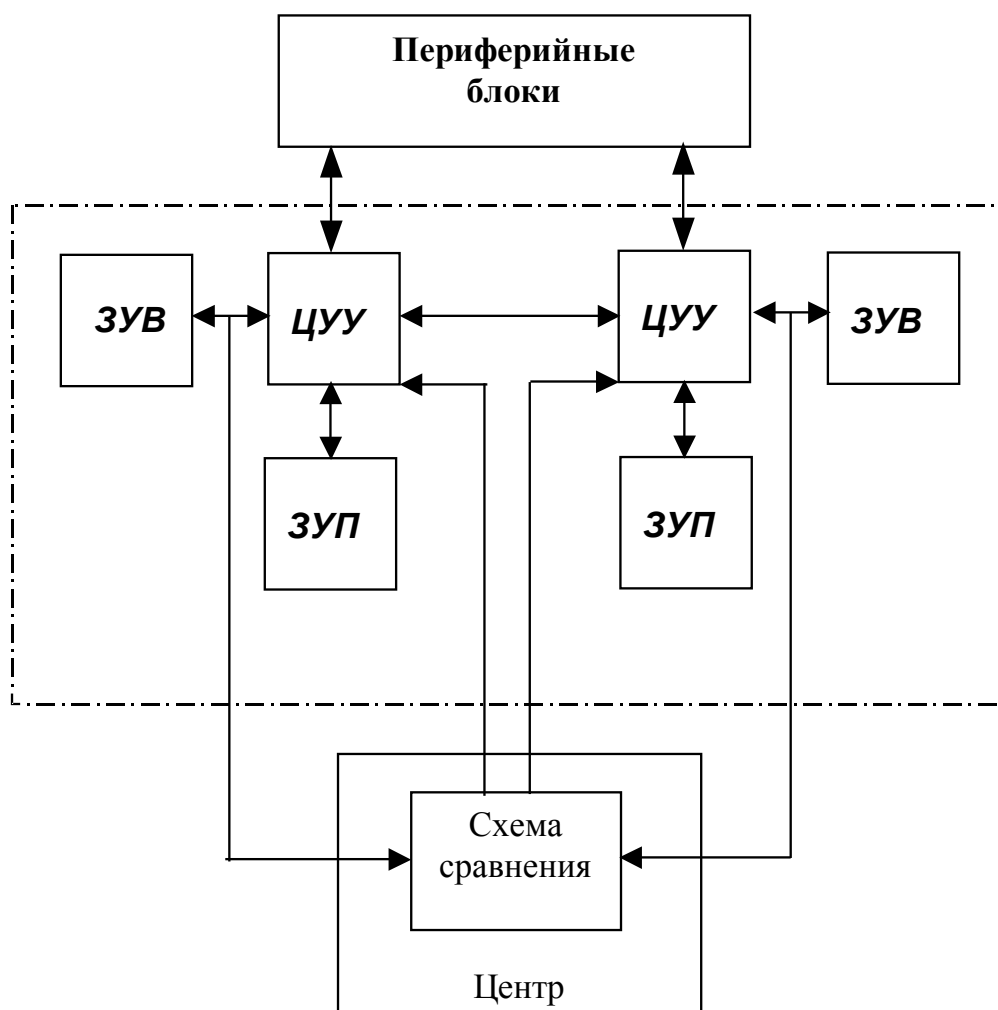


Рис. 4 2. Структурная схема системы ESS второго поколения.

Отказоустойчивость в системах ESS обеспечивается следующими мерами:

- обнаружением неисправностей путем сравнения выходов идентичных синхронно работающих блоков;
- введением избыточного кодирования (в частности, использованием кодов m из n) цифровой информации в системе;
- применением встроенных схем контроля, которые совместно с диагностическими программами позволяют устранять влияние значительной части неисправностей;
- использованием таймера аварий, который разрешает перебор возможных конфигураций системы с целью создания работоспособной структуры. Работоспособность выбранной конфигурации процессора определяется с помощью специальной программы;
- введением специального канала обслуживания, по которому производится диагностирование одного процессора другим;
- автоматическим и ручным восстановлением процессора при обнаружении неисправностей.

Восстановление работоспособности системы при неисправностях в аппаратной части осуществляется отключением рабочего процессора и запуском резервного. Переход на продолжение программы происходит в три этапа: установка состояния системы (задание начального адреса, состояний регистров ошибки, сброс триггеров остановки и др.); защита данных, хранимых в резервном ЗУ; запуск программ.

Ручное восстановление используется в тех случаях, когда автоматическое не обеспечивает приведение системы в работоспособное состояние, например, в случае безуспешной многократной инициализации.

Примером **троированной системы** может служить система реального времени August 300 (80-е годы) [60], рассчитанная на непрерывную круглосуточную работу в условиях производства. Высокая надежность системы обеспечивается тремя процессорами, тройным резервированием основных схем и принятием решений по трехступенчатому мажоритарному принципу. Основу системы August 300 составляют три модуля управляющего компьютера ССМ (Control Computer Module), выполняющие внутренние операции независимо друг от друга. Причем каждый из них реализует полный алгоритм управления, в том числе все операции по контролю средств защиты и по аварийному отключению технологического оборудования. Чтобы гарантировать возможность обнаружения и исправления ошибок любого процессора, в каждом из них предусмотрено считывание данных из ОЗУ остальными двумя ССМ через каналы, допускающие только чтение; в то же время ни один из процессоров не допускает запись в память другого процессора.

Модули ССМ связываются с оборудованием технологического процесса (с датчиками и исполнительными элементами) с помощью одного или более модулей интерфейса процессора РИМ (Process Interface Module). Модуль РИМ построен таким образом, что все входные данные каждый из ССМ может читать независимо через отдельные схемы. Выдача управляющих воздействий осуществляется через схемы, работающие по мажоритарному принципу.

Основные принципы построения August 300 представляет собой развитие концепции SIFT (Software Implemented Fault Tolerance) – отказоустойчивость, реализованная программным способом. Полный цикл работы отказоустойчивой управляющей системы August составляют прием входных данных по трем независимым каналам, мажоритарный выбор входных данных для вычислений, вычисление по заданному алгоритму, программный мажоритарный выбор результатов вычислений каждого ССМ с последующим окончательным аппаратным выбором управляющих сигналов с помощью выходной мажоритарной схемы.

Системой с полностью аппаратным способом обеспечения отказоустойчивости служит система Stratus 32 с избыточными средствами самоконтроля. Ограничение большинства программных способов состоит в том, что они контролируют правильность работы системы с определенными интервалами. В результате этого в случае обнаружения ошибки приходится возвращаться и аннулировать некоторые из результатов вычислений. Преимущество аппаратного способа обеспечения отказоустойчивости вычислительной машины заключается в том, что он действует в реальном времени, позволяя обнаружить аппаратную ошибку непосредственно после ее возникновения и сразу же принять необходимые меры. В системе Stratus 32 все схемные платы дублируются, причем они, в свою очередь, содержат идентичные логические блоки. В процессе работы осуществляется автоматическое сравнение сигналов блоков, поэтому аппаратные отказы обнаруживаются немедленно. Основной подход, принятый в системе Stratus 32, состоит в том, что на плате размещаются пары всех логических компонентов и добавляется логика, необходимая для сравнения выходных результатов и выполнения некоторых действий при несовпадении результатов. Поскольку стоимость логических схем и микропроцессоров уменьшается, такой подход оправдан экономически. После обнаружения неисправности схемной платы в Stratus 32 программными средствами технологического обслуживания для нее выполняется ряд тестов. Если тесты проходят нормально, то предполагается, что ошибка кратковременная (сбой), и плата снова включается в работу. При возникновении повторной ошибки плата отключается. В этих случаях вся информация относительно неисправностей заносится в аппаратный журнал дискового файла последующего анализа. Ремонт системы должен осуществляться человеком.

Методы резервирования современных ПТК. Ведущие фирмы изготовители промышленных контроллеров предлагают специальные средства для построения ПТК с резервированной структурой. Так, в составе изделий фирм SIEMENS и SCHNEIDER-ELECTRIC есть специальные модули для построения дублированных управляющих систем с «горячим» резервирова-

нием. Структура подобной системы приведена на рис. 4.3. Она содержит два идентичных контроллера (основной и резервный), арбитр и схему коммутации выходных сигналов. Входные сигналы от объекта поступают в оба контроллера. Принцип обеспечения отказоустойчивости в такой системе заключается в следующем. Арбитр, содержащий схему анализа работоспособности каждого контроллера, управляет схемой коммутации выходных сигналов. В исходном состоянии коммутируются сигналы основного контроллера. В каждом из контроллеров наряду с основными задачами выполняются процедуры самодиагностирования, результаты которых в реальном времени поступают в арбитр. Каждый из контроллеров проверяет работоспособность другого путем анализа поступающих от него специальных сигналов. Такая процедура называется «проверка пульса». Результаты анализа также поступают в арбитр. При обнаружении ошибок в работе одного из контроллеров выходные сигналы коммутируются от другого. Неисправный процессор диагностируется и ремонтируется. Как правило, ремонт контроллера заключается в замене неисправного ТЭЗа на исправный, а неисправный восстанавливается в помощью специального пульта. После замены ТЭЗа запускаются контрольные тесты, после успешного завершения которых контроллер включается в работу специальной процедурой восстановления.

Применение описанного способа повышения отказоустойчивости оправдано, если каждый из контроллеров имеет большое время наработки на отказ (десятки и сотни тыс. часов) и малое время восстановления (меньше часа). В этом случае вероятность одновременного отказа контроллеров ничтожно мала. Показатели надежности арбитра и схемы коммутации должны на порядок и более превышать аналогичные показатели контроллера.

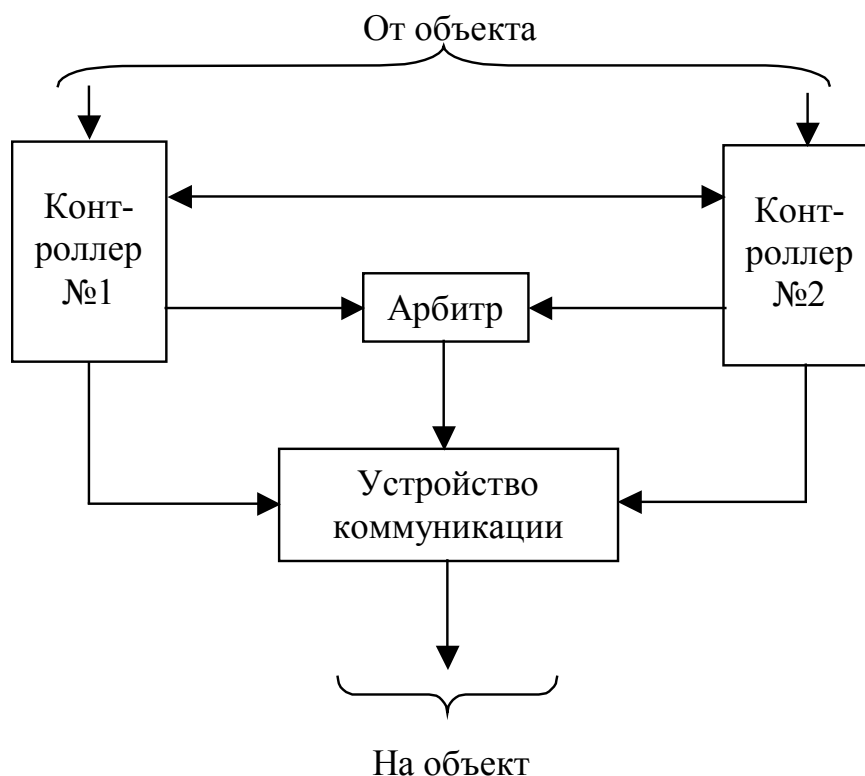


Рис. 4.3. Структура дублированной управляющей системы.

Другим способом построения отказоустойчивых ПТК является построение троированных структур. На рис. 4.4. приведена схема, иллюстрирующая работу троированной системы.

Как и в предыдущем примере, входные сигналы и другая информация поступают одновременно в три идентичных между собой контроллера, в которых синхронно выполняется одна и та же программа. После очередного цикла получения результатов для управления объектом они от каждого контроллера поступают на мажоритарную схему «2 из 3», формирующую интегральный результат.

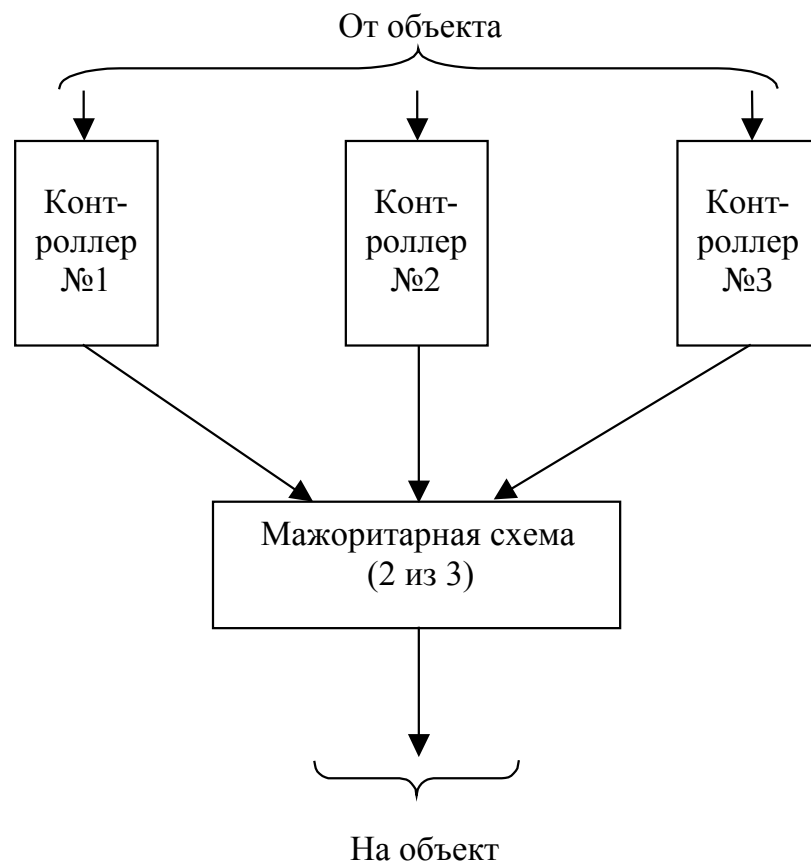


Рис. 4.4. Структура троированной системы.

4.4 Организация отказоустойчивых ПТК на базе МСКУ М

Построение отказоустойчивых ПТК на базе МСКУ М обеспечивается комплексом организационных, технологических, системных, аппаратных и программных решений как:

- тщательной конструктивной проработкой ПТК и его компонентов;
- выбором надлежащей элементной базы, входным контролем комплектующих и их выбраковкой;
- всесторонними испытаниями блоков элементов, узлов, устройств и всего ПТК;
- длительной непрерывной приработкой ПТК при граничных значениях параметров окружающей среды, что предотвращает практически все неисправности в начале эксплуатации;
- резервированными фидерами электропитания, аккумуляторной поддержкой, использованием устройств бесперебойного питания;
- возможностью резервирования (дублирование, троирование) на уровне узлов, модулей, подсистем и шин передачи данных;
- контролем выдаваемых на управляемый объект сигналов;
- безударным переходом на ручное управление исполнительными механизмами при отказе управляющей системы;
- встроенными аппаратно-программными средствами диагностирования, позволяющими обнаружить неисправность, блокировать работу неисправного компонента, а после его замены обеспечить автоматическое включение в рабочий режим.

Выбор конкретных решений по обеспечению необходимого уровня отказоустойчивости ПТК производится проектировщиком ПТК в соответствии с требованиями технического задания на разработку ПТК.

Рассмотрим системные, аппаратные и программные аспекты обеспечения отказоустойчивости ПТК на базе МСКУ М.

Отказоустойчивость сети МАПС. Программно-аппаратные коммуникационные средства нижнего и верхнего уровней МСКУ М обеспечивают построение дублированных и троированных сетей. Рассмотрим методы обеспечения отказоустойчивости коммуникационных средств ПТК на примере сети МАПС [32]. Принципы организации сети МАПС рассмотрены в главе 2 и Приложении Б. Сеть МАПС является важнейшим компонентом ПТК с точки зрения обеспечения его отказоустойчивости, т.к. ее отказ приводит к отказу ПТК в целом по функциям контроля и управления объектом (в отличие, например, от отказа рабочих станций или сетей верхнего уровня).

Повышение отказоустойчивости сети достигается за счет нескольких факторов: применением двух или трех магистралей; многократным резервированием программно-аппаратных средств арбитража сети; возможностью многократных повторов передач пакетов при обнаружении сбоев; контролем пакетов при их передаче за счет специальных контрольных кодов; мониторингом сети в реальном времени; функциональным фоновым тестированием. Первые три фактора относятся к методам аппаратной, программной и временной избыточности. Последние три фактора направлены на обнаружение сбоев и отказов в работе узлов сети в процессе штатной работы ПТК и предоставление соответствующей информации для подсистемы реконфигурации.

Подключение абонентов сети к каждой магистрали МАПС осуществляется отдельным контроллером связи. Контроллеры связи выполняют функции физического и канального уровней взаимодействия в сети МАПС. В любой момент времени управление работой магистрали МАПС осуществляет один из контроллеров связи, выполняющий, помимо абонентских функций физического и канального уровней, функции основного арбитра магистрали. Контроллеры связи других абонентов, выполняя абонентские функции, как арбитры магистрали находятся в «холодном резерве». В случае выхода из строя контроллера связи, выполняющего функции основного арбитра магистрали, основным назначается рабочий контроллер связи с текущим минимальным адресом. После ремонта неисправного контроллера связи он включается в рабочий режим с выполнением абонентских функций и переводом в «холодный резерв» как арбитра сети. Таким образом, управление каждой магистралью МАПС выглядит как N -резервированная система с восстановлением, где N - число абонентов сети.

Процедура первоначального определения состояния абонентов магистрали выполняется арбитром при инициализации его работы с целью определения конфигурации сети МАПС (состава абонентов и их типов). В процессе функционирования арбитр помимо основных функций управления (приложение Б) выполняет мониторинг - сканирует изменения конфигурации (выход из строя абонента, включение в работу ранее вышедшего из строя абонента, дополнение состава новыми абонентами и т.п.), фиксирует различные нештатные ситуации (сбои при передаче служебных и информационных пакетов, нарушения протоколов обмена и пр.).

Процесс передачи информационных и служебных пакетов контролируется с использованием механизма контрольных сумм. При обнаружении ошибки после приема пакета данный факт квалифицируется как сбой и на канальном уровне (уровне контроллера связи) организуется многократный повтор передачи пакета. При неустранении ошибки она квалифицируется как отказ, что приводит к вызову процедур диагностирования и реконфигурации. При реализации конкретных ПТК возможны варианты организации функционирования сети МАПС. В первом варианте все обмены информацией производятся по одной (рабочей) магистрали, ее диагностирование производится в фоновом режиме. Другие (резервные) магистрали диагностируются непрерывно, т. е. постоянно определяется их готовность к работе. При нарушении в передаче информации по рабочей магистрали, обнаруженной фоновыми процедурами диагностирования или механизмом контрольных сумм, подсистема реконфигурации автоматически производит изменение конфигурации сети: неработоспособная исключается из рабочей топологии с выдачей сообщения о конкретной неисправности, а одна из резервных магистралей назначается рабочей. Аналогичные действия производятся и в том случае, если обнаружена неисправность резервной магистрали. После восстановления магистрали она автоматически включается в число резервных.

Во втором варианте все обмены информацией производятся по всем магистралям. При этом магистрали диагностируются в фоновом режиме. При обнаружении нарушений в работе какой-либо магистрали, как и в первом варианте, автоматически производится реконфигурация сети: неработоспособная исключается из рабочей топологии с выдачей сообщения о конкретной неисправности, обмены информацией производятся по оставшимся магистралям. После восстановления магистрали она автоматически включается в число рабочих. Отметим, что во втором варианте проектировщик АСУ ТП должен учитывать, что может наступить событие (хотя вероятность его ничтожно мала), когда в системе работоспособна одна магистраль из трех. И в этом случае уменьшение пропускной способности сети не должно отрицательно влиять на работу АСУ ТП в целом и не вносить ограничений в работу оператора-технолога.

Отказоустойчивость верхнего уровня ПТК. Повышение отказоустойчивости сети верхнего уровня достигается применением дублированной радиальной сети Industrial Ethernet и за счет использования избыточных рабочих станций ПС 5110, на базе которых строятся рабочие места оперативного персонала, серверы архивирования, инженерные станции, файл-серверы, межсетевые шлюзы, информационно-вычислительные комплексы и пр.

Все рабочие станции могут иметь до трех сетевых портов. Избыточные рабочие станции работают в режиме «горячего резерва», при этом каждая работает практически автономно и выполняет необходимые функции по представлению информации и управлению процессом.

В качестве примера рассмотрим структурную организацию ПТК, реализующего функции информационно-вычислительной системы (ИВС) энергоблока АЭС с реактором ВВЭР-1000. Описание ИВС будет приведено в разделе 6. На рис.4.5 и 4.6 приведены структурные схемы ПТК ИВС и комплекса связи с объектом (КСО). На схемах обозначено:

- РМО НСРЦ – рабочее место начальника смены реакторного цеха;
- РМО НСБ - рабочее место начальника смены блока;
- РМО ВИУР - рабочее место ведущего инженера управления реактором;
- РМО ВИУТ - рабочее место ведущего инженера управления турбиной;
- ИРС КСО – инженерная рабочая станция;
- СДИ – станция дежурного инженера.

Все важные элементы ПТК зарезервированы. Так, функции блочного щита управления резервируются с помощью резервного щита управления. РМО ВИУР, РМО ВИУТ, инженерные станции СДИ имеют по две рабочих станции, каждая из которых укомплектована тремя мониторами. Дублированную структуру имеют серверы архивирования, вычислительные серверы. Связь между подсистемами нижнего уровня (КСО) и подсистемами верхнего уровня реализована посредством дублированной радиальной сети Industrial Ethernet. В КСО связи между МСКУ реализованы посредством дублированной сети МАПС.

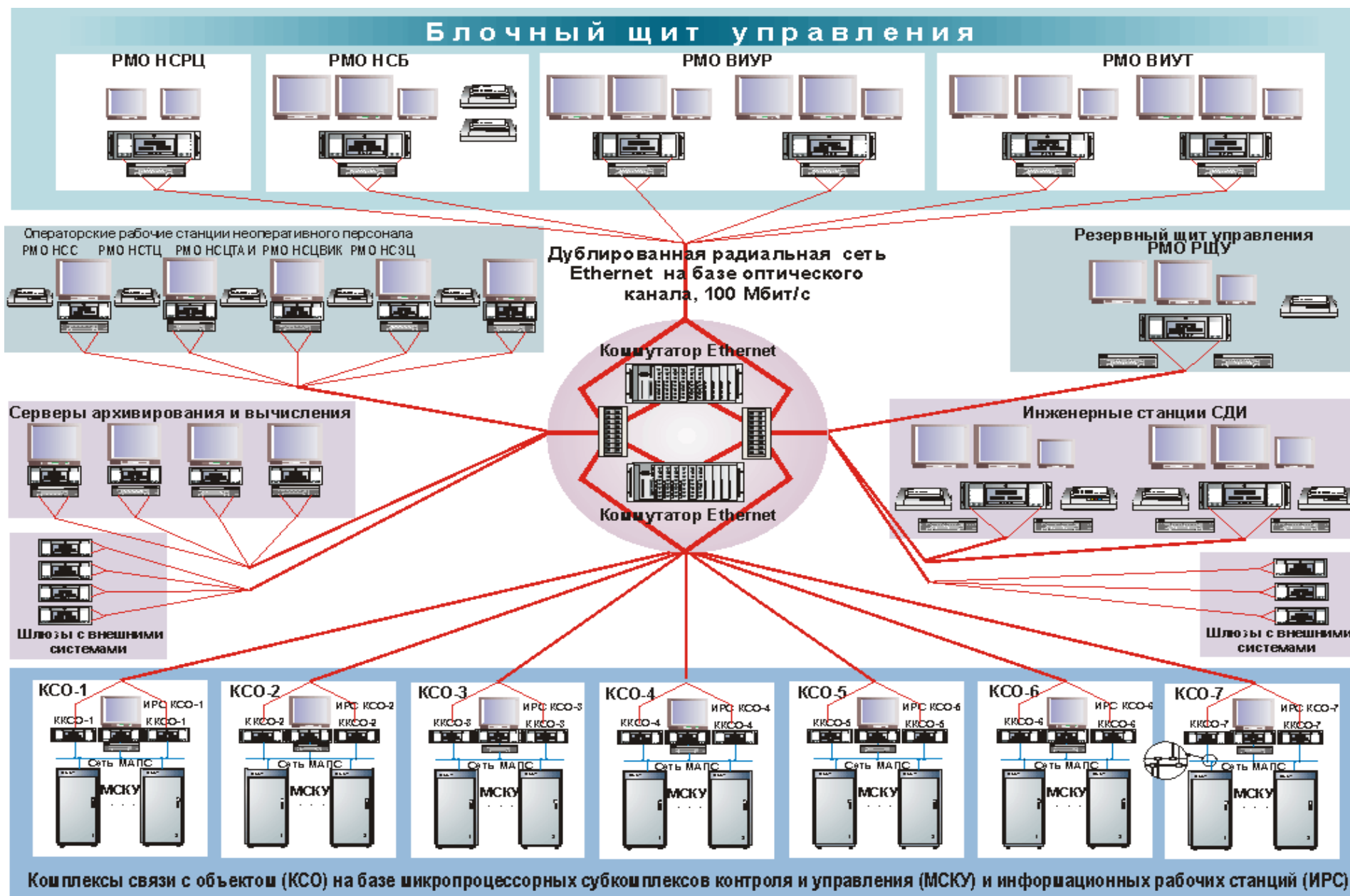


Рис. 4.5 Структурная схема комплекса технических средств ПТК ИВС

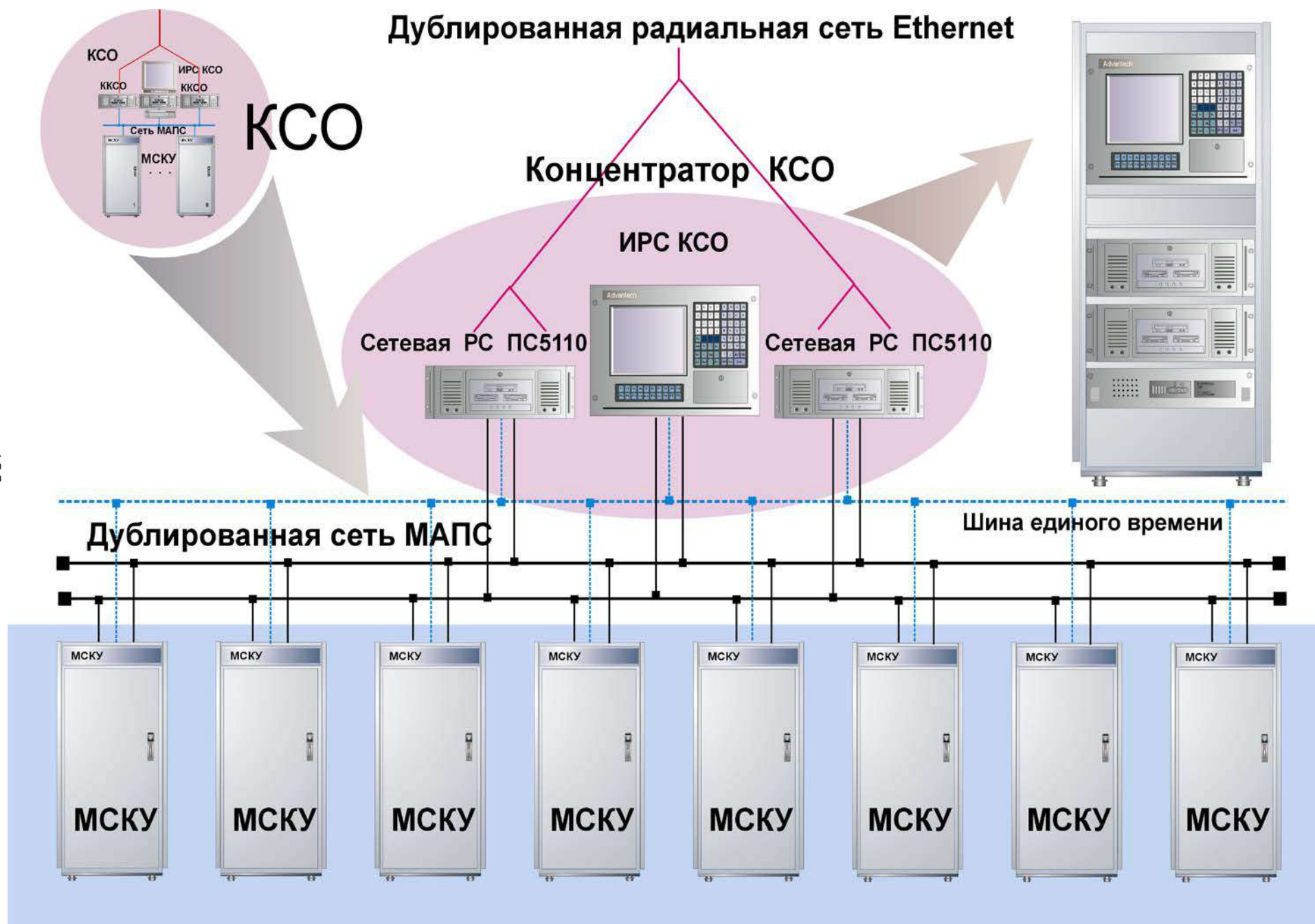


Рис. 4.6 Структурная схема комплекса связи с объектом (КСО) ПТК ИВС

Отказоустойчивость нижнего уровня ПТК. Повышение отказоустойчивости нижнего уровня ПТК может обеспечиваться двумя способами (и их комбинацией):

- резервированием нижнего уровня ПТК за счет дополнительных МСКУ;
- использованием резервированных (троированных) МСКУ.

Резервирование на уровне МСКУ является основным способом обеспечения отказоустойчивости ПТК на базе МСКУ М. Центральная часть МСКУ содержит три КМп, каждый из которых выходит на одну из трех магистралей интерфейса резервированного ИР. Межконтроллерные связи (реализованные через специальные порты) позволяют каждому КМп обращаться к двум другим КМп, обмениваться массивами данных, служебной информацией для взаимоконтроля и синхронизации. Подобная организация предполагает, что входные сигналы и другая информация поступают одновременно в три КМп, в которых выполняются копии одной и той же программы.

В МСКУ используются трехпортовые блоки связи с объектом, имеющие выход на три магистрали интерфейса ИР (Приложение А – Интерфейс резервированный). Оборудование БСО содержит как резервированные узлы (порты), так и нерезервированные (схемы преобразования и т.п.). Структура трехпортового БСО представлена на рис. 4.7. С точки зрения надежности оборудование БСО можно разделить на оборудование канала (входные/выходные преобразователи сигналов, фильтры и т.п.), оборудование связи с ИР (порты) и общее оборудование.

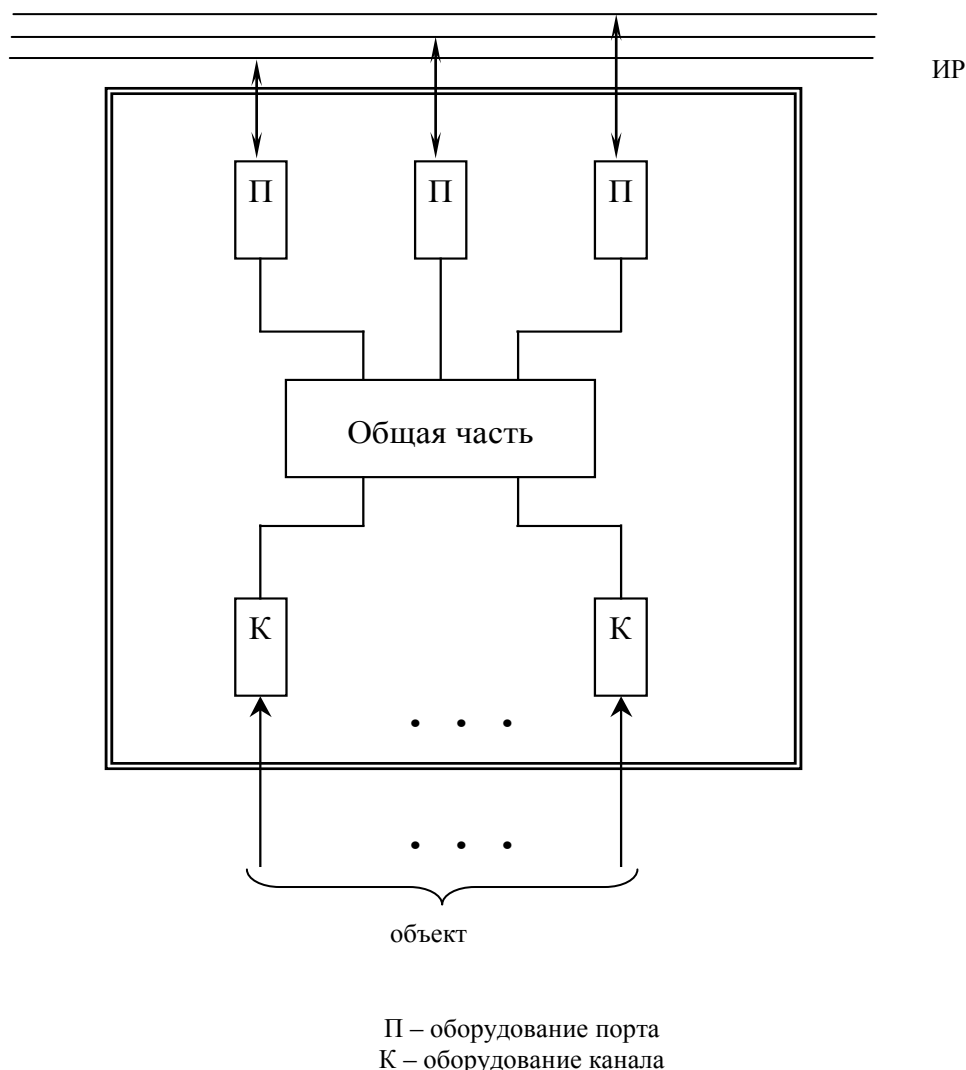


Рис. 4.7 - Структура трехпортового БСО

Отказ одного из портов БСО в резервированном МСКУ не является отказом канала ввода-вывода, так как обеспечивается его работоспособность за счет других портов и направлений интерфейса ИР.

Неисправность общей части БСО является отказом канала (группы каналов). Для предотвращения отказа канала резервирование каналов связи с объектом и ввод-вывод информации может производиться следующим образом:

- подключить резервирующие друг друга датчики или исполнительные механизмы к различным блокам связи с объектом, которые могут быть установлены как в одном, так и в разных МСКУ;

- объединить на кроссе резервирующие каналы в один канал и подключить к нему нерезервированный датчик или исполнительное устройство.

Способы организации резервирования каналов ввода аналоговых сигналов приведены на рисунках 4.8 – 4.10. При резервировании каналов приема сигналов термосопротивлений используются четыре переключающих контакта (рис. 4.10). Через нормально-замкнутые контакты реле подключается основной канал, при отказе его - производится переключение на резервный.

Способы организации резервирования каналов ввода-вывода дискретных сигналов приведены на рисунках 4.11 – 4.12. В нормальном состоянии сигнал с объекта вводится с основного канала, при его отказе происходит автоматическое переключение на резервный.

При резервировании каналов выдачи аналоговых и дискретных сигналов в каждый канал включается один или два переключающих контакта реле. В нормальном состоянии сигнал на объект выдается с основного канала, при его отказе происходит автоматическое переключение на резервный.

Возможны другие способы резервирования выходных дискретных сигналов (рисунки 4.13 – 4.15): непосредственное соединение их на кроссе параллельно (устойчивость к отказам типа "ложное выключение"), последовательное (устойчивость к отказам типа "ложное включение") или параллельно-последовательное (устойчивость к любому отказу). Последний способ требует больших аппаратных затрат, но обеспечивает устойчивость не только к отказам, но и к случайным сбоям.

Рассмотрим особенности функционирования троированного МСКУ. Все описываемые далее действия выполняются аппаратными средствами и исполнительной системой МСКУ. Сведения об исполнительной системе МСКУ приведены в п. 3.5.

Все решаемые задачи в контроллерах троированного МСКУ выполняются синхронно.

Исполнительная система МСКУ обеспечивает независимость задач пользователя от типа МСКУ (нерезервированный или резервированный) и его текущего состояния (количества работоспособных контроллеров). Исключение из процесса функционирования МСКУ неработоспособного КМп и включение его в процесс функционирования (после ремонта) обеспечивается исполнительной системой автоматически.

При выполнении операции восстановления автоматически обеспечивается приведение данных для программ восстанавливаемого контроллера в состояние, идентичное работающим контроллерам, и только после этого контроллер включается в нормальный режим функционирования.

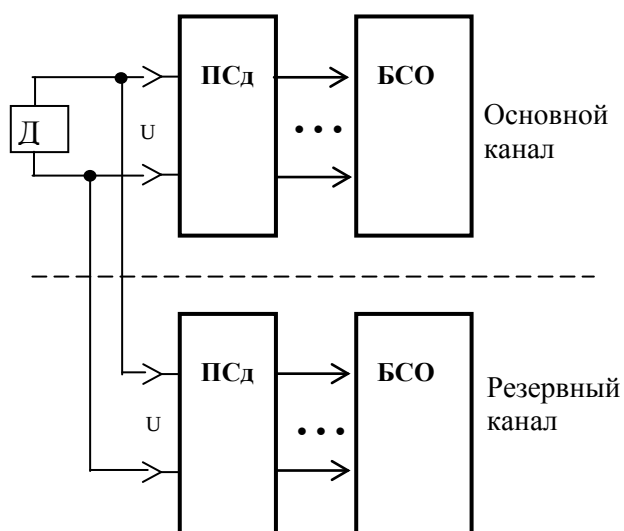


Рис. 4.8. Организация резервирования ввода аналоговых сигналов напряжения среднего уровня.

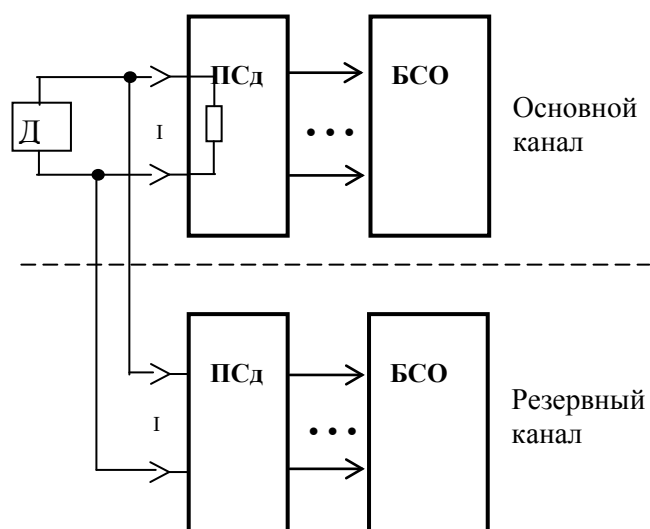


Рис. 4.9. Организация резервирования ввода аналоговых сигналов тока среднего уровня $\pm 5 \text{ mA}$, $\pm 20 \text{ mA}$.

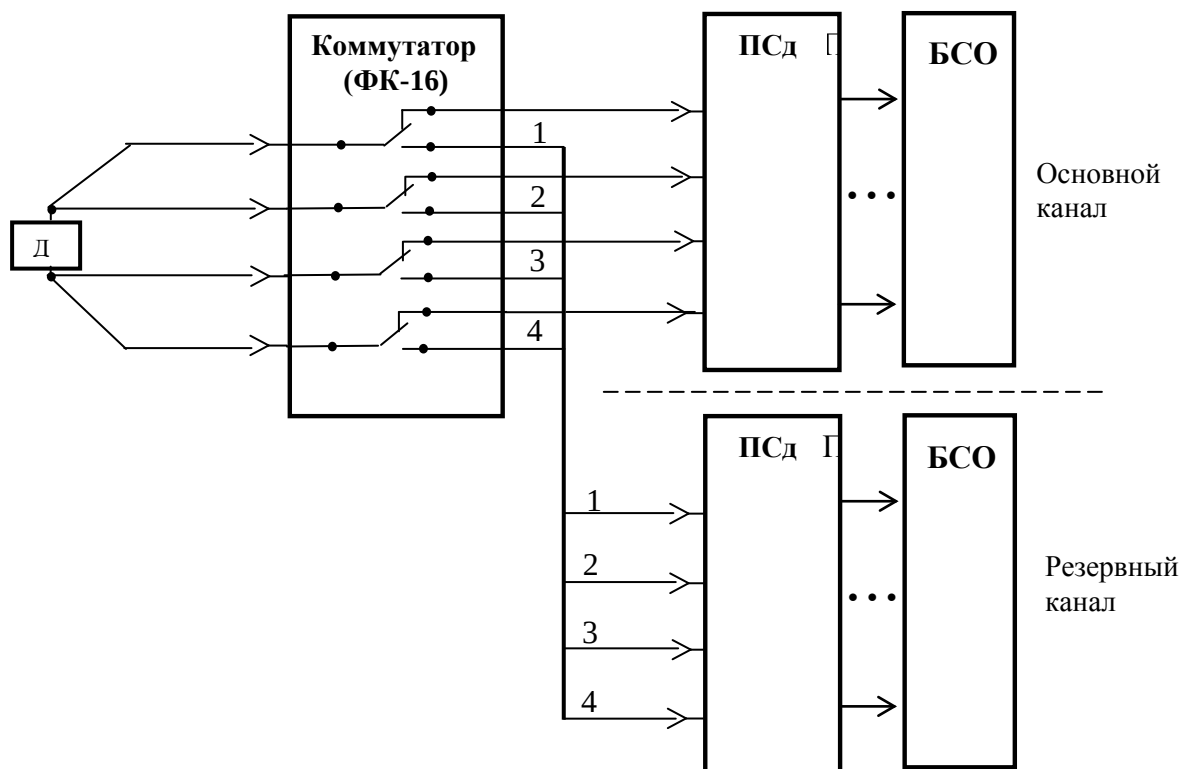


Рис. 4.10. Организация резервирования каналов ввода аналоговых сигналов сопротивления (ПТС-4/3, ПТС(В)-16/3).

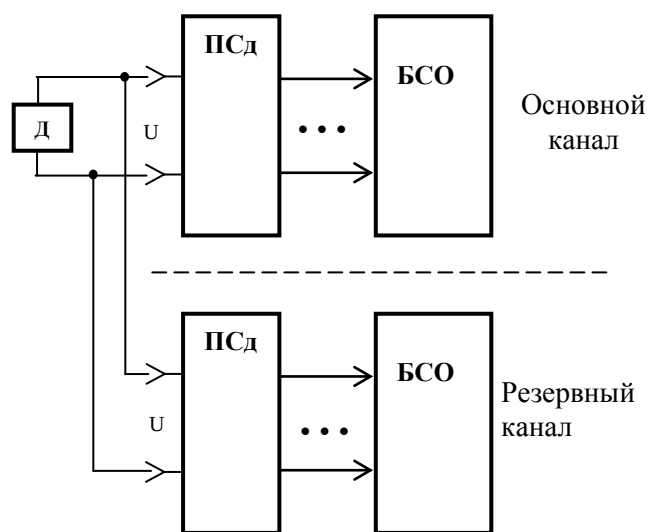


Рис.4.11. Организация резервирования каналов ввода дискретных сигналов напряжения типа «сухой контакт» (НД(РТ)-32/3, НД(РК)-16/3,

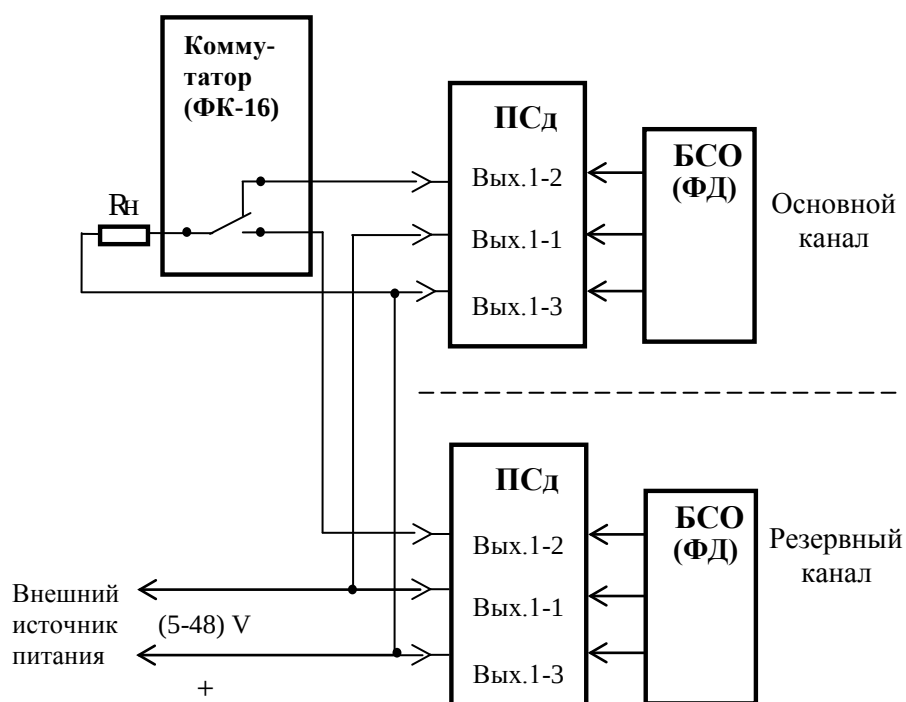


Рис. 4.12. Организация резервирования каналов ввода-вывода дискретных сигналов с переключением выходных каналов.

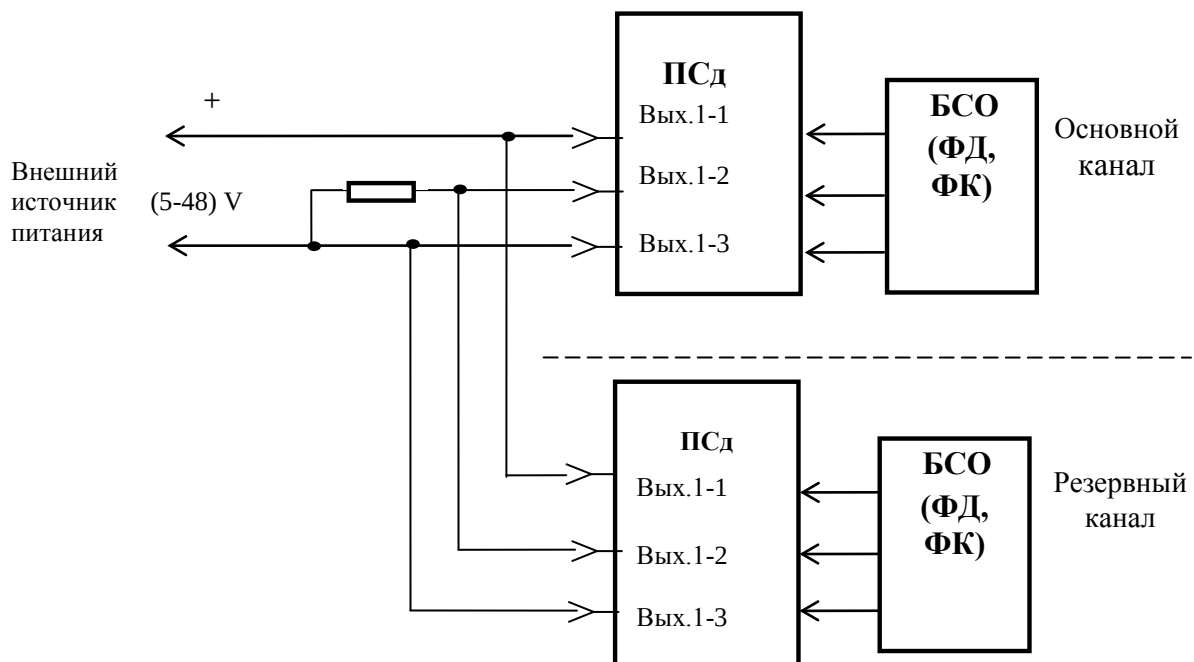


Рис. 4.13. Параллельное включение каналов выдачи дискретных сигналов.

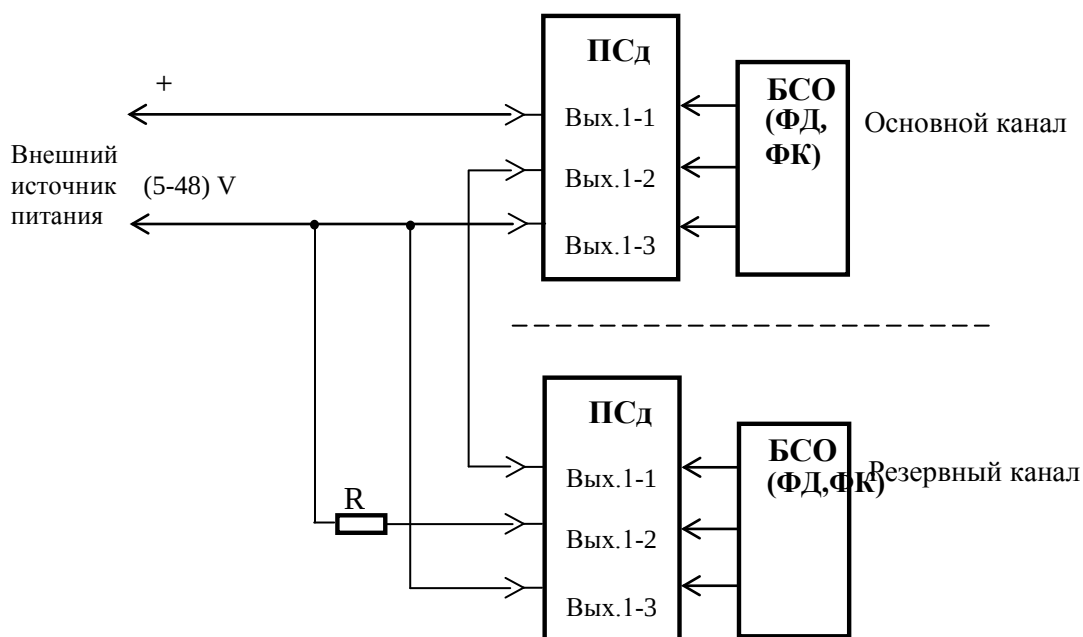


Рис.4.14. Последовательное включение каналов выдачи дискретных сигналов

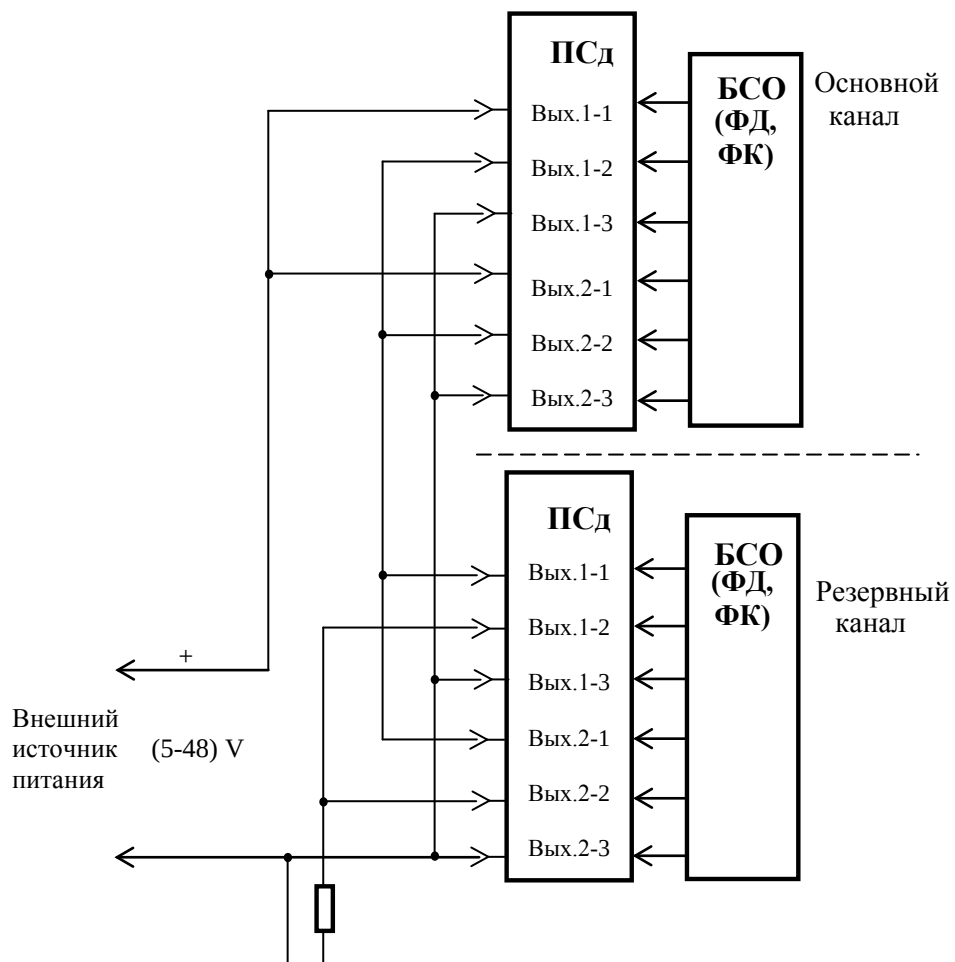


Рис. 4.15. Параллельно-последовательное включение каналов выдачи дискретных сигналов.

Операция ввода выполняется следующим образом. Первоначально исполнительная система МСКУ выполняет процедуру синхронизации - каждый КМп посылает другим КМп специальные сообщения по межконтроллерным связям и анализирует поступление аналогичных сообщений от них. В тот момент, когда КМп достигнет точки синхронизации, т.е. получит сообщения от других КМп, синхронизация состоялась. Если в течение установленного интервала времени не пришло сообщение хотя бы от одного КМп, фиксируется нарушение синхронизации работы и происходит процесс реконфигурации МСКУ – исключение из процесса функционирования МСКУ неработоспособного КМп.

Операция ввода выполняется параллельно всеми тремя контроллерами. При вводе дискретной информации все три массива должны быть равны, в противном случае это свидетельствует об отказе или случайном сбое в аппаратуре. Данные, принятые от аналоговых датчиков, могут различаться и при исправной аппаратуре (из-за погрешностей измерений). Принятые данные подвергаются так называемой «процедуре выравнивания».

«Процедура выравнивания» дискретной информации заключается в следующем. Дискретная информация представляет собой логические коды (одноразрядные двухпозиционные коды, упакованные в 16-разрядные слова). Каждый КМп сравнивает введенную им информацию с полученной от других КМп. Выравнивание осуществляется поразрядно. Выравненное значение выбирается в каждом разряде по мажоритарному принципу "два из трех". Фиксируются ситуации, в которых не все три операнда имеют одно и то же значение. В дублированном режиме работы МСКУ (когда один из КМп находится в состоянии отказа) в качестве выравненного значения может быть дизъюнкция или конъюнкция операндов, либо значение, полученное в предыдущем опросе. Конкретный вариант выравнивания для дублированного режима выбирается разработчиком АСУ ТП.

«Процедура выравнивания» аналоговой информации заключается в следующем. Результат преобразования аналоговой информации представляет собой 16-разрядное целое число. Все три результата преобразования аналоговой информации сравниваются между собой. Сравнение производится с точностью до максимально допустимой погрешности, задаваемой разработчиком АСУ ТП. Действительными являются данные, удовлетворяющие мажоритарному принципу «два из трех». Данные, отбракованные в результате сравнения, в дальнейшей операции выравнивания не участвуют, а для соответствующего канала фиксируется ошибка. Операция выравнивания может быть выполнена по одному из следующих правил (задается разработчиком АСУ ТП): выбор минимального значения, выбор максимального значения, вычисление среднего арифметического, выбор срединного (медианного) значения (остающегося после отбрасывания минимального и максимального значений). При отказе одного КМп (дублированный режим работы МСКУ) выравненное значение может быть получено одним из четырех способов: выбор минимального значения, выбор максимального значения, вычисление среднего арифметического, выбор ближайшего к предыдущему измеренному значению. Конкретный вариант выравнивания для дублированного режима выбирается разработчиком АСУ ТП. При отказе двух КМп за достоверное принимается единственное принятое значение. Каждое выравненное данное передается во все работоспособные КМп.

Выравнивание данных позволяет устранить последствия любых одиночных ошибок в принятых данных, благодаря чему гарантируется достоверность хода решения конечной задачи и конечных результатов. Возможность автоматического исправления ошибок является основным преимуществом троированного МСКУ по сравнению с нерезервированным.

Операция вывода выполняется следующим образом. Синхронизация перед операцией вывода выполняется также как и перед операцией ввода, перед выводом также может производиться процедура «выравнивания» данных (по заданию разработчика АСУ ТП). Рассмотрим особенности процедуры выравнивания на примере работы формирователя дискретных сигналов ФДР-16/3 (рисунок 4.16).

Сформированные в результате выполнения задачи 16-разрядные управляющие слова, подлежащие выдаче на объект, посылаются каждым КМп в регистры адресованного блока.

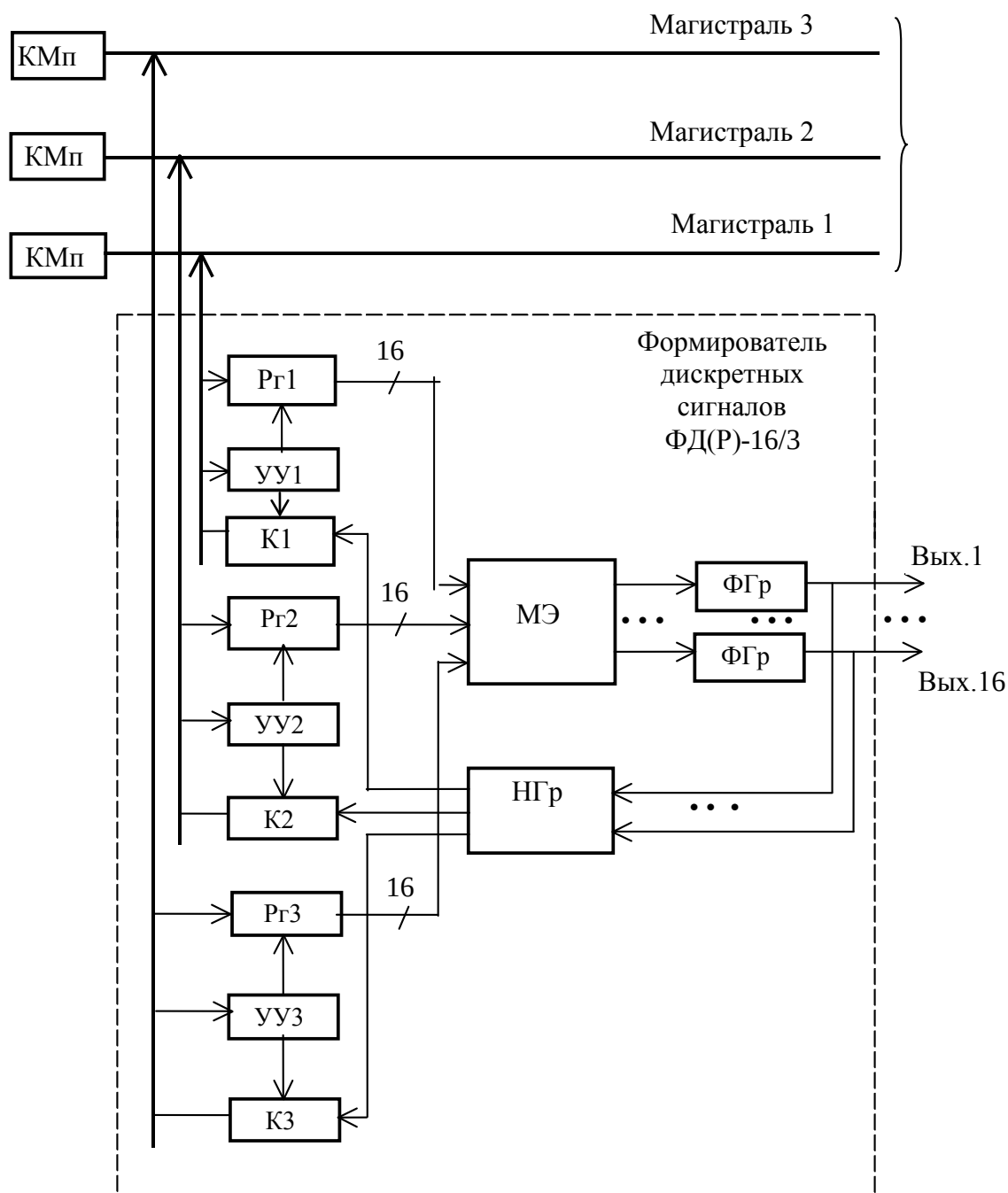


Рис. 4.16. Структурная схема формирователя дискретных сигналов ФДР-16/3.

Выходы регистров через мажоритарные элементы, выполняющие функции "два из трех", и схемы формирования сигнала подключены к выходам модуля. Таким образом, состояние каждого выхода в любой момент времени определяется значениями, сформированными, по крайней мере, двумя контроллерами. Для правильной работы мажоритарных элементов и выходных формирователей в модуле предусмотрена обратная связь для сравнения с выданными значениями.

Мажоритарные элементы выдают информацию на выход по мажоритарному принципу "два из трех". При отказе двух контроллеров на выходе устанавливается значение, принятое от единственного работоспособного контроллера и, если неработоспособны три контроллера, на выходе - предпочтительное значение (значение, заданное при проектировании блока, определяется разработчиком АСУ ТП).

В модулях вывода аналоговых сигналов алгоритмы работы сложнее. В троированном режиме они обеспечивают выдачу сигнала, соответствующего медианному (срединному) значению (остающегося после отбрасывания минимального и максимального значений), в дублированном режиме (при отказе одного из контроллеров) - выдачу сигнала из интервала между двумя значениями, в нерезервированном режиме выдачу единственного конкретного значения.

4.5 Контроль и диагностирование ПТК

Важнейшим элементом обеспечения отказоустойчивости является рациональная организация системы контроля и диагностирования ПТК. Средствами технического контроля и диагностирования ПТК решаются такие задачи, как:

- обеспечение высокого уровня достоверности функционирования и защита от выдачи неправильных воздействий на управляемый объект;
- своевременное обнаружение нарушений в работе ПТК, локализация неисправностей;
- организация рациональной эксплуатации таких систем, включая прогноз будущего состояния и обеспечение средствами восстановления работоспособности.

4.5.1. Организация процессов диагностирования в отказоустойчивых ПТК

С помощью средств диагностирования в ПТК реального времени, устойчивых к отказам, решаются задачи [61] функционального диагностирования с целью оперативного обнаружения и классификации ошибок, тестового диагностирования для поиска мест неисправностей с целью их устранения, создание условий для предотвращения распространения отказов по системе и снижения уровня деградации системы и контроля не только цифровых, но и аналоговых цепей.

Выполнение обычных диагностических функций в современных ПТК имеет особенности, связанные с тем, что эти функции реализуются для многокомпонентных систем. Для реконфигурации системы необходимо знать состояние всех компонентов системы и наметить новую структуру системы.

К организации функционального диагностирования ПТК предъявляются те же требования, что и для автономных машин - своевременность обнаружения ошибок и полнота охвата. В ПТК появляется необходимость защиты от ошибок, возникающих в связях между машинами.

В ПТК, устойчивых к отказам, средства диагностирования непосредственно связаны с системой реконфигурации, предназначенной для перестройки системы в случае обнаружения ошибки. Система реконфигурации получает информацию от средств диагностирования. Система реконфигурации реализует: переключение связей; действия, выполняемые операционной системой и действия, выполняемые оператором.

Выполнение этих действий состоит во внесении изменений в таблицы, определяющие текущую топологию системы.

В ПТК могут применяться известные аппаратные и программные средства проведения проверок и их различные модификации. К программным средствам организации проверок относятся:

- параллельный прогон задач на нескольких машинах с периодической проверкой результатов;
- измерение времени решения задачи на определенной машине (использование контрольного таймера);
- решение фоновых тестовых задач на машинах, свободных от выполнения основных функций;
- программно-логические методы организации контроля и другие.

К аппаратным средствам проведения проверок следует отнести введение дополнительных процессоров, выполняющих специальные функции диагностирования и реконфигурации и введение дополнительных связей для обеспечения процесса проверок.

Важным является тестирование операционных систем. Тестирование операционных систем, включая тестирование всех ее элементов, проводится специальными методами, которые рассчитаны на проверку работы системы прерывания, диспетчеризации, динамических характеристик. Тестирование прикладных программ также имеет важное значение, поскольку отказоустойчивость ПТК во многом определяется надежностью этих программ.

Используется два принципа организации тестирования программ - восходящее и нисходящее тестирование. Восходящее тестирование состоит в выполнении следующих этапов: тестирование модулей, подсистем, систем, приемосдаточные испытания. Тестирование модулей требует базы тестовых данных. При тестировании подсистем проверяются связи между модулями. На этапе тестирования систем проверяется система в целом. При нисходящем тестировании выделяется основная программа и один-два уровня подпрограмм в качестве ядра системы, затем проверяется само ядро системы. После установления работоспособности ядра эта система наращивается путем добавления по одному из новых модулей, которые проверяются отдельно.

4.5.2. Организация диагностирования ПТК на базе МСКУ М

Подсистема диагностирования, функционирующая в ПТК, включает как автономные программные средства (работающие на инженерных станциях ПТК), так и отдельные компоненты, находящиеся в составе основных программ, чем обеспечивается достаточная глубина и полнота диагностирования (до ТЭЗа и информационного канала) [62]. К основным функциям, выполняемым подсистемой диагностирования, относится контроль функционирования ПТК и отдельных его компонентов (рабочих станций, МСКУ, сетей), диагностирование программного обеспечения и технических средств и управление восстановлением при сбоях и отказах.

Рассмотрим организацию диагностирования ПТК на примере ПТК ИВС энергоблока АЭС с реактором ВВЭР-1000. Диагностирование ПТК ИВС осуществляется при помощи специальной функции «Диагностирование состояния технических и программных средств ИВС». Она включает следующие задачи:

- диагностирование состояния технических и программных средств МСКУ;
- диагностирование состояния технических и программных средств концентраторов КСО и РМО (на базе рабочих станций ПС 5110);
- диагностирование сетевых средств (МАПС, Ethernet);
- контроль сохранности файловой системы ИВС.

Диагностирование МСКУ заключается в обнаружении нарушений функционирования и формировании массива состояния, периодически передаваемого в концентраторы КСО. В концентраторах контролируется как факт получения массивов состояния от каждого МСКУ, так и состояние его технических и программных средств по данным, содержащимся в массиве состояния. Данные о состоянии всех МСКУ в преобразованном формате помещаются в локальную базу.

Диагностирование концентраторов КСО и РМО заключается в обнаружении нарушений функционирования устройств и периодическом формировании массивов состояния, передаваемых по сети Ethernet. Контролируется наличие массивов состояния, а данные о работоспособности записываются в общую базу данных ИВС. Диагностирование сетевых средств заключается в контроле обмена между абонентами сетей всех уровней информационными и диагностическими сообщениями. Глубина диагностирования технических средств – до типового элемента замены (ТЭЗа), программных средств – до выполняемого

процесса. Диагностирование выполняется с использованием принципов самодиагностирования компонентов ИВС.

Контроль сохранности файловой системы заключается в создании защищенного архива, содержащего характеристики версий основных компонентов программного обеспечения ИВС (исполняемых модулей, настроек, дисковых баз данных и т.п.) и проверке его соответствия текущим версиям, находящимся в рабочих директориях РМО. Режим выполнения перечисленных задач – периодический (с настраиваемым периодом) и/или эпизодически при запуске узлов, по требованию оператора.

Отображение результатов диагностирования заключается в представлении их с помощью общесистемных средств на специальных видеограммах РМО, СДИ как в целом по ИВС, так и по отдельным компонентам. Детальное диагностирование КСО, РМО и сетевых средств выполняется эпизодически по запросу оператора на инженерных станциях СДИ с возможностью вывода протоколов состояния. Программные средства инженерной станции обеспечивают вывод иерархических схем и меню, посредством которых можно от общего состояния ПТК перейти к анализу конкретных узлов. Регистрация событий по состоянию технических и программных средств (включение/отключение, нарушение/восстановление) состоит в ведении архива событий с последующим выводом отчетов.

Кроме описанных выше средств системного контроля и диагностирования существуют отдельные средства контроля и диагностирования компонентов ПТК, обеспечивающие быстрый поиск, замену и контроль отказавших узлов.

4.5.3. Контроль и диагностирование МСКУ

В МСКУ предусмотрены следующие виды контроля и диагностирования:

- проверка работоспособности в процессе функционирования (при реализации задач контроля и управления объектом), проверка производится в реальном масштабе времени в фоновом режиме;
- проверка работоспособности по контрольной задаче, выполняемой при обнаружении некорректных ситуаций в МСКУ;
- проверка работоспособности по контрольной задаче, выполняемой в процессе технического обслуживания, при ремонте, в процессе изготовления;
- контроль и поверка метрологических характеристик измерительных каналов.

Средства контроля и диагностирования МСКУ в процессе выполнения задач управления обнаруживают: некорректное функционирование КМп и блоков элементов; некорректный обмен информацией по интерфейсу ИР; расхождение между значениями одного и того же параметра, вычисленными разными КМп, и его выход за допустимую величину; расхождение между значением выданного сигнала и его действительной величиной, измеренной по обратной связи.

Диагностические тесты обнаруживают неисправность с точностью до одного сменного блока. В фоновом режиме выполняется задача взаимоконтроля КМп, формирование сигналов работоспособности КМп, программа калибровки и коррекции результатов преобразования аналоговых сигналов и программы контроля оборудования.

Проверка работоспособности МСКУ. Тестовый контроль МСКУ выполняется в процессе его штатного функционирования [63]. Он осуществляется в фоновом режиме задачей контроля оборудования, включенной в состав управляющей системы. В этом же режиме проводится проверка работы каналов приема и выдачи аналоговых и дискретных сигналов, определения метрологических характеристик с программной коррекцией результатов преобразования и измерения (т.е. калибровка измерительных каналов).

Результаты контроля запоминаются в оперативной базе данных МСКУ, которые могут быть прочитаны по команде оператора или по запросу программ пользователя, а также передаваться в виде инициативных сообщений всем абонентам МСКУ, указанным в программе генерации. Аппаратный контроль в МСКУ реализуется при помощи модуля контроля

оборудования МКО-1. Модуль контроля оборудования обеспечивает: контроль внутришкафного оборудования; ведение астрономического времени. Контроль оборудования МСКУ включает в себя контроль источников питания, блока вентиляторов, состояния дверей шкафа "открыто-закрыто" (используется для проверки санкционированного доступа), температуры внутри шкафа, контроль работоспособности измерительного тракта МКО-1 и контроль обмена информацией по интерфейсу ИР. Кроме этого модуль контролирует непрерывность поступления информации из сети единого времени и при отсутствии информации формирует в слове состояния признак "Нарушение".

Функция ведения астрономического времени реализована на двух функциональных узлах модуля, обеспечивающих: 1 - непрерывный ввод астрономического времени из сети единого времени (от синхронизатора) и 2 - автономные энергонезависимые часы реального времени модуля. Способ ведения времени настраивается через параметр, указываемый при конфигурации управляющей системы. Модуль обеспечивает ежесекундный ввод и преобразование кодовой последовательности от синхронизатора в параллельный код и ведет счет времени в пределах секунды. Если ежесекундная кодовая последовательность прекращает поступать на вход модуля, счет секунд, минут и часов продолжается, а в слове состояния формируется признак "Нарушение". При возобновлении поступления сигналов значения даты и времени автоматически приводятся в соответствии с поступающей кодовой последовательностью.

Для формирования сигнала "Работа МСКУ" модуль по каждому направлению интерфейса ИР принимает сигнал перезапуска сторожевого таймера (Watchdog) с интервалом не более 1 с. Сигнал "Работа МСКУ" снимается при отсутствии сигнала перезапуска Watchdog или всех трех сигналов РАБ интерфейса ИР.

Кроме этого модуль формирует сигнал "Состояние МСКУ", приводящий к смене цвета свечения индикатора на передней двери МСКУ с зеленого на красный при следующих нарушениях в работе МСКУ: отсутствуют сигналы РАБ на интерфейсе ИР; обнаружена неисправность оборудования (источников питания, блоков УСО, блока вентиляторов и т.п.); отсутствие сигнала "Работа" МСКУ; температура внутри шкафа МСКУ превышает 60 °С; установлен в слове состояния признак "Нарушение". Для организации контроля используется коммутатор напряжения. Контролируемые сигналы от источников питания через распределитель сигналов поступают непосредственно на коммутатор напряжения.

Сигналы контроля блоков вентиляторов и дверей шкафа представляют собой сигналы "сухого контакта". Питание этих цепей осуществляется от одного из источников питания МСКУ. Контрольные сигналы от блоков вентиляторов объединяются по схеме "проводного ИЛИ", контроль производится по обобщенному сигналу.

Контроль температуры внутри шкафа осуществляется при помощи встроенного термометра сопротивления. Напряжение аккумуляторной батареи подается непосредственно на коммутатор.

После преобразования в АЦП все контролируемые сигналы поступают в микропроцессорный контроллер для их обработки. При проверке информационных и управляющих цепей внутреннего интерфейса ИР контролируется обмен между блоками связи с объектом и контроллером. Данная проверка реализована аппаратными средствами.

Контроль работоспособности интерфейса ИР построен на использовании диагностических функций узла ИР. Устанавливая в регистре управления ИР признаки блокировки, выполняется проверка исходного состояния шин данных и управления, контролируется выдача сигнала "Работа" на шины ИР своего и соседних контроллеров. Данный вид контроля выполняется в каждом основном цикле процедуры ввода управляющей системы при наличии соответствующего параметра в программе генерации.

Работоспособность каналов связи с объектом проверяется при выполнении операций ввода-вывода, выравнивания и обработки данных. По результатам контроля формируется в оперативной базе МСКУ состояние канала (0 - работоспособен, 1 - неработоспособен во всех контроллерах), признак обрыва линии связи и признак нарушения метрологических

характеристик измерительного канала. Канал объявляется неработоспособным в *i*-ом контроллере при выполнении любого из следующих условий:

- *i*-й контроллер отсутствует в МСКУ или неработоспособен;
- неисправность шин ИР *i*-го контроллера;
- канал замаскирован;
- блок не адресуется по интерфейсу ИР;
- значение параметра в *i*-ом контроллере отбраковано при выравнивании;
- выход параметра за диапазон при масштабировании;
- сформирован признак обрыва линии связи.

Признаки обрыва линий связи и нарушение метрологических характеристик ПТС(В)-16, ПТС(П)-8, ПТП(В)-16, ПТП(П)-8, ПКИ-3 формируются программно, на основании анализа контрольных сигналов, вводимых от блока.

В троированных МСКУ периодически выполняется задача взаимоконтроля контроллеров. При этом каждый контроллер посылает другим контроллерам специальные запросы (так называемая «проверка пульса»). Не получив ответа через определенное время, он фиксирует неисправность соответствующего контроллера и формирует сообщение о неисправности. При обнаружении неисправности в троированных МСКУ операционная система МСКУ производит автоматическую реконфигурацию структуры МСКУ.

Проверка по контрольной задаче. Тестирование в режиме "контрольная задача" выполняется по вызову оператора при обнаружении ненормальных ситуаций, при проведении профилактических работ. Контрольная задача представляет собой набор перемещаемых тестов. Задача генерируется под конкретную конфигурацию МСКУ. Блоки подсистемы вывода аналоговых сигналов подвергаются проверке по контрольной задаче через систему обратной связи, ПСд при этом являются имитаторами нагрузки. Все операции, связанные с тестированием, выполняются с отключением МСКУ от объекта, кроме каналов ввода аналоговых сигналов, которые проверяются через систему калибровки. Каналы вывода аналоговых и дискретных сигналов проверяются по расхождению между выданным значением управляющего воздействия и величиной выходного сигнала, полученной по цепи обратной связи. При работе контрольной задачи необходимо использовать специальное контрольное оборудование.

Контрольная задача обеспечивает: однократную проверку указанного оператором оборудования; периодическую проверку заданного оборудования. Она включает следующие тесты: 1 - тест контроллера; 2 - тесты проверки каналов функциональных блоков связи с объектом; 3 - тест модуля связи МПСв16-1; 4 - тест контроля оборудования.

Контроль работоспособности контроллера предусматривает два уровня:

- 1) начальное тестирование при включении питания или после программного сброса;
- 2) периодическое тестирование в процессе эксплуатации и проведения профилактических работ.

При начальном тестировании выполняются следующие виды проверок: тестирование узлов контроллера: блока взаимодействия с сетью МАПС, блока взаимодействия с интерфейсом ИР, блока межконтроллерного взаимодействия; контроль целостности дискового пространства контроллера производится посредством: процедур чтения файлов с диска, подсчета текущей контрольной суммы файла и сравнения его с аналогом из паспорта диска.

Результаты тестирования помещаются в специально отведенную для этого область памяти, где создается протокол тестирования. При необходимости он может быть прочитан инструментальной ПЭВМ в процессе работы управляющей системы подготовки дискового пространства или инженерной панелью контроллера. При периодическом контроле работоспособности контроллера обеспечивается фиксация факта неработоспособности, локализация неисправности и выдача сообщений о неисправностях.

Контроль работоспособности контроллера осуществляется с помощью тестов в режиме "Диагностика КМп". В данном режиме выполняются следующие виды проверок:

- проверка узла взаимодействия с RS-232C;

- проверка узла взаимодействия с сетью МАПС;
- проверка узла взаимодействия с интерфейсом ИР;
- проверка межконтроллерного обмена.

Обеспечена возможность задания проверки контроллеров, как по полному списку тестов, так и выборочно. Во время тестирования контроллеров в режиме "Диагностика КМп" другие программы не выполняются и после завершения тестовых проверок необходима перезагрузка МСКУ.

Поверка метрологических характеристик. Поверка каналов ввода аналоговых сигналов и поиск неисправностей осуществляется в одном из следующих режимов: в режиме определения систематической, случайной и суммарной погрешности поверяемых каналов - "оценка МХ"; метрологической настройки измерительных каналов с использованием образцовых средств измерения (ПЗ20, ЦЗ1, магазины сопротивлений), подключенных по схемам, что и в режиме "оценка МХ" - "наладка"; метрологической настройки измерительных каналов с переходом программы в состояние "пауза" - "наладка (шаг)". Этот режим позволяет проверить все электрические цепи и обеспечивает удобный поиск неисправностей. При этом контрольные сигналы задаются вручную от калибратора ПЗ20 и магазинов сопротивлений.

Поверка каналов вывода аналоговых сигналов осуществляется по каждому поверяемому каналу путем поиска в диапазоне изменения выходного сигнала точки с наихудшим запасом точности (наибольшая погрешность). Измерение на выходах поверяемых каналов осуществляется образцовыми средствами измерения (цифровой вольтметр типа ЦЗ1).

Измерение динамических характеристик МСКУ. Измерение динамических характеристик технических средств, входящих в состав МСКУ, осуществляется по специальным программам, загружаемым в системную память контроллеров - режим "измерение характеристик". Операция измерения характеристик осуществляется при отключенных каналах от объекта, т.е. не выполняются задачи контроля и управления объектом. Программный комплекс измерения динамических характеристик технических средств МСКУ работает в следующих режимах: 1 - выполняется измерение временных характеристик контроллера арифметических операций (сложение, вычитание и т.д.); 2 - время опроса одного канала приема аналоговых сигналов и одной группы приема дискретных сигналов; 3 - время выдачи информации в канал формирования аналоговых сигналов и в одну группу формирования дискретных сигналов.

Измерение времени выполнения арифметических операций в МСКУ выполняется многократно, при этом фиксируется время перед первой операцией и после последней. Требуемое время вычисляется как разность времен конца и начала циклов, деленная на число повторений. После выполнения подсчета времени выдается соответствующее сообщение.

Измерение времени опроса одного канала (группы) приема в МСКУ выполняется после задания адреса блока и канала (группы) в блоке, многократный опрос заданного канала (группы) МСКУ, фиксируется время перед первой операцией и после последней. Требуемое время вычисляется как разность времен конца и начала циклов, деленная на число повторений. По завершении операции на экран монитора выдается сообщение «Время опроса одного канала xxx xxx мкс».

Измерение времени выдачи сигнала в канал (группу), формирование сигналов в МСКУ выполняется после задания адреса блока и канала (группы) в блоке, путем многократной выдачи данных, фиксируется время перед первой операцией и после последней. Требуемое время вычисляется как разность времен конца и начала циклов, деленная на число повторений. По завершении операции на экран монитора выдается сообщение «Время выдачи сигнала xxx xxx мкс».

Программа измерения времени блокировки цикла управления при восстановлении контроллера в резервированных МСКУ производит программный сброс одного из выбранных контроллеров, а после его восстановления формирует на экране монитора сообщение о времени блокировки цикла управления.

Время останова цикла управления объектом в процессе восстановления одного из направлений в МСКУ не должно превышать установленного значения.

Проверка датчиков и аналого-цифровых преобразователей. Как правило, в ПТК проверке подлежит полностью информационный канал, включающий датчик, линии связи, согласующие устройства, АЦП. Время наработки на отказ информационного канала весьма низкое. При контроле датчиков используются такие методы: мажорирование; сравнение с эталонным датчиком и сравнение с заранее рассчитанными уставками.

Метод мажорирования состоит в том, что для измерения некоторого параметра используют дополнительные датчики, а результат измерения получают путем присвоения измеренному параметру значения по значениям большинства датчиков. Мажорирование может осуществляться как за счет аппаратной реализации, так и за счет имеющегося запаса времени.

Метод сравнения с эталонным датчиком заключается в том, что измеренные значения параметра сравниваются со значением эталонного датчика (с некоторым допуском). По результатам сравнения принимается решение об исправности датчиков.

Способ сравнения с заранее заданными пределами предполагает проверку датчиков на относительно грубые неисправности и учитывает неисправности, при которых датчик вообще не выдает сигнал, либо выдает, но в несколько раз отличающийся от реального. Этот метод реализуется только программным путем.

Для контроля АЦП можно также использовать мажорирование либо дублирование. Одним из методов контроля является использование эталонного канала, суть которого состоит в подключении к АЦП регулируемого эталонного источника напряжения, которому соответствуют в памяти машины цифровые коды. Другой метод состоит в том, что сигнал из одного ЦАП, являющегося контрольным, подается на вход АЦП. В этом случае появляется возможность программно управлять получением аналогового сигнала на выходе ЦАП и формировать эталонный тестовый сигнал для проверки АЦП.

Организация отладки ПТК. Правильная организация отладки ПТК значительно уменьшает вероятность ошибок проектирования и производства. Она имеет много общего с организацией процессов диагностирования технических систем. На ранних стадиях создания системы нужно решить ряд вопросов, таких как подготовка и имитация входных воздействий, выбор точек и способов подачи входных воздействий, выбор способов съема выходных реакций. Обычно используют отладку по функциям ПТК. Предварительно выполняется отладка комплекса технических средств. С учетом сложности процесса отладки его организуют по принципу "от простого к сложному", т.е. путем постепенной отладки и наращивания состава взаимодействующих средств и решаемых задач. Выделяют два основных этапа отладки: автономный и комплексный. При автономной отладке устраняются внутренние ошибки каждого компонента системы. Уровень отлаженности, определяемый как отношение числа ошибок, выявленных при автономной и комплексной отладках, составляет 0,3 - 0,6.

При комплексной отладке проводится проверка взаимодействия программно-технического комплекса и прикладного программного обеспечения при выполнении основных функций ПТК. Уровень отлаженности составляет 0,8 - 0,9.

Наиболее типичные ошибки, которые встречаются во время отладки, делятся на три группы:

- алгоритмические, связанные с некорректной постановкой задач контроля и управления, отсутствием учета второстепенных факторов;
- системные, проявляющиеся в некорректном взаимодействии между отдельными устройствами и подсистемами в составе системы;
- схемные и программные, зависящие от уровня организации работ, квалификации исполнителей, уровня технологии и надежности технических средств.

Важнейшей задачей ранних этапов является анализ технического задания на полноту, непротиворечивость. Такой анализ снижает долю ошибок в системе.

Подобный анализ выполняется с помощью математического моделирования функционирования ПТК, в частности, с использованием имитационных моделей. При непосредственной отладке ПТК используются физические и имитационные модели, либо их

различные сочетания. При комплексной отладке систем управления используются аппаратные имитаторы внешних условий, заменяющих реальный объект.

Для отладки задач пользователя и отработки алгоритмов контроля и управления в составе сервисного оборудования МСКУ М имеются имитаторы различного функционального назначения. Данные имитаторы обеспечивают формирование различных типов сигналов датчиков и нагрузок, имитирующих реальный объект.

Вопросы

1. Что понимается под отказоустойчивостью вычислительной системы?
2. Какие существуют способы обеспечения отказоустойчивости ПТК?
3. В чем сущность структурного резервирования?
4. Что является основными средствами обнаружения нарушений в работе ПТК?
5. Каковы функции диагностирования?
6. Для чего необходима реконфигурация ПТК?
7. Какие существуют уровни восстановления?
8. В чем состоят особенности ПО отказоустойчивых ПТК?
9. Какие методы применяются для обеспечения устойчивости ПО к ошибкам?
10. В чем состоит сущность программной избыточности?
11. Каково существо метода изоляции ошибок?
12. Какие источники ошибок могут быть в разрабатываемом ПО?
13. Какие существуют пути уменьшения числа проектных ошибок в ПО?
14. Какими средствами обеспечена отказоустойчивость в системе ESS?
15. Какими средствами обеспечена отказоустойчивость системы реального времени August 300?
16. Какими методами обеспечена отказоустойчивость системы Stratus 32?
17. В чем заключается принцип обеспечения отказоустойчивости ПТК с дублированной структурой?
18. В чем заключается принцип обеспечения отказоустойчивости ПТК с троированной структурой?
19. Какими средствами обеспечивается отказоустойчивость ПТК на базе МСКУ М?
20. Какими методами обеспечено повышение отказоустойчивости локальных сетей МСКУ М?
21. Как обеспечивается повышение отказоустойчивости верхнего уровня ПТК?
22. Как обеспечивается повышение отказоустойчивости нижнего уровня ПТК?
23. Как реализовано резервирование МСКУ?
24. Как организовано резервирование каналов ввода аналоговых сигналов в МСКУ?
25. Как организовано резервирование каналов выдачи аналоговых и дискретных сигналов в МСКУ?
26. В чем состоят особенности функционирования троированного МСКУ?
27. Как выполняется операция ввода в троированном МСКУ?
28. Как выполняется операция вывода в троированном МСКУ?
29. Для чего необходимы средства технического контроля и диагностирования ПТК?
30. В чем особенности средств диагностирования ПТК реального времени, устойчивых к отказам?
31. В чем состоят функции системы реконфигурации ПТК?
32. Какие существуют принципы организации тестирования программ?
33. Какие основные функции выполняются подсистемой диагностирования ПТК?
34. Как организовано диагностирование ПТК ИВС энергоблока АЭС с реактором ВВЭР-1000?
35. Какие виды контроля и диагностирования предусмотрены в МСКУ?
36. Как выполняется проверка работоспособности МСКУ?

- 37.Как выполняется проверка МСКУ по контрольной задаче?
- 38.В чем заключается поверка метрологических характеристик МСКУ?
- 39.Какие методы используются при контроле датчиков и АЦП?
- 40.В чем заключается отладка ПТК?

ОСНОВНЫЕ МЕТОДЫ ПРОЕКТИРОВАНИЯ И ИССЛЕДОВАНИЯ ПТК

5.1 Основные методы проектирования ПТК

Наиболее важными этапами создания ПТК с хорошими системными характеристиками являются этапы системотехнического и аппаратурно-технического синтеза ПТК. На этих этапах проводится сравнительный анализ различных вариантов реализации информационного, программного и технического обеспечения ПТК. Современные ПТК относятся к классу сложных систем. Научно-методической основой разработки ПТК, как сложной системы, являются методы системного анализа и синтеза. Применением методов оптимизации сложных систем обеспечивают оптимальные (или близкие к оптимальным) характеристики, как всего ПТК в целом, так и отдельных его подсистем и компонентов. Одним из путей повышения качества разработки ПТК является *формализация методов анализа и проектирования систем*, основанных на определенной методологии. У каждой методологии есть свои положительные и отрицательные аспекты.

Процесс проектирования ПТК представляет собой последовательность этапов анализа и синтеза, взаимосвязанных между собой. Синтез сложных систем заключается в определении структуры синтезируемой системы и процессов ее функционирования, реализующих заданную совокупность функций системы совокупностью элементов ее будущей структуры. До недавнего времени одним из методов синтеза сложных систем являлся [64] функционально-структурный подход (ФСП).

Характерными особенностями ФСП являются: целостный подход к анализу и синтезу многоуровневых систем; учет взаимосвязи функций со структурой объектов при определяющей роли функции по отношению к структуре; учет информационных связей, как между элементами системы, так и системы с внешней средой. Одним из основных понятий ФСП является функционально-структурная организация системы (ФСО). Она представляет собой модель на основе функциональной и структурной организации, отражающей взаимодействие системы с окружающей средой и внутренние взаимосвязи элементов в процессе функционирования системы. Основу описания ФСО составляет дерево функций системы, адекватное по своей структуре иерархической организации систем. Дерево функций системы является основой для синтеза структуры системы на базе функциональных модулей. Функциональный модуль - функционально законченный объект, реализующий конечный набор функций определенного уровня и рассматриваемый безотносительно к средствам реализации.

Функции, реализуемые системой, делятся на основные и дополнительные. Под основными понимается набор функциональных операторов, определяемый целевым назначением системы ее проблемной ориентацией. Дополнительные функции отражают сервис системы, возможность расширения сферы системы. В многоуровневых системах реализуется следующая совокупность операторов: обработки, хранения, управления и передачи и обмена. Задача построения функционально-структурной схемы системы основывается на эквивалентных преобразованиях на уровне операторов и функциональных модулей. ПТК на основе функционально-структурного подхода рассматривается как иерархическая совокупность программно-аппаратных средств.

Одним из разновидностей ФСП является эволюционный синтез систем (ЭСС). Стратегия ЭСС сводится к последовательным формированием и преобразованиям ряда моделей системы, соответствующих различным уровням ФСО. С позиций ЭСС современные системы должны создаваться в минимальные сроки при максимальном использовании унифицированных компонентов. Методология ФСП предоставляла в распоряжение разработчиков строгие формализованные методы описания принимаемых технических решений. Они основаны на

наглядной графической технике: для описания различного рода моделей системы используются схемы и диаграммы. ФСП позволял разработчикам и будущим пользователям системы с самого начала неформально участвовать в ее создании, обсуждать и закреплять понимание основных технических решений. Однако при ручной разработке сложных систем обычно возникали следующие проблемы:

- неадекватная спецификация требований;
- неспособность обнаруживать ошибки в проектных решениях;
- низкое качество документации, снижающее эксплуатационные качества;
- затяжной цикл и неудовлетворительные результаты тестирования.

На практике ФСП использовался для синтеза ПТК с небольшим уровнем детализации.

Логическое представление процессов в ПТК обеспечивается посредством *технологии программирования*. Методы технологии программирования включают *структурное программирование, структурное проектирование и структурный анализ*. Эти методы необходимы для создания сложного программного обеспечения со сложной логикой. Методы информационной технологии позволяют упростить задачу автоматизации управления за счет рациональной организации баз данных. Основной предпосылкой информационной технологии является то, что в центре АСУ ТП находятся данные. Данные хранятся и ведутся с помощью различных видов программного обеспечения систем данных. Тенденции развития современных информационных технологий приводят к постоянному возрастанию сложности программного обеспечения ПТК. Проекты ПТК характеризуются, как правило, следующими особенностями:

- сложность описания (достаточно большое количество функций, процессов, элементов данных и сложные взаимосвязи между ними), требующая тщательного моделирования и анализа данных и процессов;
- наличие совокупности тесно взаимодействующих компонентов (подсистем), имеющих свои локальные задачи и цели функционирования, например, традиционные задачи, связанные с обработкой и отображением информации об объекте, различные регламентные задачи с детерминированными потоками запросов к базам данных и приложения аналитической обработки (системы поддержки оператора, экспертные системы) с недетерминированными потоками запросов;
- отсутствие прямых аналогов, ограничивающее возможность использования каких-либо типовых проектных решений и прикладных систем;
- необходимость интеграции существующих и вновь разрабатываемых приложений;
- функционирование в неоднородной среде на нескольких типах аппаратных средств;
- существенная временная протяженность проекта, обусловленная различной степенью готовности отдельных подсистем к модернизации.

Для успешной реализации проекта он должен быть, прежде всего, адекватно описан, должны быть построены полные функциональные и временные модели. Опыт проектирования ПТК показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов.

5.1.1 Методологии и технологии проектирования ПТК

Методы проектирования ПТК реализуются через конкретные технологии и поддерживающие их стандарты, методики и инструментальные средства. Технология проектирования определяется как совокупность трех составляющих: 1 - пошаговой процедуры, определяющей последовательность технологических операций проектирования (рис.5.1); 2 - критериев и правил, используемых для оценки результатов выполнения технологических операций; 3 - нотаций (графических и текстовых средств), используемых для описания проектируемой системы.

Технологические инструкции, составляющие основное содержание технологии, должны состоять из описания последовательности технологических операций, условий, в зависимости от которых выполняется та или иная операция, и описаний самих операций (рис. 5.1).

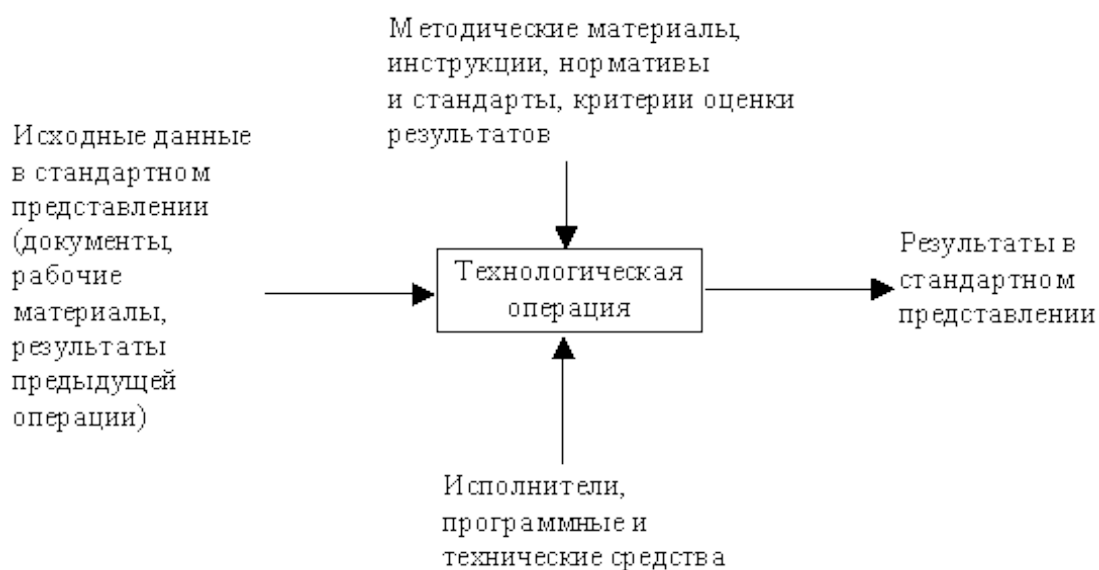


Рис. 5.1. Представление технологической операции проектирования.

Опыт разработки крупных ПТК (класса ПТК ИВС энергоблока ВВЭР-1000) показывает, что для повышения эффективности работ необходимо разбить проект на отдельные слабо связанные по данным и функциям подсистемы. Реализация подсистем должна выполняться отдельными группами специалистов. При этом необходимо обеспечить координацию ведения общего проекта и исключить дублирование результатов работ каждой проектной группы, которое может возникнуть в силу наличия общих данных и функций. Технология проектирования и разработки ПТК должна удовлетворять следующим общим требованиям:

- поддерживать полный жизненный цикл программного обеспечения;
- обеспечивать гарантированное достижение целей разработки ПТК с заданным качеством и в установленное время;
- обеспечивать возможность выполнения крупных проектов в виде подсистем (т.е. возможность декомпозиции проекта на составные части, разрабатываемые группами исполнителей ограниченной численности с последующей интеграцией составных частей);
- обеспечивать возможность ведения работ по проектированию отдельных подсистем небольшими группами (3-7 человек). Это обусловлено принципами управляемости коллектива и повышения производительности за счет минимизации числа внешних связей;
- обеспечивать минимальное время получения работоспособного ПТК;
- предусматривать возможность управления конфигурацией проекта, ведения версий проекта и его составляющих, возможность автоматического выпуска проектной документации и синхронизацию ее версий с версиями проекта;
- должна быть поддержана комплексом согласованных CASE-средств, обеспечивающих автоматизацию процессов, выполняемых на всех стадиях жизни ПТК.

Реализация ПТК в целом в короткие сроки может потребовать привлечения большого числа разработчиков, при этом эффект может оказаться ниже, чем при реализации в более короткие сроки отдельных подсистем меньшим числом разработчиков. Практика показывает, что даже при наличии полностью заверщенного проекта, внедрение ПТК идет последовательно по отдельным подсистемам. Реальное применение любой технологии проектирования, разработки и сопровождения ПТК невозможно без выработки ряда стандартов (правил, соглашений), которые должны соблюдаться всеми участниками проекта. К таким стандартам относятся следующие: стандарт проектирования; стандарт оформления проектной документации и стандарт пользовательского интерфейса.

Стандарт проектирования должен устанавливать:

- набор необходимых моделей на каждой стадии проектирования и степень их детализации;
- правила фиксации проектных решений, в том числе: правила именования объектов (включая соглашения по терминологии), набор атрибутов для всех объектов и правила их заполнения на каждой стадии и т. д.;
- требования к конфигурации рабочих мест разработчиков, включая настройки операционной системы, настройки CASE-средств, общие настройки проекта и т. д.;
- механизм обеспечения совместной работы над проектом, в том числе: правила интеграции подсистем проекта, правила поддержания проекта в одинаковом для всех разработчиков состоянии (регламент обмена проектной информацией, механизм фиксации общих объектов и т.д.), правила проверки проектных решений на непротиворечивость и т. д.

Стандарт оформления проектной документации должен устанавливать:

- комплектность, состав и структуру документации на каждой стадии проектирования;
- требования к ее оформлению (включая требования к содержанию разделов, подразделов, пунктов, таблиц и т.д.),
- правила подготовки, рассмотрения, согласования и утверждения документации с указанием предельных сроков для каждой стадии;
- требования к настройке издательской системы, используемой в качестве встроенного средства подготовки документации;
- требования к настройке CASE-средств для обеспечения подготовки документации в соответствии с установленными требованиями.

Стандарт интерфейса пользователя должен устанавливать:

- правила оформления экранов (шрифты и цветовая палитра), состав и расположение окон и элементов управления;
- правила использования клавиатуры и мыши;
- правила оформления текстов помощи;
- перечень стандартных сообщений;
- правила обработки реакции пользователя.

Постоянный рост производительности компьютеров, позволивший автоматизировать большинство этапов проектирования, а также внедрение сетевых технологий привели к появлению CASE-технологии проектирования сложных ПТК. Под термином CASE-средства (CASE - Computer Aided Software Engineering) понимаются программные средства, поддерживающие процессы проектирования баз данных, прикладного ПО (приложений), тестирование, документирование, обеспечение качества, а также другие процессы [65]. CASE-средства вместе с системным ПО и техническими средствами образуют полную среду разработки программного и информационного обеспечения ПТК. CASE-технология представляет собой метод проектирования, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения системы и разрабатывать приложения. Большинство существующих CASE-средств основано на методологиях структурного (в основном) или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств [66, 67].

Современные средства программирования включают методы структурного и модульного программирования, языки проектирования, языки описания системных требований и спецификаций и т.д. Недостатком использования CASE-технологии является трудность ее использования в комплексе с другими подобными средствами. Это объясняется как различными подходами, поддерживаемыми различными средствами, так и проблемами передачи данных и управления от одного средства к другому; Широкому использованию CASE-средств пока мешает необходимость долгосрочных затрат на их эксплуатацию, быстрое моральное старение,

частое появление новых версий, а также постоянные затраты на обучение и повышение квалификации персонала.

Одним из базовых понятий методологии проектирования ПТК является понятие жизненного цикла ее программного обеспечения (ЖЦ ПО). ЖЦ ПО - это непрерывный процесс, который начинается с момента принятия решения о необходимости его создания и заканчивается в момент его полного изъятия из эксплуатации. Основным нормативным документом, регламентирующим ЖЦ ПО, является международный стандарт ISO/IEC 12207 [68] (ISO - International Organization of Standardization - Международная организация по стандартизации, IEC - International Electrotechnical Commission - Международная комиссия по электротехнике). Он определяет структуру ЖЦ, содержащую процессы, действия и задачи, которые должны быть выполнены во время создания ПО. Структура ЖЦ ПО по стандарту ISO/IEC 12207 базируется на трех группах процессов: 1 - основные процессы ЖЦ ПО (приобретение, поставка, разработка, эксплуатация, сопровождение); 2 - вспомогательные процессы, обеспечивающие выполнение основных процессов (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит, решение проблем); 3 - организационные процессы (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение самого ЖЦ, обучение).

Разработка включает в себя все работы по созданию ПО и его компонентов в соответствии с заданными требованиями, включая оформление проектной и эксплуатационной документации, подготовку материалов, необходимых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала и т.д. Разработка ПО включает в себя, как правило, анализ, проектирование и реализацию (программирование).

Эксплуатация включает в себя работы по внедрению компонентов ПО в эксплуатацию, в том числе конфигурирование базы данных и рабочих мест пользователей, обеспечение эксплуатационной документацией, проведение обучения персонала и т.д., и непосредственно эксплуатацию, в том числе локализацию проблем и устранение причин их возникновения, модификацию ПО в рамках установленного регламента, подготовку предложений по совершенствованию, развитию и модернизации системы.

Проектирование связано с вопросами планирования и организации работ, создания коллективов разработчиков и контролем за сроками и качеством выполняемых работ. Техническое и организационное обеспечение проекта включает выбор методов и инструментальных средств для реализации проекта, определение методов описания промежуточных состояний разработки, разработку методов и средств испытаний ПО, обучение персонала и т.п. Обеспечение качества проекта связано с проблемами верификации, проверки и тестирования ПО. Верификация - это процесс определения того, отвечает ли текущее состояние разработки, достигнутое на данном этапе, требованиям этого этапа. Проверка позволяет оценить соответствие параметров разработки с исходными требованиями. Проверка частично совпадает с тестированием, которое связано с идентификацией различий между действительными и ожидаемыми результатами и оценкой соответствия характеристик ПО исходным требованиям. В процессе реализации проекта важное место занимают вопросы идентификации, описания и контроля конфигурации отдельных компонентов и всей системы в целом.

5.1.2 Структурный подход к проектированию ПТК

Сущность структурного подхода к разработке ПТК заключается в его декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы "снизу-вверх" от отдельных задач ко всей

системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов. Все наиболее распространенные методологии структурного подхода базируются на ряде общих принципов [64, 66]. В качестве двух базовых принципов используются следующие: принцип декомпозиции - принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения и принцип иерархического упорядочивания - принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне. Выделение двух базовых принципов не означает, что остальные принципы являются второстепенными, поскольку игнорирование любого из них может привести к непредсказуемым последствиям (в том числе и к провалу всего проекта). Основными из этих принципов являются следующие: принцип абстрагирования - заключается в выделении существенных аспектов системы и отвлечения от несущественных; принцип формализации - заключается в необходимости строгого методического подхода к решению проблемы; принцип непротиворечивости - заключается в обоснованности и согласованности элементов и принцип структурирования данных - заключается в том, что данные должны быть структурированы и иерархически организованы.

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых являются следующие [69]: SADT (Structured Analysis and Design Technique) модели и соответствующие функциональные диаграммы; DFD (Data Flow Diagrams) диаграммы потоков данных; ERD (Entity-Relationship Diagrams) диаграммы "сущность-связь".

В качестве формального аппарата, с помощью которого реализованы модели, используются сети Петри [70, 71]. Сети Петри – это двудольный ориентированный граф с вершинами двух типов: позициями (местами), куда помещаются маркеры, сопоставленные транзактам, и переходами, инициирующими перемещения маркеров. Стрелками соединяются лишь вершины разного типа. Различают «классические» (канонические) сети Петри, для которых существуют эффективные методы анализа, и многочисленные расширения (модификации) сетей Петри.

На стадии проектирования системы модели расширяются, уточняются и дополняются диаграммами, отражающими структуру программного обеспечения: архитектуру ПО, структурные схемы программ и диаграммы экранных форм. Перечисленные модели в совокупности дают полное описание системы независимо от того, является ли она существующей или вновь разрабатываемой.

5.1.2.1. Пример синтеза структуры ПТК учета электрической энергии

Исходные данные для синтеза структуры комплекса учета электропотребления содержатся в следующих технических требованиях:

- Требования к функциям комплекса: 1 - комплекс должен обеспечить построение многоуровневых автоматизированных систем учета электропотребления; 2 - комплекс должен обеспечить для каждого из подключенных 64 счетчиков электроэнергии: прием и преобразование импульсных сигналов, соответствующих ГОСТ 26035-83 (т.е. амплитуда входного импульсного сигнала - от 6 до 20 *mA*; амплитуда паузы от 0 до 2 *mA*; длительность импульса - от 15 *ms* до 200 *ms*; максимальная частота - до 3 *Hz*, амплитуда остаточного тока - не более 1 *mA*); 3 - контроль величины импульса (попадание в диапазон); 4 - выявление импульсных помех (импульсов величиной менее 15 *ms*); 5 - измерение периода следования импульсных сигналов; 6 - формирование и хранение в базе данных числа импульсов, поступивших по каждому каналу за каждый текущий получас, в течение двух календарных месяцев; 7 - многотарифный расчет потребления электроэнергии по каждому счетчику или группе счетчиков для коммерческого учета; 8 - оперативный расчет текущей мощности электропотребления по каждому каналу, усредненной за 1,3,5,10,30 минут;

- По командам, формируемым с пульта управления комплекса, он должен обеспечивать расчет и вывод на индикацию или печатающее устройство следующих показателей потребления активной и реактивной электроэнергии и мощности по указанному каналу или группе учета за прошедший или текущий учетный период: 1 - общую потребленную электроэнергию; 2 - потребленную электроэнергию по тарифным зонам; 3 - потребленную электроэнергию по сменам работы предприятия; 4 - среднюю мощность; 5 - среднюю мощность по тарифным зонам; 6 - текущую мощность, усредненную за 1, 3, 5, 10, 30 минут; 7 - мгновенную мощность по каналу учета, определяемую по величине временного интервала между двумя смежными импульсами. С помощью пульта управления должна быть обеспечена возможность ввода в следующей настроечной информации: наименование каналов, групп учета; тип, заводской номер, емкость счетного механизма, коэффициент пересчета, начальные показания счетчика; коэффициенты трансформации трансформаторов тока и трансформаторов напряжения; границы учетных периодов, тарифных зон, смен предприятия; лимиты потребления электроэнергии: общий, по тарифным зонам; астрономическое время: год, месяц, число, часы, минуты, секунды; список паролей, определяющих право и уровень доступа должностных лиц к комплексу. На индикацию пульта управления должна выводиться следующая информация технического контроля: текущие показания счетных механизмов счетчиков (табло); текущий счет импульсов по каналу учета; количество и номера каналов, от которых прекратилось поступление счетных импульсов с указанием даты и времени;

- Комплекс должен иметь связь с верхним уровнем по коммутируемым или выделенным линиям телефонной сети общего пользования. По запросам от верхнего уровня комплекс должен обеспечить: передачу информации из базы данных по запросам от верхнего уровня, ввод исходных данных и формирование отчетных документов. Должен быть обеспечен контроль работоспособности технических средств и самодиагностика.

5.1.2.2. Синтез функциональной структуры технического обеспечения

Автоматизированная система учета и контроля электроэнергии, в состав которых должен входить КУЭП, является трехуровневой. Верхним - командным уровнем – может быть АРМ энергетика. АРМ энергетика инициирует работу комплекса учета - среднего уровня. Связь комплекса с нижним уровнем - со счетчиками - осуществляется радиальными двухпроводными линиями связи. В состав комплекса входят программные и аппаратные средства. Комплекс должен выполнять диалог с оператором, принимать и контролировать импульсы, поступающие от счетчиков, производить расчет необходимых параметров энергопотребления, обслуживать запросы верхнего уровня.

Исходя из многоуровневой организации системы, в которой функционирует комплекс, весь реализуемый набор функций можно разделить на пять групп: 1 - сопряжения с верхним уровнем; 2 - связи с оператором; 3 - связи со счетчиками; 4 - обработки информации в комплексе; 5 - сопряжения со стандартным печатающим устройством. Приняв эти функции в качестве базовых, можно произвести их декомпозицию в соответствии с методикой, изложенной в [64].

Дерево функций комплекса:

1 уровень: F_0 - автоматизированный учет потребления электроэнергии;

2 уровень: F_1 - функции сопряжения с верхним уровнем; F_2 - функции связи с оператором; F_3 - функции связи со счетчиками; F_4 - функции управления и обработки информации в комплексе; F_5 - функции сопряжения со стандартным печатающим устройством.
Дополнительные функции: F_6 - диагностика технических средств комплекса;

3 уровень:

F_1 $\left\{ \begin{array}{l} F_{11} - \text{прием инициативных сообщений верхнего уровня;} \\ F_{12} - \text{выполнение запросов верхнего уровня;} \\ F_{13} - \text{передача ответных сообщений на верхний уровень.} \end{array} \right.$

$$\begin{array}{lcl}
F_2 & \left\{ \begin{array}{l} F_{21} - \text{прием кодовой комбинации, поступившей с пульта управления;} \\ F_{22} - \text{обработка запроса оператора;} \\ F_{23} - \text{передача информации на индикатор;} \end{array} \right. \\
F_3 & \left\{ \begin{array}{l} F_{31} - \text{прием и контроль импульсных сигналов от 64 каналов;} \\ F_{32} - \text{измерение периода времени между соседними импульсами;} \\ F_{33} - \text{хранение результатов измерений;} \end{array} \right. \\
F_4 & \left\{ \begin{array}{l} F_{41} - \text{координация выполнения всех функций комплекса;} \\ F_{42} - \text{вычисление необходимых параметров энергопотребления;} \\ F_{43} - \text{хранение базы данных с полученными расчетами;} \end{array} \right. \\
F_5 & \left\{ \begin{array}{l} F_{51} - \text{передача информации на принтер через стандартный параллельный} \\ \text{интерфейс;} \\ F_{52} - \text{чтение и хранение слова состояния принтера;} \end{array} \right. \\
F_6 & \left\{ \begin{array}{l} F_{61} - \text{аппаратный контроль технических средств комплекса;} \\ F_{62} - \text{проверка функций комплекса контрольной задачей;} \\ F_{63} - \text{тестирование технических средств комплекса диагностическими} \\ \text{тестами в фоновом режиме.} \end{array} \right.
\end{array}$$

Дерево функций, построенное с учетом перечисленных требований, позволяет перейти к последующим этапам структурного синтеза. Многофункциональный комплекс, способный выполнять перечисленные функции, представляет собой сочетание многофункциональных и специализированных модулей. На основе сформированного дерева функций можно построить обобщенную операторную модель комплекса.

На рис. 5.2 приведена обобщенная операторная модель с учетом проведенной декомпозиции базовых функций и выделением типовых операторов. На операторной модели с учетом проведенной декомпозиции выделены функциональные модули высокого уровня: ФМ1 - модуль сопряжения с верхним уровнем; ФМ2 - модуль связи с оператором; ФМ3 - модуль связи с нижним уровнем; ФМ4 - модуль управления, вычислений и хранения результатов; ФМ5 - модуль связи с принтером; ФМ6 - модуль диагностики.

С учетом проведенного анализа функций комплекса в функциональной структуре комплекса могут быть выделены следующие компоненты:

- подсистема сопряжения с верхним уровнем (ПСВ);
- подсистема связи с оператором - пульт управления (ПУ);
- подсистема связи с нижним уровнем (ПСН);
- центральная подсистема (ЦП) - концентратор;
- подсистема связи с принтером (ПСП);
- подсистема диагностики (ПД).

Обобщенная операторная модель приведена с учётом декомпозиции базовых функций и выделением типовых Р, С, М, Т операторов. Операторная модель может быть использована для формирования структуры системы на основе функциональных модулей с учетом требований по быстродействию, элементной базы и т.п. Результатом последовательного выполнения этапов синтеза является структура комплекса на уровне реальных конструктивных модулей.

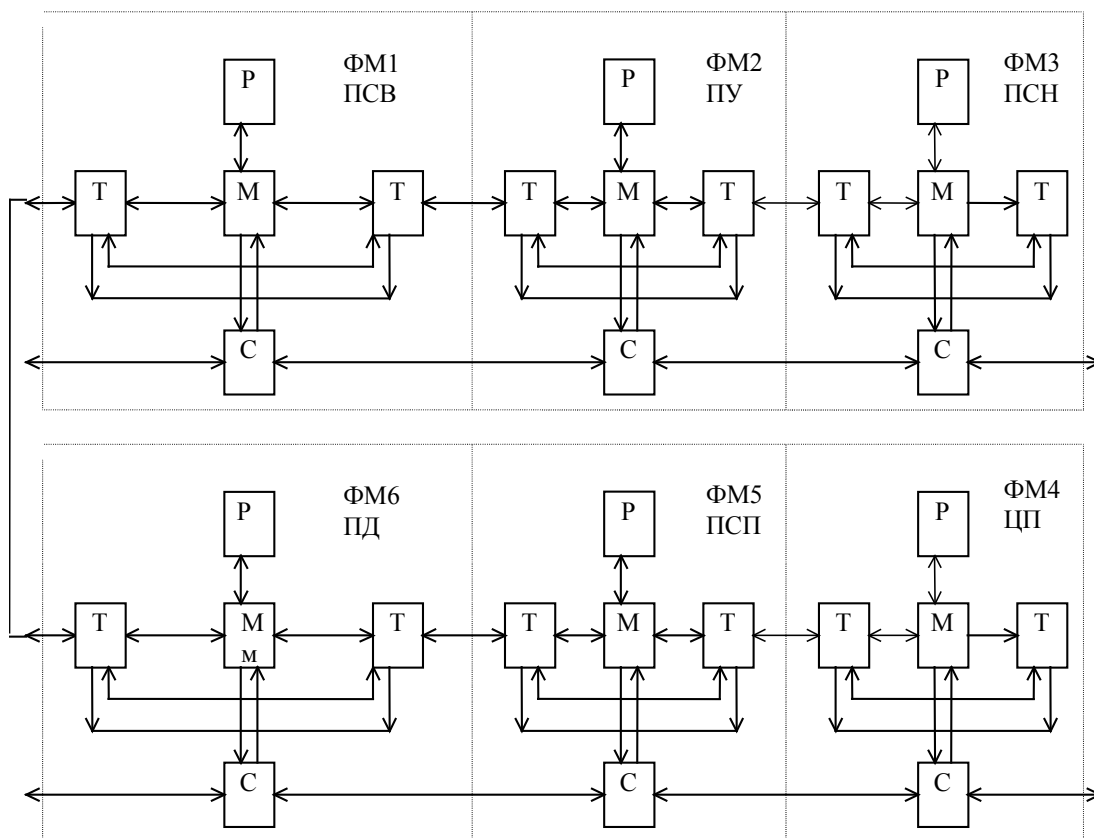


Рис. 5.2. Операторная модель технического обеспечения комплекса.

5.1.2.3. Синтез функциональной структуры программного обеспечения

Функциональная структура программного обеспечения может быть получена с применением методики, использованной для синтеза функциональной структуры технического обеспечения. Дерево функций комплекса отражает и функции программного обеспечения. На рис. 5.3 приведена обобщенная модель программного обеспечения комплекса с учетом декомпозиции базовых функций. На обобщенной модели с учетом проведенной декомпозиции выделены такие программные модули высокого уровня: ПМ1 - программный модуль сопряжения с верхним уровнем; ПМ2 - программный модуль связи с оператором; ПМ3 - программный модуль связи с нижним уровнем; ПМ4 - программный модуль вычислений и хранения результатов; ПМ5 - программный модуль связи с принтером; ПМ6 - программный модуль диагностики; ПМ7 - управляющая система комплекса.

На рисунке обозначено: Т - оператор передачи; Р - оператор обработки; М - оператор хранения; С - оператор управления.

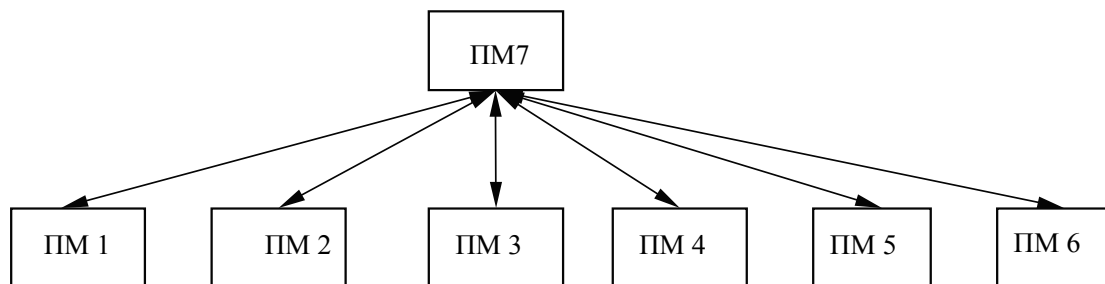


Рисунок 5.3 Обобщенная модель программного обеспечения комплекса

5.2 Основные методы исследования ПТК

Анализ ПТК необходим на этапах системотехнического (эскизного) и аппаратно-технического синтеза для оценки эффективности различных вариантов организации вычислительных процессов, выбора решений при проектировании информационного, программного, технического обеспечения. Методами анализа ПТК, как разновидности сложной системы, в основном, являются аналитические и имитационные методы моделирования. Широко распространенным аппаратом исследования ПТК является теория массового обслуживания, позволяющая провести моделирование их работы во времени [72]. Методы моделирования позволяют решить такие задачи анализа, как:

- определение производительности проектируемой системы и характеристик ее эффективности;
- определение загрузки оборудования, выявление узких мест в системе и ее балансировку;
- исследование эффективности алгоритмов работы программных модулей, управляющих работой системы;
- исследование поведения системы при изменении ее конфигурации и т.д.

Методы теории массового обслуживания основаны на представлении функционирования ПТК в виде соответствующей системы массового обслуживания (СМО), причем различают аналитические модели СМО и статистические модели СМО [73]. Аналитические методы исследования сводятся к построению математических моделей, в которых зависимость между характеристиками и параметрами объекта представляется совокупностью математических выражений (формул) - алгебраических, дифференциальных, интегральных и др. В аналитических моделях зависимость между характеристиками и параметрами может быть представлена в явной аналитической форме в виде выражений $Y_i = F_i(x_1, \dots, x_N)$, ($i = 1 - M$), где N - число параметров, M - число характеристик исследуемой системы.

Как правило, получение аналитических выражений связано с определенными допущениями и упрощениями в описании и поведении объектов: независимостью одних факторов от других, линейностью некоторых зависимостей и т.п. Во многих случаях допущения приводят к существенным отличиям модели от реального объекта, вследствие чего моделируемая зависимость существенно отличается от реальной и характеристики представляются с большой погрешностью. Аналитические методы дают хорошие результаты для относительно простых объектов. Использование аналитических методов для исследования таких сложных объектов, как современные ПТК, представляющие собой многоуровневые вычислительные сети, ограничено. Аналитические модели часто позволяют изучить лишь предельные значения характеристик, худшие или лучшие ситуации в системе. Современный математический аппарат не позволяет в полной мере решать такие вопросы как, например: формальный выбор дисциплин обслуживания абонентов, анализ функционирования сетей с адаптивными изменяемыми параметрами и др. Поэтому широко используются методы имитационного моделирования.

Методы имитационного моделирования позволяют исследовать системы любой степени сложности и, в принципе, не ограничивают уровень детализации в моделях. Особенности организации и функционирования, препятствующие использованию аналитических методов, без затруднений воспроизводятся в имитационных моделях. Недостатками имитационных методов являются большие затраты времени на моделирование и частный характер получаемых результатов. Число операций, обеспечивающее воспроизведение представленных интервалов функционирования системы, оказывается значительным и при моделировании систем умеренной сложности составляет миллиарды операций на одну реализацию модели. Чтобы определить зависимость между характеристиками и параметрами, необходимы многократные прогоны модели.

Несмотря на указанные недостатки, методы имитационного моделирования в силу их универсальности являются незаменимым инструментом при теоретических исследованиях и проектировании программных и технических средств ПТК.

Для анализа детерминированных систем возможно использование аппарата сетей Петри [70]. Есть попытки использования моделей на основе стохастических сетей Петри [74].

Вопросы

1. Что представляет собой процесс проектирования ПТК?
2. Что такое функционально-структурная организация системы?
3. Каким требованиям должна удовлетворять технология проектирования, разработки ПТК?
4. Что должен устанавливать стандарт проектирования?
5. Что должен устанавливать стандарт оформления проектной документации?
6. Что должен устанавливать стандарт интерфейса пользователя?
7. Что представляет собой CASE-технология?
8. Приведите пример синтеза структуры ПТК.
9. Какие основные методы исследования ПТК существуют?

ПРОГРАММНО-ТЕХНИЧЕСКИЕ КОМПЛЕКСЫ НА БАЗЕ МСКУ М

6.1 ПТК на объектах энергетики

6.1.1 ПТК АСУ ТП энергоблока 800 MW Запорожской ГРЭС

Совместная разработка Харьковского института комплексной автоматизации (ХИКА) и АО "Импульс", внедрена на энергоблоках Запорожской ГРЭС [10].

Основные функции системы: сбор и обработка информации от датчиков и автономных по отношению к АСУ ТП систем; отображение (контроль) расчетных параметров; регистрация информации в нормальном режиме и при аварийных ситуациях; обмен информацией с внешними системами, в том числе с общестанционной системой (АСУ ТЭС); расчет и анализ технико-экономических показателей оборудования энергоблока и неизмеряемых параметров; диагностика технологического оборудования, защита и блокировка; дистанционное и логическое управление; контроль процессов управления, автоматическое регулирование.

Структурная схема АСУ ТП энергоблока 800 MW приведена на рис. 6.1

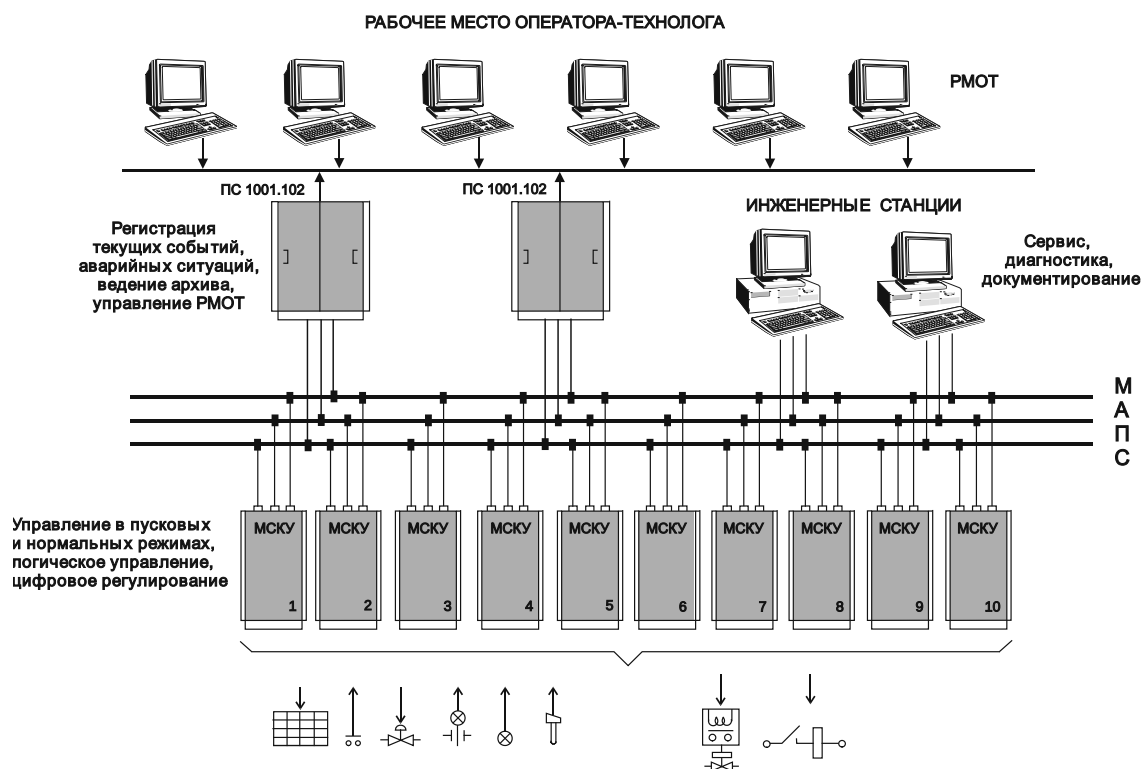


Рис. 6.1 Структурная схема ПТК АСУ ТП энергоблока 800 MW Запорожской ГРЭС.

Основные характеристики системы:

- число контролируемых параметров - до 12 тыс.;
- цикл обновления информации на экранах дисплеев - 1-2 с;

- реакция на изменение входных сигналов аналоговых/дискретных - 2-3/1-2 с;
- разрешающая способность фиксации защитных дискретных сигналов - 0,01 с;
- цикл расчета и выдачи команд по цифровому управлению для регуляторов/защит - 0,5/0,03 с.

6.1.2 Информационно-вычислительная система технологического контроля энергоблока Кольской АЭС с реактором ВВЭР-440

ИВС внедрена в 1993 г. и 1995 г. на действующих энергоблоках №1 и №2 Кольской АЭС вместо выработавшей свой ресурс системы информационного контроля [8]. Структурная схема АСУ ТП 1-го и 2-го энергоблоков показана на рис.6.2. Система разработана с учетом современных требований МАГАТЭ и нормативных документов по ядерной безопасности.

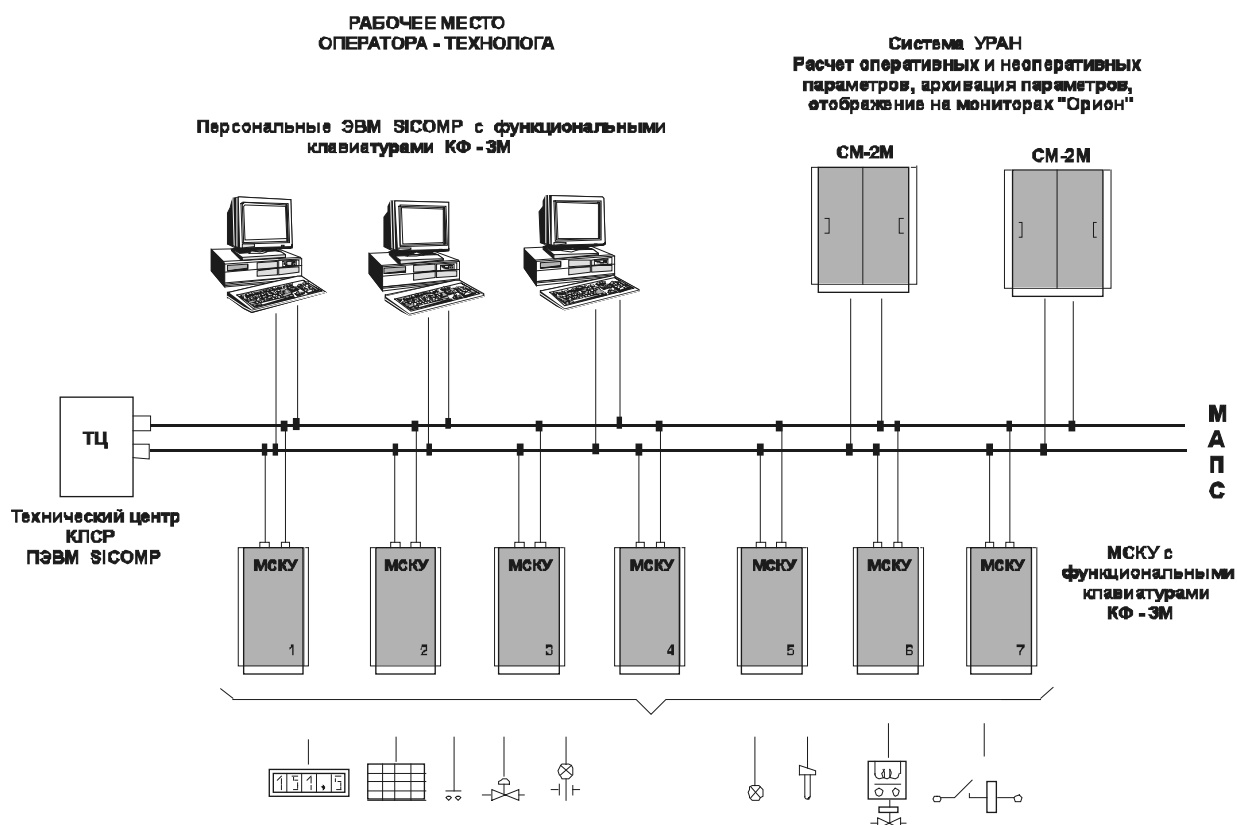


Рис. 6.2. Структурная схема ПТК АСУ ТП 1 и 2 энергоблоков Кольской АЭС.

Основные функции системы:

- оперативный сбор, обработка, сигнализация, отображение, регистрация и представление информации о состоянии технологического процесса в условиях нормальной эксплуатации для первого и второго контуров, а также систем безопасности энергоблока при нормальной эксплуатации и в аварийных режимах;
- передача информации в вышестоящую систему "Уран" (на базе УВК CM-2М) для расчета оперативных и неоперативных параметров, их архивации и отображения на мониторах;
- контроль работоспособности и диагностика собственных технических и программных средств.

Применение новой системы обеспечивает существенное повышение функциональных возможностей, в том числе прием и обработку 1000 аналоговых и 400 дискретных сигналов; регистрацию изменений аналоговых и дискретных сигналов с разрешающей способностью 0,1; 2 с; реакцию системы от момента измерения входного сигнала до появления сообщения на

средствах отображения в пределах 2 с; период обновления информации на видеокадрах, БЦИ и табло сигнализации в пределах 2 с.

6.1.3 Информационно-вычислительная система АСУ ТП ОРУ Курской АЭС

Система разработана Белорусским теплоэнергетическим НИИ для 3 очереди Курской АЭС [9]. Система реализована на четырех МСКУ. Все МСКУ, а также магистральная подсистема МАПС, связывающая МСКУ с сетью персональных ЭВМ, дублированные.

Основные функции системы:

- сбор и первичная обработка аналоговых и дискретных параметров с частотой опроса 1 с;
- опрос дискретных сигналов с периодом 10 мс, характеризующих состояние устройств релейной защиты, автоматики и коммутационной аппаратуры;
- цифровое осциллографирование аварийных процессов в электрической части АСУ с частотой опроса от 1 до 256 мс. Количество регистрируемых параметров - 160 аналоговых сигналов и 20 дискретных.

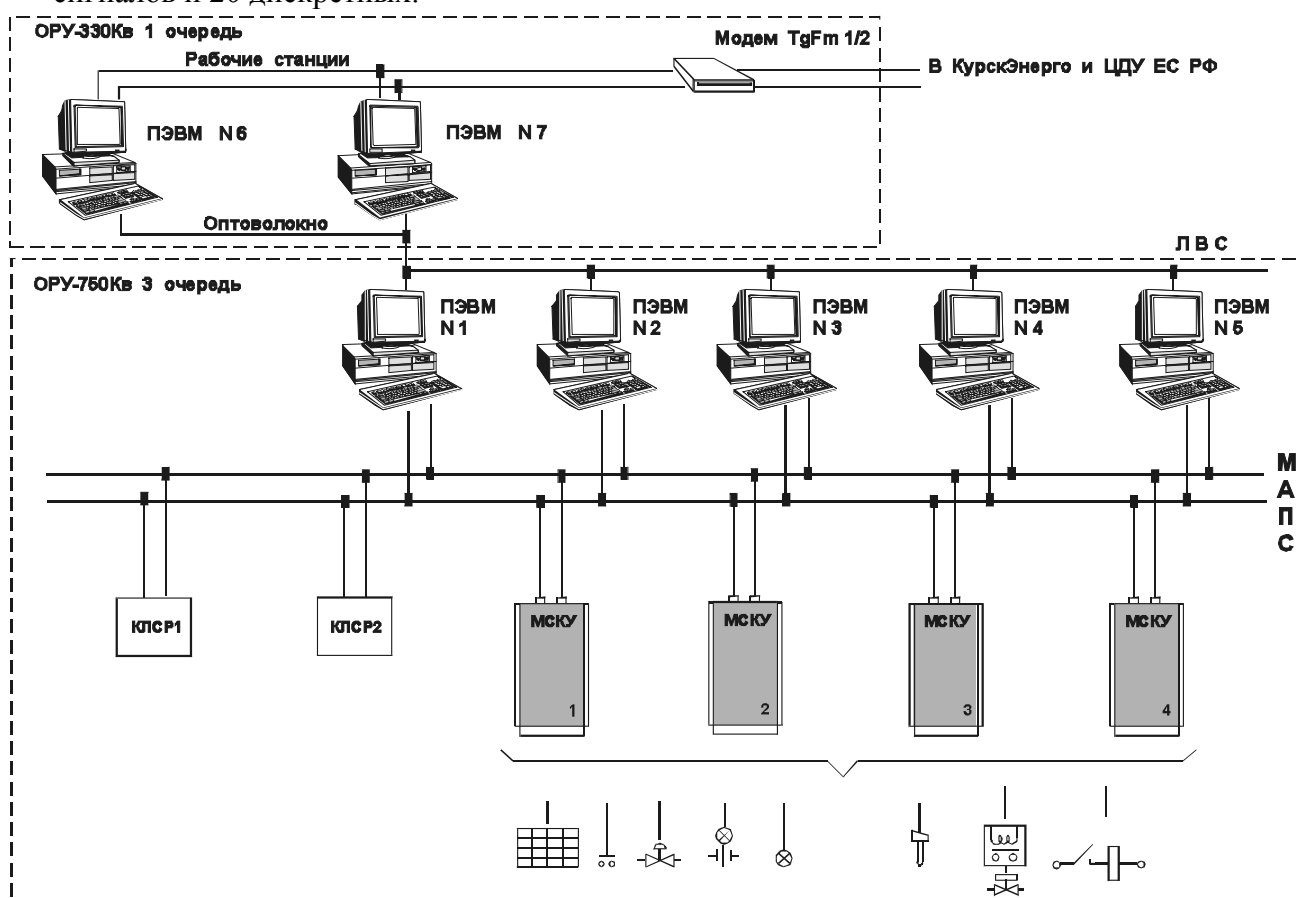


Рис. 6.3. Структурная схема ПТК ИВС ОРУ-750 kW.

Собранная информация от аналоговых и дискретных датчиков передается в локальную вычислительную сеть ПТК ИВС, где ведется многоуровневый архив оперативной информации, а также решается в реальном масштабе времени комплекс технологических задач: контроль и оперативное отображение информации на экранах ПЭВМ; автоматическое ведение суточных ведомостей; автоматическое ведение ведомости переключений коммутационной аппаратуры; контроль положения переключающих устройств, накладок в цепях релейной защиты и противоаварийной автоматики; регистрация аварийных ситуаций; диагностика и расчет остаточного коммутационного ресурса высоковольтных выключателей.

6.1.4 ПТК модернизированной системы внутриреакторного контроля реактора ВВЭР-1000 (ПТК СВРК-М)

ПТК СВРК-М - совместная разработка АО "Импульс", ИЯР РНЦ "Курчатовский институт", СП "ИНИТ". Внедрена на 3 энергоблоке Запорожской АЭС.

СВРК-М предназначена для замены выработавшей ресурс аппаратуры эксплуатирующихся на АЭС СВРК. СВРК-М разработана на базе программно-технических средств МСКУ М. Прикладное программное обеспечение СВРК-М разработано с максимальным использованием имеющего опыта контроля и управления реактора ВВЭР-1000 [75]. ПТК соответствует требованиям современных стандартов Украины, норм и рекомендаций МАЭК и МАГАТЭ для систем важных для безопасности АЭС.

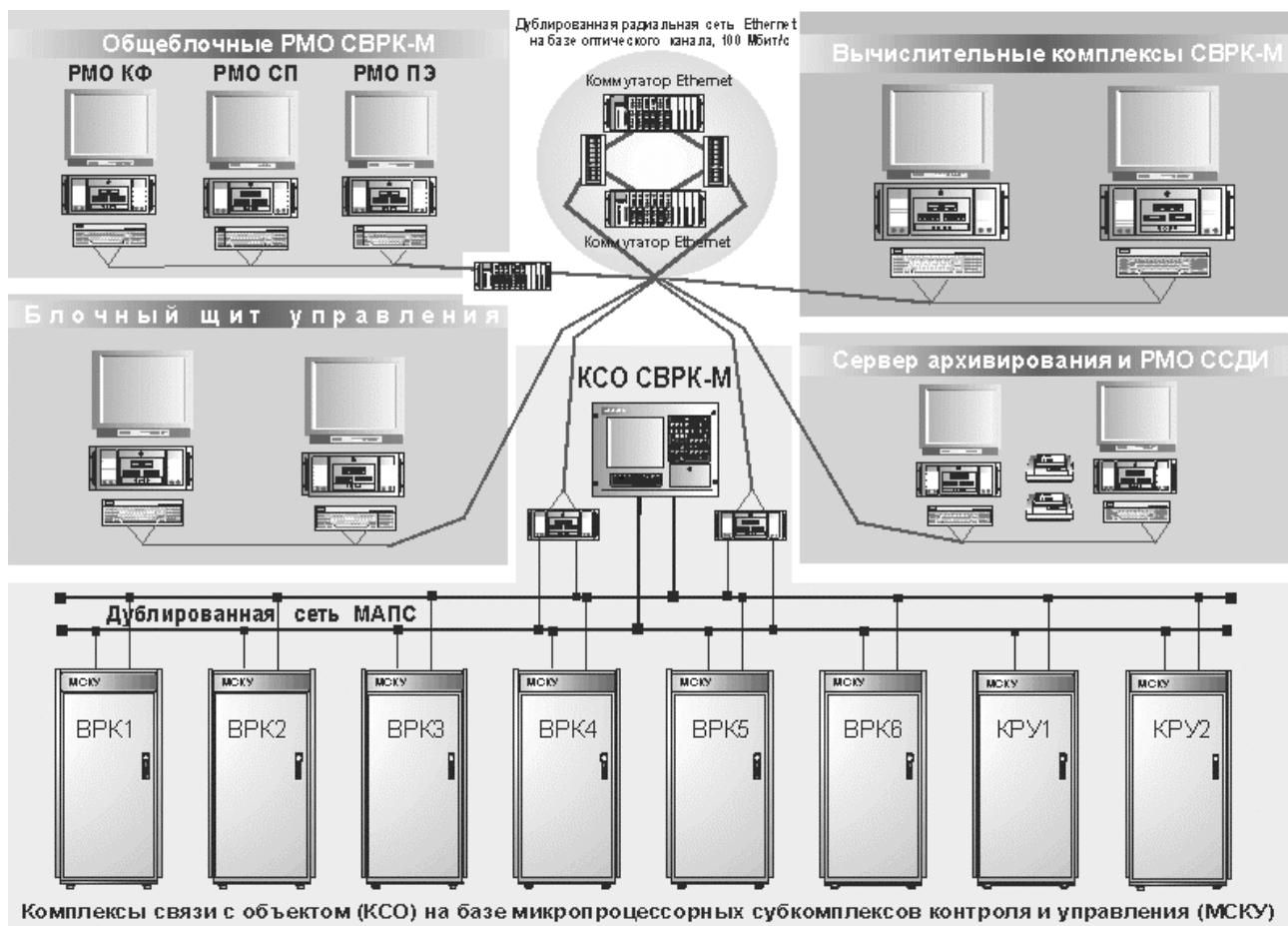


Рис. 6.4. Структурная схема ПТК СВРК-М.

Основные функции СВРК-М:

- контроль технологических процессов в реакторной установке и поддержание параметров этих процессов в пределах, установленных регламентом безопасной эксплуатации АЭС;
- своевременное обнаружение выхода параметров реактора, определяющих пределы безопасной эксплуатации, за допустимые пределы и оповещение персонала с целью предотвращения развития аварий и исключения повреждения основного технологического оборудования реактора;
- обеспечение информационной поддержки для оптимизации протекания технологических процессов реактора в переходных режимах;
- контроль работоспособности и метрологических характеристик измерительных каналов СВРК-М в процессе эксплуатации;
- архивация работы активной зоны реактора и состояния теплоносителя I-го контура.

СВРК-М обеспечивает возможность контроля активной зоны по ее текущему состоянию с соответствующим повышением экономичности и безопасности эксплуатации топлива в активной зоне реактора, в том числе: возможность работы реактора в более маневренном режиме; переход к контролю состояния активной зоны по локальным параметрам, определяющим безопасность активной зоны, включая максимальную линейную нагрузку на ТВЭЛ и запас до кризиса теплообмена; возможность повышения рабочей мощности реактора; возможность внедрения новых топливных циклов. СВРК-М представляет собой многофункциональную интегрированную многоуровневую систему с резервированием наиболее важных функций, открытую для дальнейшего расширения.

Таблица 6.1

Основные технические характеристики ПТК		
СВРК-М		
№ п/п	Наименование	Значение
1	Объем измеряемых сигналов, шт	2800
	в том числе:	
	по энерговыделению	
	<i>сигналы датчиков ДПЗ, шт.</i>	<i>448</i>
	<i>сигналы фоновой составляющей ДПЗ, шт.</i>	<i>448</i>
	<i>сигналы проводимости изоляции ДПЗ, шт.</i>	<i>896</i>
	по термоконтролю	
	<i>сигналы ТП и ТС внутриреакторных датчиков, шт.</i>	<i>240</i>
	<i>сигналы ТП и ТС КНИТ, шт.</i>	<i>248</i>
	нормированные аналоговые сигналы, <i>шт.</i>	<i>192</i>
	дискретные сигналы, <i>шт.</i>	<i>256</i>
2	Количество МСКУ, шт.	8
3	Периодичность измерения информации, с	0,6
4	Погрешность измерений, %	0,04 - 0,1

Нижний уровень ПТК СВРК-М реализован на базе МСКУ, верхний уровень - на базе промышленных рабочих станций ПС 5110.

6.1.5 ПТК информационно-вычислительной системы энергоблока с реактором ВВЭР-1000

ПТК информационно-вычислительной системы (ПТК ИВС) предназначен для представления персоналу оперативной информации о ходе технологического процесса, состоянии оборудования, управляющих системах безопасности и систем нормальной эксплуатации, позволяющей принимать решения по управлению энергоблоком с реактором ВВЭР-1000.

ПТК ИВС обеспечивает:

- функции контроля защит и блокировок;
- архивирование и протоколирование значений параметров и событий;
- возможность анализа работы оборудования и действий персонала в базовых, нестационарных и аварийных режимах и при проведении испытаний.

ПТК ИВС соответствует требованиям существующих стандартов Украины, норм и рекомендаций МЭК и МАГАТЭ для систем, важных для безопасности АЭС.

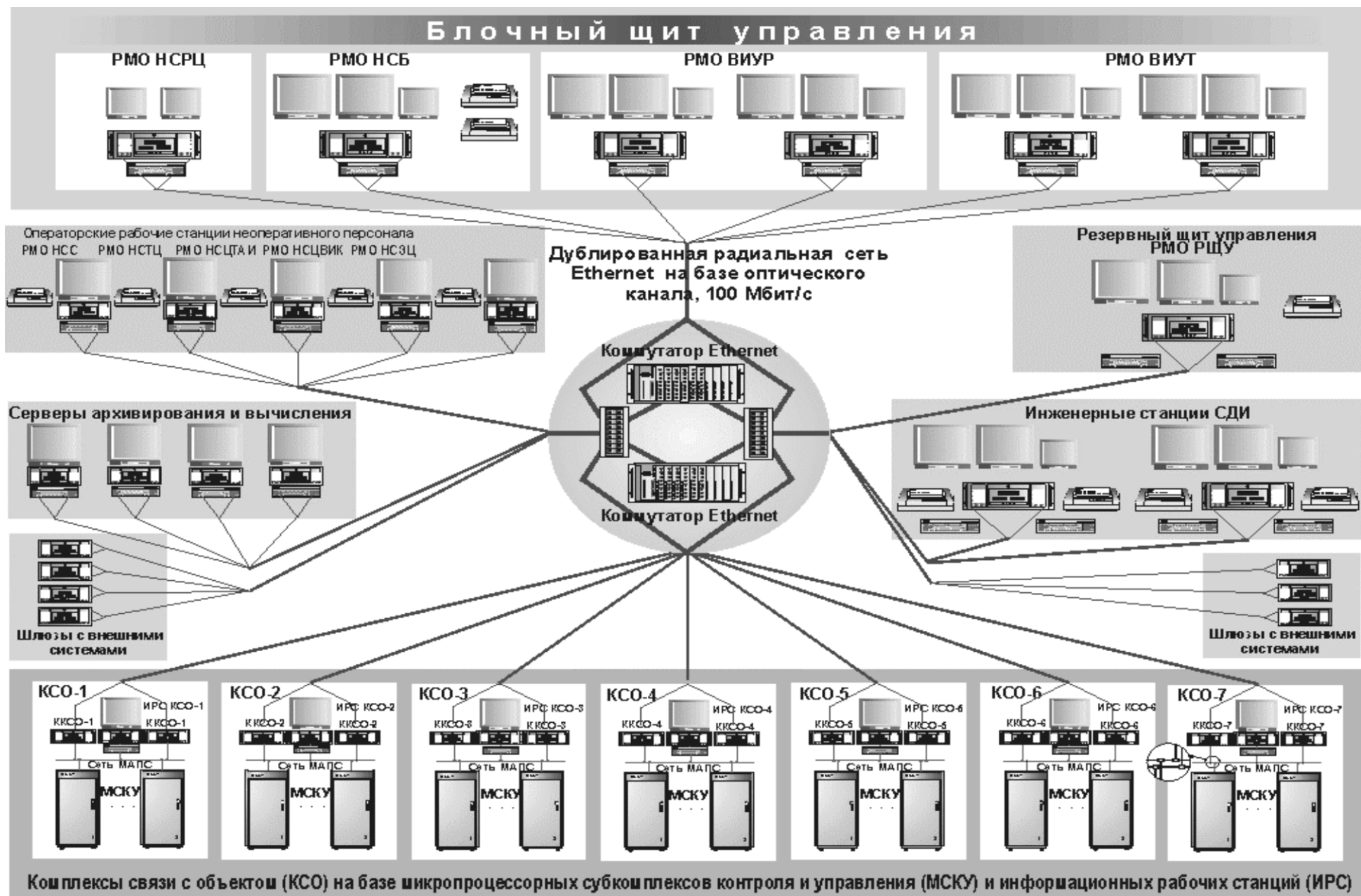


Рис. 6.5. Структурная схема ПТК ИВС энергоблока с реактором ВВЭР-1000.

Основные технические характеристики ПТК ИВС

№№ п/п	Наименование характеристики	Единица измерения	Значение
1	Объем входной и выходной информации по РО		
1.1	Входные аналоговые сигналы	шт.	1800
1.2	Входные дискретные сигналы	шт.	10000
1.3	Выходные дискретные сигналы (ЦОЗ)	шт.	445
2	Объем входной и выходной информации по ТО		
2.1	Входные аналоговые сигналы	шт.	1600
2.2	Входные дискретные сигналы	шт.	3800
2.3	Выходные дискретные сигналы (ЦОЗ)	шт.	280
3	Динамические характеристики		
3.1	Период обновления аналоговой и дискретной информации в базе данных	s	1
3.2	Период обновления информации на дисплеях операторских станций	s	1
3.3	Разрешающая способность фиксации в архиве по дискретным быстроизменяющимся сигналам по остальным дискретным сигналам по отклонениям аналоговых сигналов	ms ms s	10 100 2
3.4	Реакция на изменение входного сигнала от входа МСКУ до экрана дисплея: аналогового сигнала дискретного сигнала	s s	2-3 1-1,5
4	Предел допустимой приведенной погрешности канала с отображением на дисплеях		
4.1	Каналы с унифицированным токовым сигналом	%	±0,15
4.2	Каналы термоэлектрических преобразователей сопротивления	%	±0,25
4.3	Каналы термоэлектрических преобразователей	%	±0,25
5	Количество видеокладов		
5.1	Вызываемых через экранное меню	шт.	Не огран.
5.2	Вызываемых через функциональную клавиатуру по РО по ТО	шт. шт.	120 120
6	Характеристики функции архивирования		
6.1	Период фиксации аналоговых сигналов в нормальном режиме	s	2
6.2	Период фиксации в аварийных режимах	s	1
6.3	Количество архивируемых аналоговых сигналов		Все
6.4	Емкость оперативного архива	сутки	15
6.5	Емкость долговременного архива		Время эксплуат. блока

В табл. 6.2 обозначено: РО – реакторное отделение; ТО – турбинное отделение; ЦОЗ – централизованное опробование защит.

Технические средства ПТК ИВС соответствуют требованиям ГОСТ 29075, и комплексу стандартов ГОСТ 25804.1 ÷ ГОСТ 25804.8. ПТК ИВС на базе МСКУ 2М обеспечивает:

- получение лучших надежностных, метрологических и временных характеристик системы по сравнению с существующим ПТК;
- совместимость по параметрам входных и выходных сигналов МСКУ 2 с существующей аппаратурой АСУ ТП энергоблока;
- высокую помехозащищенность системы за счет реализации полного гальванического разделения каналов ввода-вывода информации для всех видов аналоговых и дискретных сигналов;
- размещение оборудования МСКУ 2 на площадях помещений установки КСО М-64 и УЛУ2-ЭВМ.

ПТК ИВС построен как децентрализованная резервированная система, состоит из ПТК реакторного отделения (РО), турбинного отделений (ТО) и общеблочных компонентов. Система температурного контроля генератора реализована на двух резервированных МСКУ 2 в составе ПТК турбинного отделения. Он имеет двухуровневую структуру. Нижний уровень образуют комплексы связи с объектом, реализованные на МСКУ 2. Верхний уровень ПТК построен на базе промышленных рабочих станций ПС 5110. Оперативный обмен данными между нижним и верхним уровнями ПТК осуществляется через промышленную дублированную ЛВС.

6.2 ПТК на объектах газового комплекса

6.2.1 ПТК автоматизированных систем управления компрессорными станциями

Одним из примеров реализации АСУ ТП является автоматизированная система управления (АСУ) газлифтной компрессорной станцией (ГЛКС) Анастасьевского месторождения (АО «Укрнефть»), разработанная на базе МСКУ М [76]. Структурная схема АСУ ТП компрессорной станции представлена на рисунке 6.6.

ГЛКС предназначена для компримирования нефтяного газа первой ступени сепарации при газлифтной добыче нефти. Она состоит из компрессорных станций (КС) низкого и высокого давлений. Каждая из КС включает три компрессорные установки (КУ) на базе турбокомпрессорных агрегатов (ТКА) с центробежными компрессорами (ЦК) и авиационными приводами ДЗ36-2-1 мощностью 6,3 MW (АО «Мотор Сич», г. Запорожье), а также систему осушки газа.

ГЛКС производит сбор конденсата, осушку газа, регенерацию гликоля, подачу осушенного газа на газлифтные скважины. АСУ управляет этими процессами, координирует работу КС низкого и высокого давлений.

Нижний уровень образуют системы автоматизированного управления двигателем (САУД) типа ДЗ36 и МСКУ. Связь САУД с МСКУ осуществляется по интерфейсу ИРПС, а также по физическим линиям через каналы ввода-вывода типа «сухой контакт».

Средний уровень образуют рабочие места операторов-технологов (РМОТ) КС низкого и высокого давлений, а верхний – РМОТ ГЛКС, созданные на базе промышленных персональных ЭВМ с процессором Intel Pentium. МСКУ и РМОТ объединены посредством промышленной ЛВС МАПС.

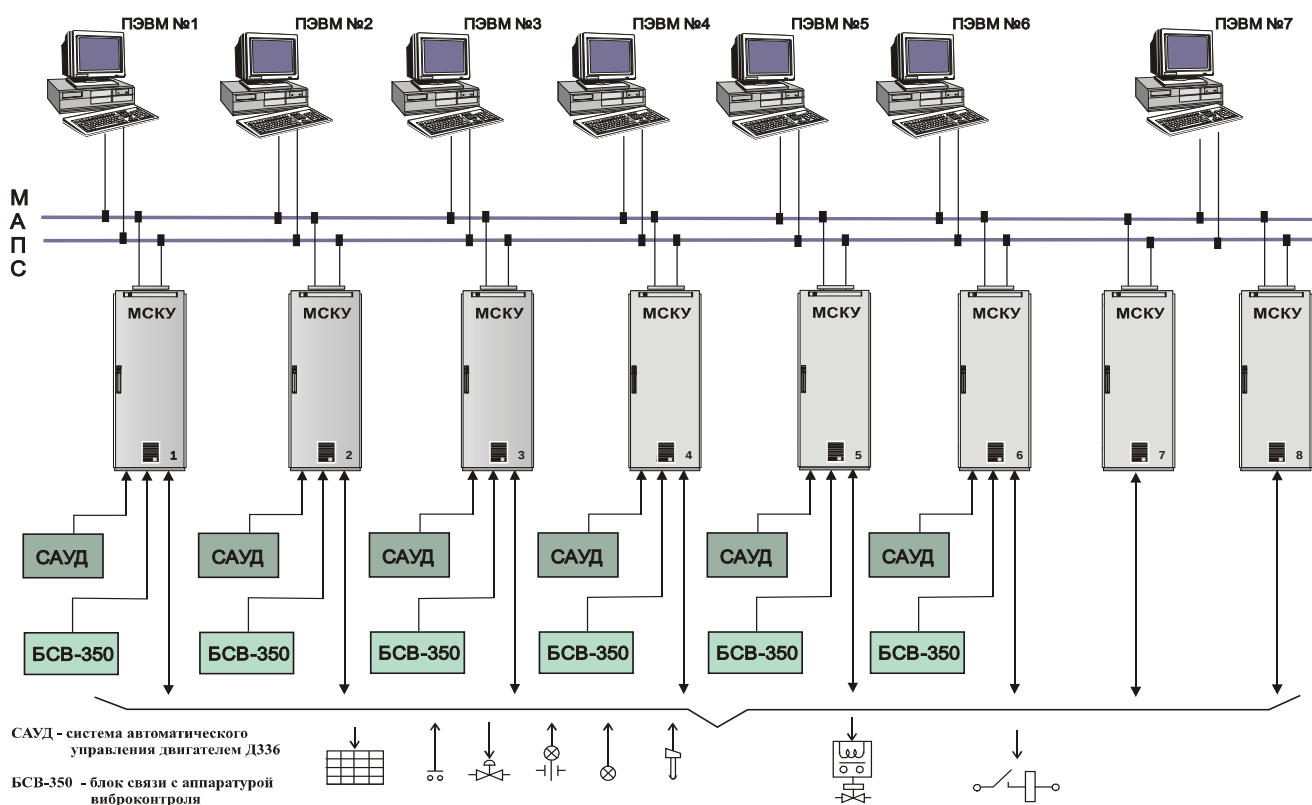


Рис. 6.6 Структурная схема ПТК АСУ ТП компрессорной станции Анастасьевского месторождения

Основные функции системы

Технические и программные средства АСУ ГЛКС обеспечивают:

- циклический контроль технологических параметров, измерение и фиксацию событий по предупредительным и аварийным остановам;
- вычисление косвенных параметров;
- автоматическое обнаружение, отображение и звуковую сигнализацию отклонений технологических параметров от заданных пределов;
- запоминание сигналов, вызвавших аварийный останов, а также значений основных технологических параметров агрегата, положений исполнительных механизмов и кранов при срабатывании защиты с возможностью ретроспективного анализа состояния ГЛКС за последние 100 с до начала аварии;
- непрерывный контроль выполнения команд управления и регулирования, контроль исправности цепей дискретных датчиков аварийной сигнализации, измерительных цепей датчиков давления, перепада давления, температуры и уровня, цепей управления исполнительными механизмами, обеспечивающими аварийный останов ТКА, КУ и ГЛКС;
- контроль метрологических характеристик измерительных каналов с помощью стандартных средств измерения;
- периодический контроль работоспособности оборудования с выдачей сообщений на пульт оператора (периодичность контроля и выдачи сообщений устанавливается оператором, длительность цикла не менее 1 с);
- автоматическое управление режимами работы ГЛКС и КУ;
- дистанционное ручное управление вспомогательными механизмами при работающем оборудовании ГЛКС; автоматическое обнаружение пожара и управление исполнительными элементами системы пожаротушения;

- автоматический перезапуск с интервалом 3 с вспомогательных механизмов после кратковременного (от 1 до 3 с) пропадания электропитания системы (технические средства системы должны обеспечивать автоматический останов агрегата при исчезновении напряжения переменного тока);

- блокировку выполнения команд оператора, если они не предусмотрены алгоритмом управления или регулирования.

Характеристики АСУ ГЛКС

Число входных каналов:	
- аналоговых (-10...+10 V)	480
- аналоговых (-20...+20 mA)	424
- частотных (0...20 000 Hz)	24
- от термопреобразователей сопротивлений	324
- от термопар	80
- дискретных 48 V и типа «сухой контакт»	1792
Число выходных каналов:	
- аналоговых постоянного тока (4-20 mA)	28
- дискретных (48 V; 0,2 A)	720
- дискретных типа «сухой контакт»	96
- дискретных (220 V; 1A)	732
Число команд с клавиатур КФ-3М	1130
Число основных алгоритмов управления	536
Число кадров мнемосхем представления информации оператору-технологу	85

Основные функции МСКУ:

- ввод информации по входным каналам связи с объектом с заданными индивидуальными для каждой группы каналов периодами опроса;
- контроль достоверности принятых данных;
- фильтрация принятых данных аналоговых величин, их линеаризация и масштабирование;
- оперативное изменение аварийных, предупредительных и управляющих установок, а также значений временных задержек при работающем агрегате;
- вывод сигналов на исполнительные механизмы;
- передача информации в РМОТ;
- контроль (в реальном масштабе времени) работоспособности аппаратуры;
- прием управляющих воздействий и других данных от оператора-технолога и включение по ним соответствующих модулей прикладного программного обеспечения;
- формирование времени суток и даты;
- обеспечение регламента выполнения прикладных задач.

Основные функции РМОТ:

- представление текущей информации об управляемой установке (в виде мнемосхем, графиков, таблиц, диаграмм и т. п.) с период обновления данных не более 2 с;
- оперативное отображение (с одновременным включением звуковых сигналов) текстовых сообщений о различных нарушениях в работе установки и других событиях, требующих вмешательства оператора-технолога;
- оповещение обслуживающего персонала о сбоях и отказах в работе аппаратуры системы;
- формирование архива измеряемых значений и состояний технологических параметров, исполнительных механизмов установки (при этом должны формироваться следующие

подархивы: трехсекундный в течение 5 min, пятиминутный в течение 2 h и двухчасовой в течение 90 суток);

- обеспечение оперативного доступа оператора-технолога (по его запросу) к архиву с возможностью получения данных из архива за указанный промежуток времени в форме таблиц, графиков, диаграмм на экране, а также в форме таблиц при выводе на печать;

- формирование архива нарушений с обеспечением доступа к нему, аналогично описанному выше;

- формирование архива аварийных остановок ГЛКС и КУ с запоминанием значений параметров и сигналов, вызвавших остановку и сопутствующих ей на временном интервале в течение 5 min после остановки (с формированием трехсекундных подархивов);

- индикация команд, вводимых оператором, контроль их допустимости;

- контроль выполнения команд;

- индикация наличия команд управления исполнительными механизмами.

6.2.2 ПТК системы автоматического управления ГПА-Ц6,3В

Система автоматического управления газоперекачивающим агрегатом (САУ ГПА) ГПА-Ц-6,3В с двигателем НК-12СТ, предназначена для автоматизации управления ГПА по выполнению функций автоматического управления агрегатным оборудованием, регулирования, контроля и защит для обеспечения безаварийной работы ГПА в режимах продолжительной работы и останова.

Функции управления:

- проверка пусковой готовности;

- проверка исправности каналов защиты ГПА;

- «холодная» прокрутка двигателя; автоматический пуск ГПА с автоматическим выводом его на заданный режим и поддержание режима работы в соответствии с заданием, выдаваемым оператором-технологом;

- нормальный останов;

- аварийный останов со стравливанием или без стравливания газа;

- автоматическое управление исполнительными механизмами и кранами газовой обвязки агрегата в соответствии с алгоритмами управления;

- отработка режимов работы, задаваемых оператором;

- автоматическая защита по технологическим параметрам;

- дистанционное управление исполнительными механизмами с панели управления и от рабочей станции;

- управление исполнительными механизмами системы пожаротушения по заданному алгоритму;

- автоматический перезапуск, с интервалом 3 s, вспомогательных механизмов по заданному алгоритму;

- экстренный останов ГПА по заданному алгоритму, по команде оператора.

Функции контроля:

- автоматический непрерывный контроль исправности цепей управления особо ответственными исполнительными механизмами и вспомогательным оборудованием ГПА;

- автоматический непрерывный контроль цепей аналоговых датчиков и цепей дискретных датчиков, участвующих в аварийных защитах;

- контроль состояния оборудования и отклонений технологических параметров при достижении параметрами предельных значений (уставок);

- автоматический контроль исправности САУ ГПА на уровне блоков;

- защита программного обеспечения САУ ГПА от несанкционированного доступа.

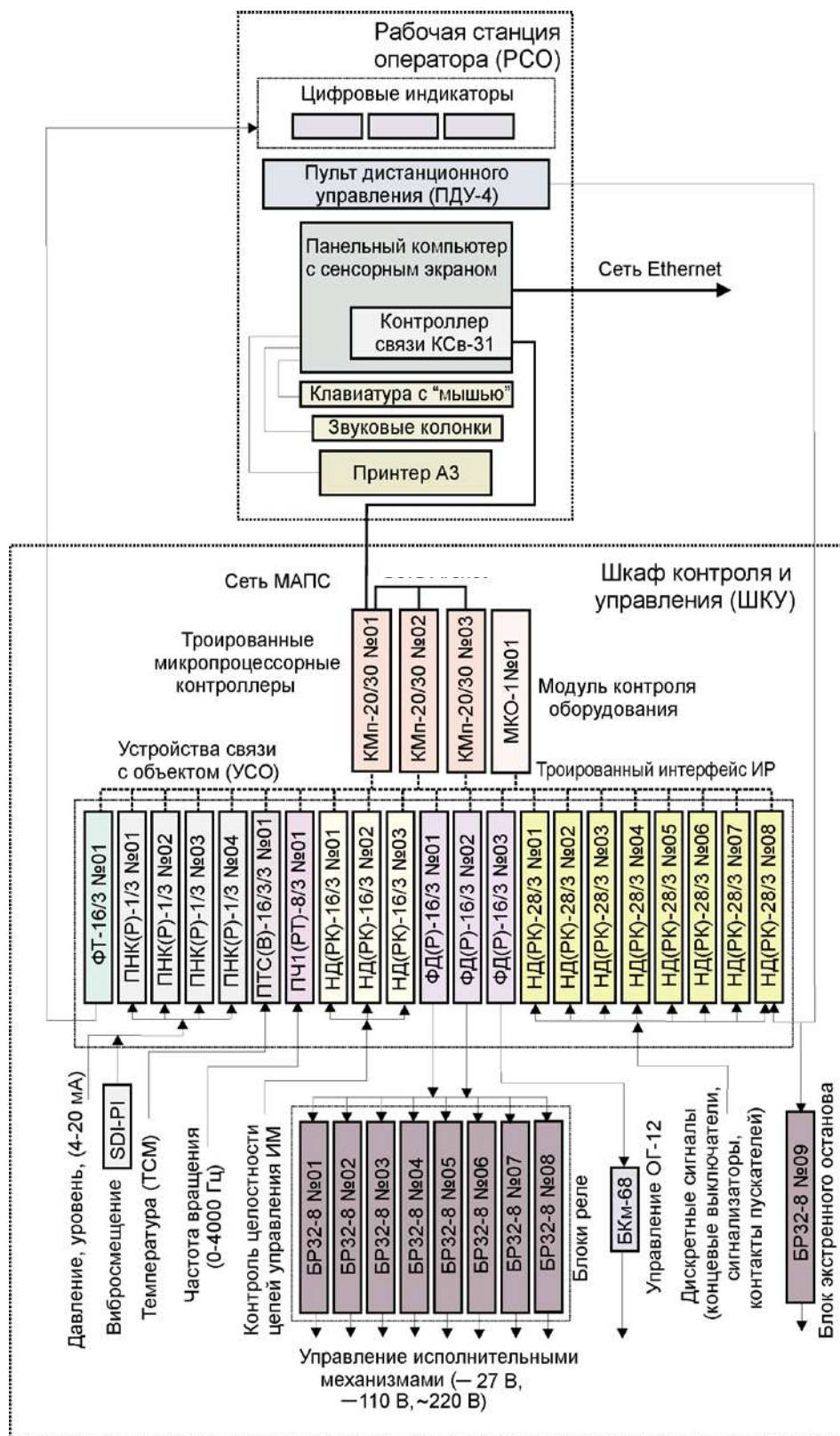
Таблица 6.3

Основные характеристики САУ ГПА:

Параметры системы	Параметры сигналов	Количество каналов, шт.
1. Аналоговые входные:		
- температура ТС, ТП	0 - 200 Ω , 0 – 80 mV	не менее 16
- давление, перепад давления, уровень, вибрация	4 - 20 mA	не менее 32
- частота вращения двигателя	0 - 32000 Hz	не менее 8
2. Дискретные входные:		
- типа "сухой контакт" с контролем линии связи	0 – 130 Ω - замкнутый, 1.4 – 1.8 $k\Omega$ – разомкнутый, > 20 $k\Omega$ - обрыв	не менее 112
- контроль цепей исполнительных механизмов	~220 V, -220 V, -27 V	не менее 64
3. Дискретные выходные:		
- управление исполнительными механизмами постоянного и переменного тока (напряжение/ток);	27 V/5 A, 220 V /3 A	не менее 64
4 Аналоговые выходные:		
- управление исполнительными механизмами	4-20 mA	не менее 16

Информационные функции:

- непрерывный контроль технологических параметров, в том числе измерение и представление по вызову оператора на экране рабочей станции оператора (РСО) аналоговых параметров САУ ГПА с одновременным указанием уставок (предупредительных и аварийных);
- вызов группы контролируемых параметров по желанию оператора с отображением в виде трендов;
- отображение вычисляемых параметров;
- представление на экране РСО мнемосхем агрегата с указанием значений контролируемых параметров и положений исполнительных механизмов;
- постоянное представление на цифровых табло температуры газа, частоты вращения и перепада давлений «масло-газ»;
- отображение, звуковая и мигающая световая сигнализации при достижении технологическими параметрами предупредительных и аварийных уставок;
- представление информации о невыполненных предпусковых условиях;
- представление информации об основных режимах работы агрегата: «готов к пуску», «автоматический пуск», «кольцо», «магистраль», «нормальный останов», «требование нормального останова», «аварийный останов», «экстренный останов»;
- запоминание сигналов, вызвавших аварийный останов, а также значений основных параметров агрегата, положения исполнительных механизмов и кранов при срабатывании защиты с возможностями ретроспективного анализа состояния агрегата (с дискретностью 0.1 s) за 10 *min* до начала аварии и 5 *min* после аварии;
- формирование массивов текущей и ретроспективной информации в виде непрерывно обновляемых массивов данных технологических параметров, режимов работы, отклонения от заданных уставок и действий оператора.



6.7. Структура САУ ГПА.

На рис.6.7 приведена структурная схема ПТК САУ ГПА. САУ ГПА обеспечивает следующее быстродействие для измерительных, вычислительных, управляющих и информационных каналов САУ ГПА:

- время от изменения текущего значения параметра до выдачи выходного сигнала на исполнительный механизм;
- для функций логического управления и защиты ГПА, не более – 0,3 s;

- время от изменения текущего значения параметра до его выдачи на средство отображения или до его записи в архив, учитывая скорость обмена информацией между ШКУ и РСО ГПА, не более:

- для функций архивации - 0,1 с;
- для функций представления визуальной информации – 1,0 с.

6.3 ПТК на объектах нефтехимии

6.3.1 АСУ ТП узла К-105, К-106/І, II установки концентрирования пропилена на АО "УФАОРГСИНТЕЗ"

Комплекс технических средств (КТС) АСУ ТП узла К-105, К-106/І, II организован в виде распределенной микропроцессорной системы, на базе локальной вычислительной сети МАПС и состоит из центральной станции, инженерной станции, рабочих станций и МСКУ. Структурная схема ПТК АСУ ТП представлена на рис. 6.8. Центральная станция осуществляет регистрацию истории изменения параметров во времени и сообщений о нарушениях; расчет технико-экономических показателей и формирование и печать отчетных документов. Инженерная станция решает задачи анализа работы АСУ ТП, ее настройки, тестирования и переконфигурирования, загрузки и перезагрузки абонентов сети.

Рабочие станции выполняют визуализацию состояния объектов управления в реальном масштабе времени, сигнализацию отклонений технологического процесса от регламентных норм; управление регуляторами, исполнительными механизмами и электроприводами в дистанционном режиме и визуализацию архивных данных, регистрацию сообщений о нарушениях и технологических сообщений.

Шлюз выполняет функции связи с системой автоматического управления компрессорами. МСКУ реализуют следующие функции:

- автоматический сбор сигналов с аналоговых и дискретных датчиков, вторичных преобразователей;
- контроль достоверности принятой информации;
- определение значений технологических параметров по измеренным сигналам;
- фильтрацию, сглаживание измеренных величин;
- обнаружение отклонений технологических параметров от регламентных норм;
- автоматическое регулирование технологических параметров;
- защиту технологического процесса и оборудования при возникновении аварийной ситуации (противоаварийную защиту).

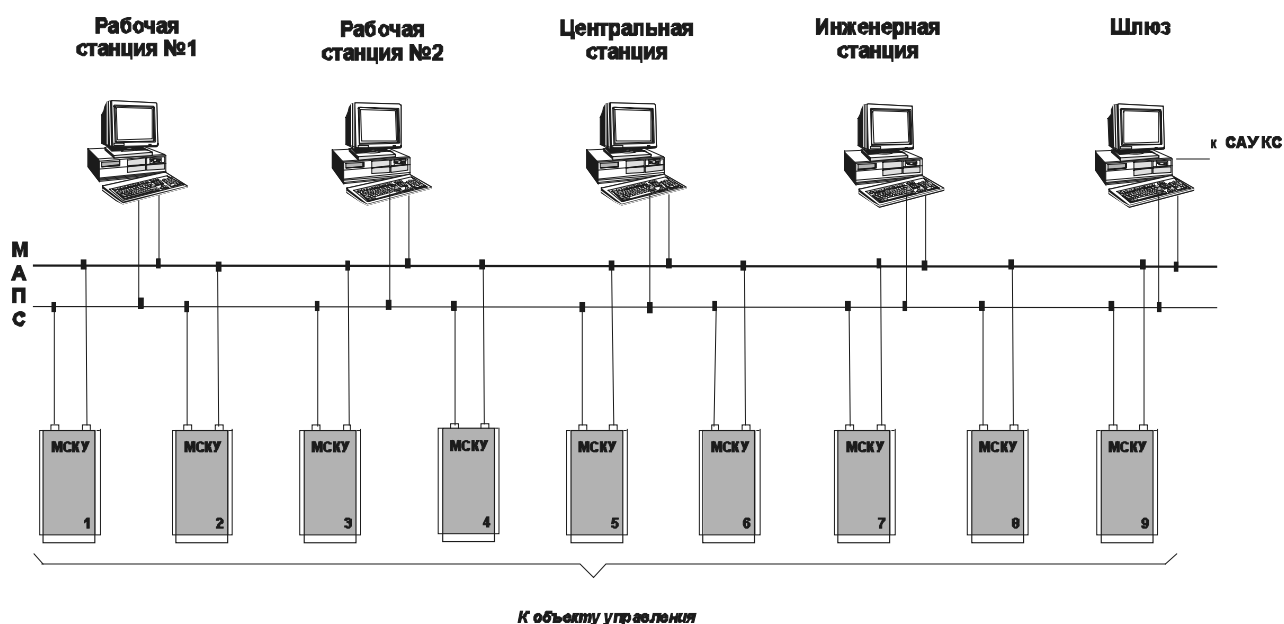


Рис. 6.8. Структурная схема АСУ ТП установки концентрирования пропилена на АО «Уфаоргсинтез».

6.3.2 АСУ ТП нагревательных печей нефтеперегонной установки на АО «Укртатнефть»

Система предназначена для автоматизированного контроля и управления технологическим процессом подогрева отбензиненной нефти в многосекционной нагревательной печи нефтеперегонной установки АО «Укртатнефть». Структурная схема АСУ-Печь приведена на рис. 6.9.

Основу технических средств АСУ ТП составляют микропроцессорные средства контроля и управления МСКУ 2. Автоматическое регулирование технологических параметров нагревательной печи производится при помощи многоканальных регуляторов Protronic 550. В качестве оборудования рабочего места оператора-технолога применены два персональных компьютера промышленного исполнения.

Основные функции системы

Система выполняет следующие функции контроля, автоматического регулирования и противоаварийной защиты:

- непрерывный контроль технологических параметров процесса;
- представление на рабочем месте оператора-технолога мнемосхем технологического процесса с указанием значений контролируемых параметров и положения исполнительных механизмов;
- отображение, звуковая и мигающая световая сигнализация при достижении технологическими параметрами предупредительных и аварийных уставок;
- запоминание сигналов, вызвавших срабатывание защит, а также значений технологических параметров и положения исполнительных механизмов с возможностью ретроспективного анализа состояния технологического процесса с дискретностью 0,5 с;

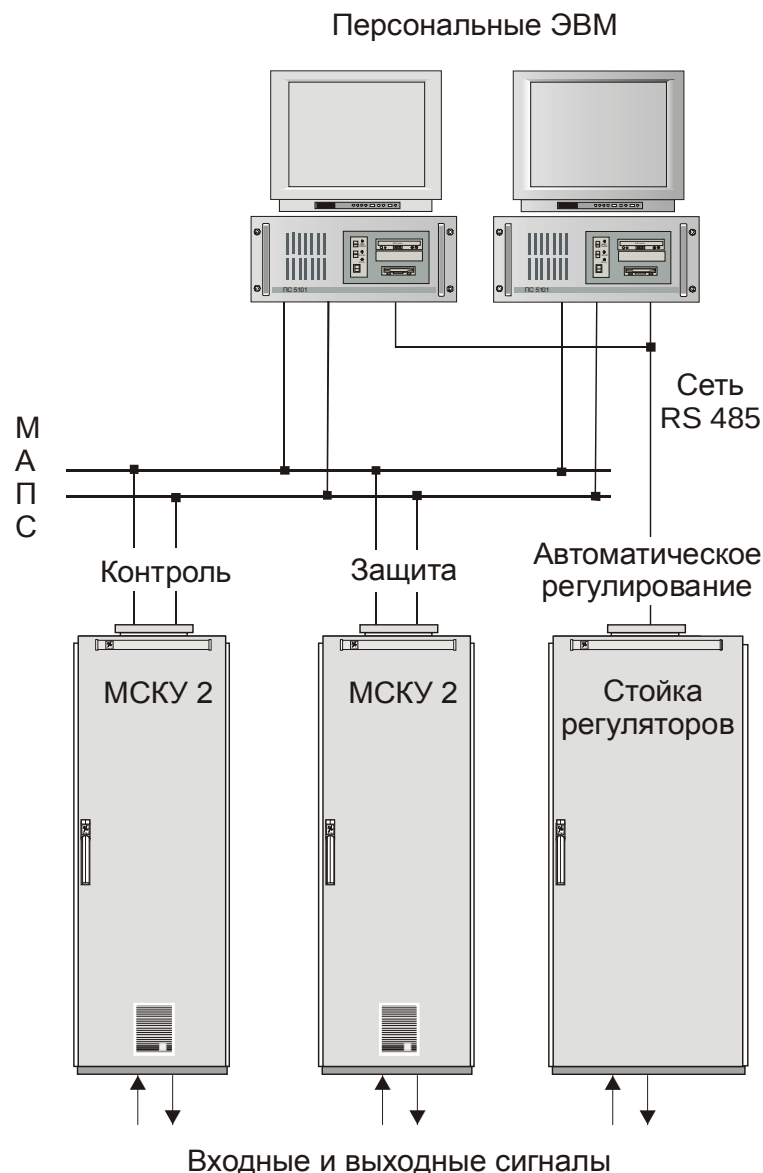


Рис. 6.9 Структурная схема ПТК АСУ ТП нагревательной печи

- формирование массивов текущей и ретроспективной информации в виде непрерывно обновляемых массивов данных технологических параметров, отклонения от заданных уставок и действий оператора;
- автоматическое регулирование основных технологических параметров;
- противоаварийная защита нагревательной печи;
- ведение и печать сменного рапорта;
- постоянная диагностика технических средств системы;
- обмен информацией с системой высшего уровня.

Показатели назначения системы:

ввод аналоговых параметров 4- 20 <i>mA</i> , шт.	150
ввод дискретных сигналов типа «сухой контакт », шт.	18
вывод аналоговых сигналов 4 – 20 <i>mA</i> , шт.	60
вывод дискретных сигналов типа «сухой контакт », шт.	16

Общие показатели:

температура окружающей среды	от + 5 до + 40 °С
относительная влажность при + 35 °С	80 %
атмосферное давление	от 84 до 107 kPa
степень защиты	IP20
стойкость к механическим воздействиям	группа L3
устойчивость к электростатическим разрядам	степень жесткости 2
электрическое сопротивление изоляции	не менее 20 MΩ
изоляция электрически несвязанных цепей	1500 V

6.4 ПТК на объектах пищевой промышленности

6.4.1 АСУ ТП основного производства сахарного завода

Основные особенности рассмотрены на примере внедрения АСУ ТП Кагарлыкского сахарного завода [12]. Эта система разработана в рамках модернизации устаревшей системы управления следующими технологическими участками: выпарная установка; фильтрация; дефекосатурация. Созданная АСУ ТП является двухуровневой распределительной системой (рис. 6.10).

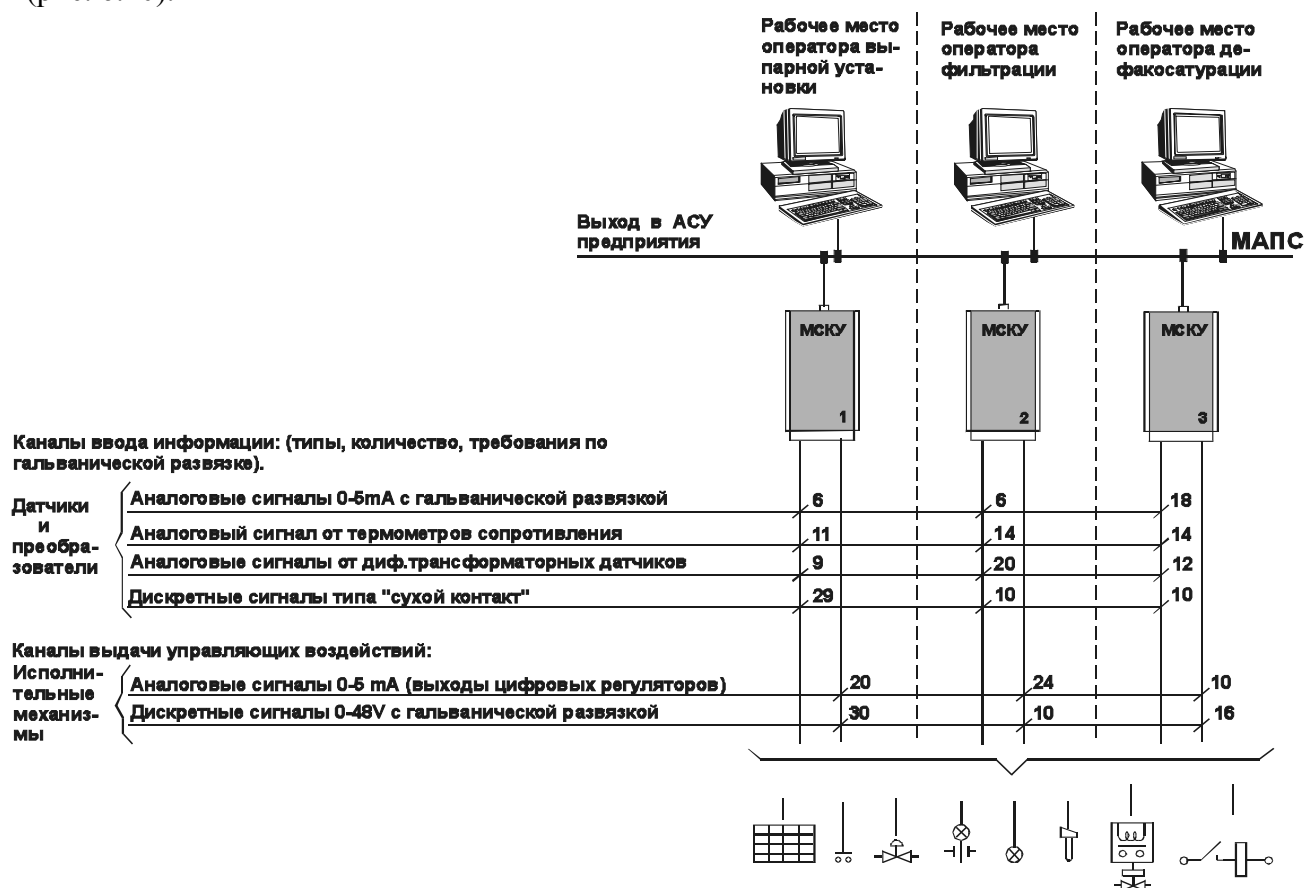


Рис. 6.10. Структурная схема АСУ ТП Кагарлыкского сахарного завода.

ПТК АСУ ТП реализован на технических и программных средствах системы МСКУ М. Нижний уровень системы выполнен на базе микропроцессорных субкомплексов контроля и управления (МСКУ), верхний – на ПЭВМ, совместимых с IBM PC/AT. МСКУ выполняют функции сбора, обработки информации, поступившей от установок, агрегатов и обеспечивают непосредственное цифровое управление исполнительными механизмами по заданиям,

полученным от верхнего уровня. На ПЭВМ реализованы рабочие места операторов технологических участков. Общий объем контролируемых параметров в системе – 300, число цифровых регуляторов – 50, число каналов управления дискретными исполнительными механизмами – 60.

К основным функциям АСУ ТП сахарного завода относятся: информационные, управляющие и технологические. Последние разделяются на регулирующие и контролирующие.

Информационные функции АСУ ТП: отображение хода технологического процесса на экране монитора в виде мнемосхем; регистрация и сбор информации; отображение и регистрация аварийных ситуаций и расчет и представление показателей ТЭП.

Управляющие функции: формирование команд управления дискретными исполнительными механизмами с клавиатуры ПЭВМ и расчет и выдача заданий регуляторам технологических параметров.

Технологические функции: регулирование уровня в сборнике перед выпарной установкой, температуры на подогревателе, уровней по всем корпусам, уровней по всем сборникам конденсата, давления греющей камеры, разрежения в надсоковом пространстве конденсатора, температуры сиропа на подогревателе перед фильтрацией, уровня в сборнике сиропа перед вакуум-аппаратами, температуры барометрической воды на конденсаторе, подачи аммиачной воды в сборнике перед выпарной установкой, *pH* преддефекации, *pH* 1 сатурации и *pH* 2 сатурации. Контролю подлежат следующие параметры: давление в надсоковом пространстве в каждом корпусе, расход сока на выпарную установку, температура в греющей камере всех корпусов, температура в надсоковом пространстве корпусов, состояние насосов подачи сока и откачки сиропа, состояние подачи сиропа в сборник на вакуум-аппараты и состояние насосов на конденсатах.

Сигнализации подлежат следующие технологические параметры (по выходу за пределы номинальных значений): уровень в сборнике сока перед выпарной установкой, уровень корпусов, давление в греющей камере, разрежение в конденсаторе, уровень в сборнике сиропа перед вакуум-аппаратами и температура барометрической воды на конденсаторе.

6.4.2 АСУ ТП ТЭЦ сахарного завода

Возможности системы автоматизации ТЭЦ сахарных заводов рассмотрены на примере АСУ ТП ТЭЦ Лучанского сахарного завода [12].

Разработка АСУ ТП направлена на оптимизацию выполнения основной задачи ТЭЦ – обеспечение бесперебойного снабжения сахарного завода электрической и тепловой энергией в период сахароварения с минимальным расходом энергоресурсов, минимизацию производственных и эксплуатационных затрат, а также выполнение требований по охране окружающей среды и обеспечение безопасной эксплуатации оборудования. Основными объектами автоматизации на ТЭЦ являются котлоагрегаты БМ-35М и Темпелла-Карлсона с рабочим давлением пара 3,9 МПа, температурой перегретого пара 440 °С и расходом соответственно 35000 и 25000 kg/h. АСУ ТП удовлетворяет всем предъявленным к ней требованиям. Эта система выполняет автоматизированное управление котлоагрегатами ТЭЦ и вспомогательным оборудованием во всех режимах их работы с решением задач минимизации расхода энергоресурсов и обеспечением необходимого уровня технологических защит и защит оборудования.

С целью достижения необходимых характеристик по производительности, надежности и живучести АСУ ТП ТЭЦ реализована на базе технических и программных средств МСКУ М. ПТК АСУ ТП является блочно-модульной системой с сетевой резервированной структурой, обеспечивающей автоматическую реконфигурацию и работоспособность при отказах отдельных ее компонентов. Структурная схема системы приведена на рис. 6.11.

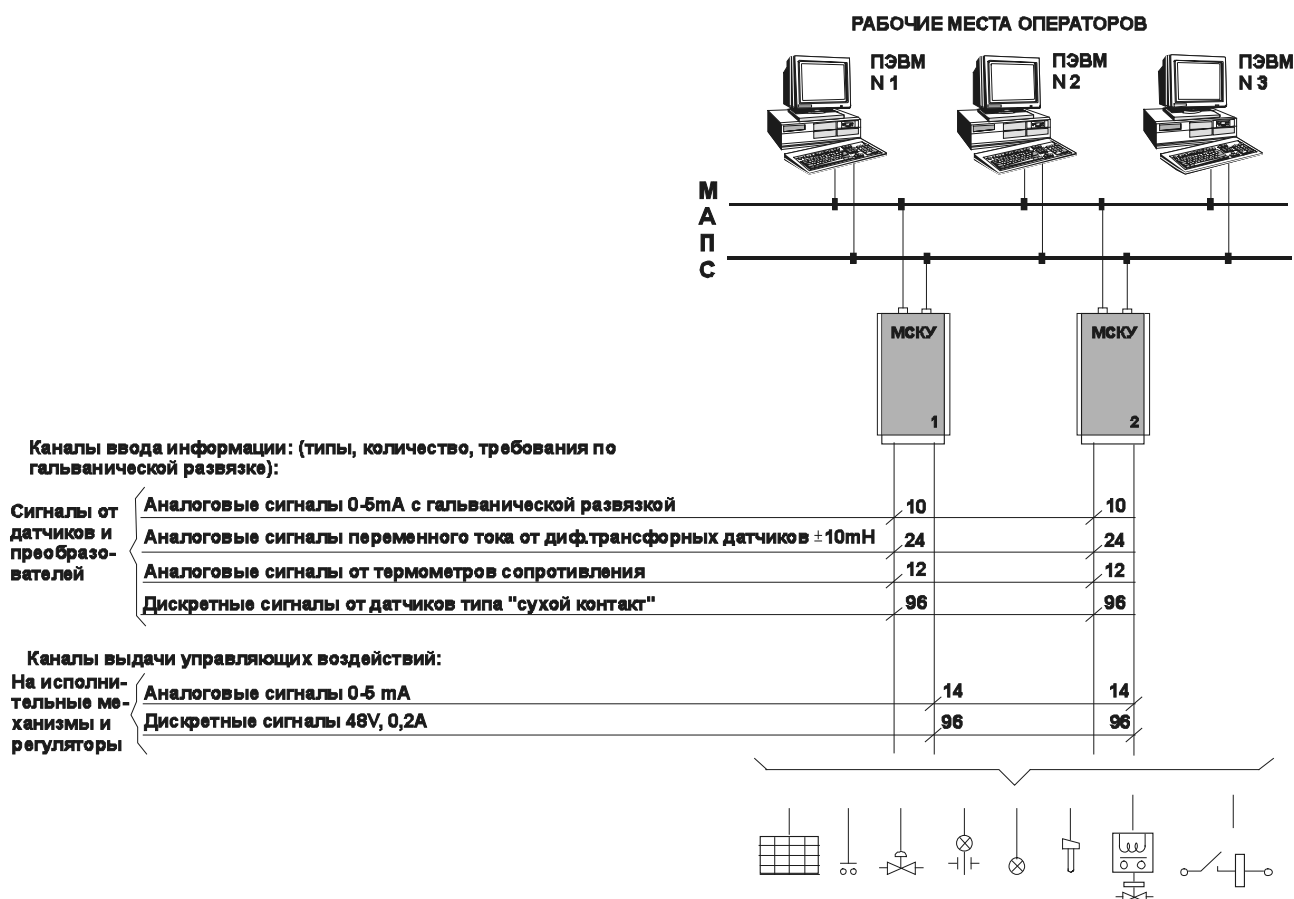


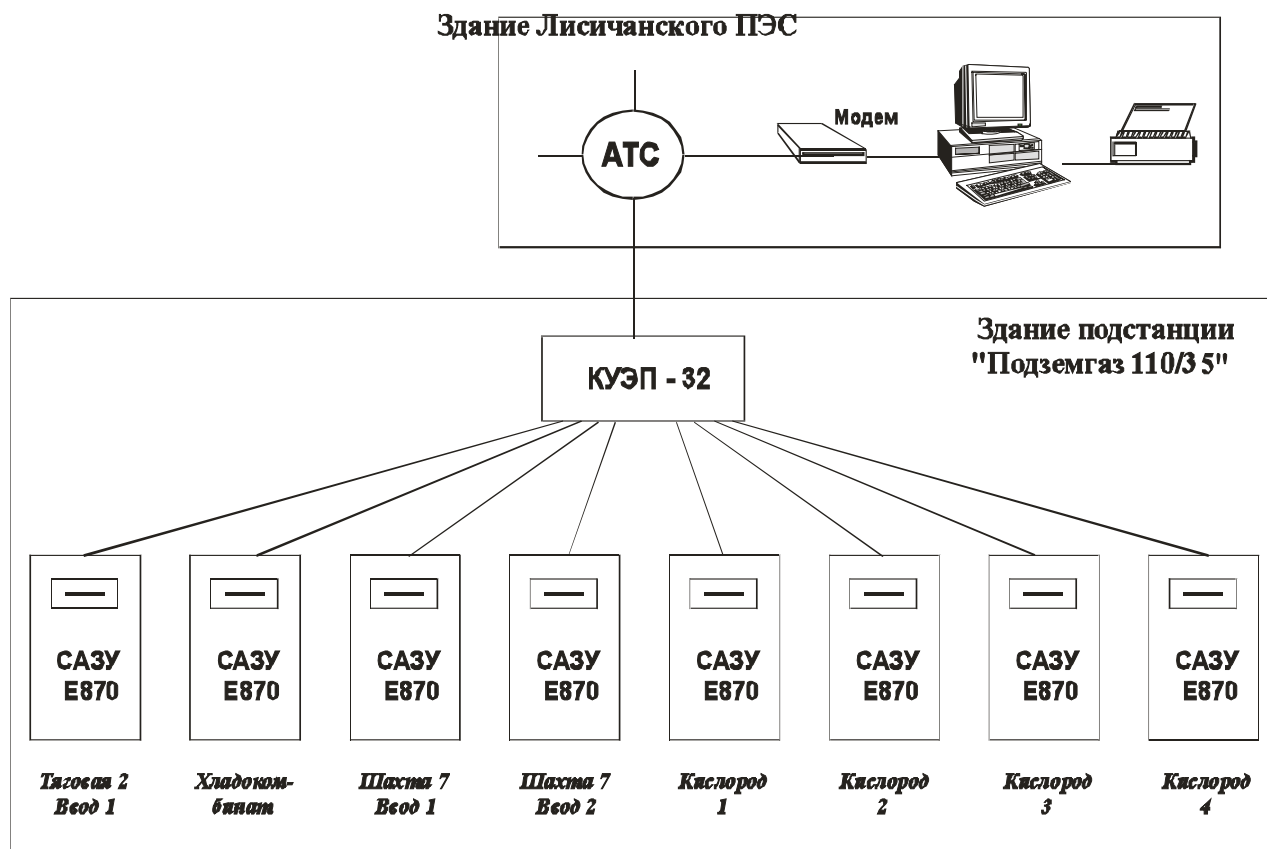
Рис. 6.11. Структурная схема ПТК АСУ ТП ТЭЦ Лучанского сахарного завода.

Система имеет два уровня, связанных между собой по дублированной магистральной промышленной сети МАПС. Нижний уровень реализован на базе заказных конфигураций МСКУ с внутренним резервированием (дублированием) оборудования. МСКУ обеспечивают непосредственное управление энергоагрегатами и их защитами. Верхний уровень представляет собой взаиморезервируемые рабочие места операторов, реализованные на базе ПЭВМ совместимых с IBM PC/AT. На верхнем уровне производится обработка текущей информации, выполняются расчеты оптимального управления энергоагрегатами и ТЭЦ в целом, формируются команды и задания для МСКУ, а также обеспечивается представление информации операторам и ее регистрация.

6.5 Системы энергоучета

6.5.1 Автоматизированная система учёта и контроля электроэнергии на подстанции "Подземгаз 110/35" Лисичанского ПЭС

Система спроектирована на базе программно-технического комплекса учета энергопотребления КУЭП-32 производства СНПО «ИМПУЛЬС». Структурная схема автоматизированной системы учета и контроля электроэнергии на подстанции "Подземгаз 110/35" Лисичанского ПЭС приведена на рис.6.12.



САЗУ - счетчики электроэнергии.

Рис. 6.12. Структурная схема автоматизированной системы учета и контроля электроэнергии на подстанции "Подземгаз 110/35" Лисичанского ПЭС.

КУЭП представляет собой законченное аттестованное изделие с комплектами программного обеспечения и эксплуатационной документации. КУЭП является многоканальным концентратором (на 32 и 64 канала) с центральным процессором MPC 5012A фирмы OCTAGON SYSTEM, USA (программно совместимым с IBM PC). Он обеспечивает радиальное независимое подключение счетчиков электроэнергии:

- индукционных, оснащенных метрологически аттестованными устройствами преобразования оборотов счетного механизма в импульсы тока постоянного напряжения типа УП-3;
- электронных счетчиков всех типов (фирм ABB, Landis&Gyr, ДНЕСТА и др.) через импульсный телеметрический выход.

Функции и технические характеристики КУЭП. КУЭП обеспечивает выполнение следующих основных функций:

- прием и преобразование импульсных сигналов от счетчиков электроэнергии;
- формирование базы данных с привязкой к астрономическому времени;
- хранение накопленной информации о потребленной электроэнергии за текущий и предыдущий месяцы;
- автоматический переход на зимнее и летнее время;
- автоматическую коррекцию хода часов;
- ввод с пульта управления необходимых параметров и уставок;
- многотарифный расчет потребления электроэнергии по каждому счетчику или группе счетчиков для коммерческого учета;
- расчет мощностей для контроля потребления электроэнергии;

- регистрацию внешних и внутренних событий: попытки несанкционированного доступа, вскрытие шкафа, изменение констант, времени, отключение и включение питания, простои, ложные импульсы и др.

- контроль работоспособности технических средств, самодиагностику, метрологическую поверку;

- вывод на индикацию или печатающее устройство результатов расчета и контроля работоспособности по указанию оператора.

По командам, формируемым с пульта управления, КУЭП обеспечивает расчет и выводит на индикацию или печатающее устройство следующие показатели потребления активной и реактивной электроэнергии и мощности по указанному каналу учета (счетчику) или группе учета за прошедший или текущий учетный период: общее потребление электроэнергии, потребление электроэнергии по тарифным зонам, абсолютное и относительное (в процентах) отклонение потребления электроэнергии от лимита, потребление электроэнергии по сменам работы предприятия, среднюю мощность, среднюю мощность по тарифным зонам, абсолютное и относительное (в процентах) отклонение средней мощности от лимита по тарифным зонам, восемь наибольших значений средней получасовой мощности в каждой тарифной зоне с фиксацией даты и времени, текущую мощность, усредненную за 1, 3, 5, 10, 30 минут, мгновенную мощность по каналу учета. С помощью пульта управления КУЭП обеспечивает возможность ввода следующих данных:

- наименование (номер) канала или группы учета;
- тип, заводской номер, емкость счетного механизма, коэффициент пересчета (количество оборотов диска счетчика электроэнергии на 1kW/h), начальные показания счетчиков;
- коэффициент трансформации трансформаторов тока и трансформаторов напряжения;
- границы учетных периодов тарифных зон, смен предприятия;
- лимиты потребления электроэнергии;
- астрономическое время: год, месяц, число, часы, минуты, секунды;
- список паролей, определяющих право и уровень доступа должностных лиц к комплексу.

На индикацию пульта управления КУЭП выводится следующая информация технического контроля: текущие показания счетных механизмов счетчиков (табло); текущий счет импульсов по каналу учета; количество и номера каналов учета, от которых прекратилось поступление счетных импульсов, с указанием даты и времени; отказы электропитания с указанием даты и времени; результаты проверки работоспособности аккумулятора; вскрытие шкафа с указанием даты и времени.

Основные технические характеристики КУЭП-32/64.

- Количество подключаемых счетчиков - 32/64.
- Количество групп учета - до 64.
- Количество тарифных зон - до 4.
- Количество временных зон за сутки - 48.
- Частота входных счетных импульсов - до 20 Hz.
- Последовательный порт - RS-232C (COM1).
- Параллельный порт - Centronics (LPT1).
- Выносной или встроенный пульт управления: индикатор - вакуумно-люминисцентный, 48 знакомест (3x16), знакоместо 5x7 точек, размер индикатора - 203x88 mm, 30 клавиш алфавитно-цифровых, функциональных, управления; интерфейс - RS-232C или ИРПС.

- Электропитание резервированное: два фидера $220\text{ V} \pm 20\%$, $(50 \pm 1)\text{ Hz}$; аккумулятор типа VmV12017 или mV12007 фирмы VARTA. Время сохранения полной работоспособности при пропадании питания от обоих фидеров 220 V – 12 h или 5 h, в зависимости от типа используемого аккумулятора.

- Индивидуальное питание устройств формирования импульсов, установленных в счетчики электроэнергии, напряжение питания - $(12 \pm 1,2)\text{ V}$.

- Гальваническое разделение входных цепей и цепей управления. Электрическая

прочность изоляции между каналами учета и между входными и выходными цепями – 1500 V.

- Защита от помех со стороны подключения датчиков и индикация прохождения счетных импульсов от датчиков.

- Габаритные размеры, масса и потребляемая мощность КУЭП соответствует: 400 х 260 х 460 *mm*, 18 *kg*, 30 V·A - для КУЭП-32; 400 х 260 х 600 *mm*, 25 *kg*, 40 V·A - для КУЭП-64.

В КУЭП предусмотрен режим "черного ящика".

ИНТЕРФЕЙС РЕЗЕРВИРОВАННЫЙ (ИР)

В данном приложении рассмотрена реализация ИР - алгоритмы и дисциплина обмена, логические структуры областей обмена, алгоритмы анализа и принятия решений в зависимости от степени резервирования и др.

А.1 Структура и характеристики ИР

Структура магистралей. Интерфейс резервированный (ИР) является внутрисистемным интерфейсом МСКУ. Он представляет собой магистральный асинхронный интерфейс, предназначенный для обмена информацией между абонентами. Режим обмена – полудуплексный. Абоненты интерфейса могут быть двух типов: ведущий и ведомый. Ведущим называется активный абонент, управляющий интерфейсом. Как правило, им является КМп. Ведомым называется пассивный абонент, выполняющий предписания ведущего. ИР состоит из одной (нерезервированный вариант), двух или трех идентичных магистралей. Часть интерфейса, представленная одной магистралью, называется направлением интерфейса ИР. К каждой магистрали может быть подключен только один ведущий абонент. Количество ведомых абонентов определяется способом конструктивной реализации ИР. Абонент ИР, как ведущий, так и ведомый, может быть однопортовым (нерезервированным), двух и трехпортовым. Каждый порт абонента подключается к одной магистрали ИР. К интерфейсу может быть подключен один трехпортовый ведущий абонент, либо три (два, один) однопортовых ведущих абонента (по одному на каждой магистрали).

Структура абонентов. Ведущий абонент содержит функциональный узел, управляющий магистралями ИР. Ведомый абонент - блок связи с объектом (БСО), может состоять из одного или нескольких ресурсов. Ресурсом БСО называют функционально законченную, логически независимую программно-аппаратную часть, выполняющую определенную функцию. БСО могут быть одноресурсными и многоресурсными. Каждый ресурс имеет набор входных или/и выходных данных.

По способу управления ресурсы делятся на два вида: управляемые одним словом (командой); управляемые массивом (цепочкой) слов (команд). Тип ресурса определяется совокупностью функции, способа управления и структур входных/выходных данных. Однотипными называются ресурсы, у которых совпадают функции, способ управления и структуры входных/выходных данных. Многоресурсный БСО может состоять из нескольких ресурсов как одного, так и разных типов. Каждый из ресурсов такого БСО имеет логический адрес внутри БСО и адресуется со стороны интерфейса ИР специальным полем адреса ресурса в адресном слове.

По способу доступа со стороны интерфейса ИР ведомые абоненты (БСО) делятся на два типа: БСО с произвольным доступом и БСО с протокольным доступом. Произвольный доступ могут иметь, как правило, одноресурсные БСО, или БСО, имеющие несколько однотипных ресурсов, управляемых одним словом (командой). Протокольный доступ могут иметь БСО, содержащие разнотипные ресурсы, управляемые цепочкой команд. Алгоритмы доступов будут рассмотрены далее.

А.2 Физическая среда передачи данных

Физическая среда передачи данных каждого направления ИР представлена идентичными магистралями, обеспечивающими полудуплексный обмен. Каждая из магистралей полностью

автономна по отношению к другим магистралям в части электрической и логической составляющей процесса обмена данными.

Магистраль предназначена для асинхронного параллельного обмена данными между ведущим абонентом и одним из ведомых. Пересылка порции информации по магистрали называется сеансом обмена. Все сеансы обмена по магистрали инициируются ведущим абонентом по отношению к одному из ведомых.

Способ адресации ведомых абонентов - радиальный сигнал выборки. Максимальное количество адресуемых ведомых абонентов - 32. Ведущий абонент удерживает активным сигнал выборки до окончания сеанса обмена. Ведомый абонент не воспринимает сигналы магистрали, если он не выбран.

Магистраль допускает два типа сеансов: одиночный и страничный. Одиночный сеанс обеспечивает пересылку одного информационного слова, а страничный - последовательную пересылку массива слов. Магистраль содержит 23 сигнальных шины и две шины синхронизации (табл. А.1). Тип сигнала приведен по отношению к ведомому абоненту. Активный уровень всех сигналов – низкий.

Таблица А.1

Назначение, тип и характеристики шин магистрали

Наименование шины	Обозначение шины	Тип сигнала
Адрес/данные (16 шин)	Д00-Д15	Вход/выход
Выборка (радиальная, до 32-х)	ВБР1-ВБР32	Вход
Подтверждение выборки	ПВБР	Выход
Запрос данных	ЗД	Вход
Наличие данных	НД	Вход
Подтверждение операции	ПОП	Выход
Прерывание (радиально-магистральная)	ПР	Выход
Работа	РАБ	Вход
Синхронизация (2 шины)	С1, С2	Вход-выход

Узел сопряжения с магистралью ИР обеспечивает независимый между магистралями обмен данными. Узел сопряжения с каждой магистралью содержит регистровые элементы для временного запоминания адресного слова и слова данных при записи со стороны ИР, и занесения слова данных, соответствующего адресному слову, при чтении со стороны ИР. БСО, поддерживающие доступ только по чтению или только по записи, имеют соответственно один регистр хранения слова данных.

А.3 Конструктивная реализация интерфейса ИР

Реализация ИР. ИР реализован на объединительной панели (многослойная печатная плата) в виде магистральных связей между 96-контактными соединителями типа СНП-59. Информационные связи реализуются печатными проводниками. Допускается комбинированная реализация информационных связей.

Размещение соединителей на объединительной плате обеспечивает установку модулей типа Е2 с двумя соединителями с шагом не менее 20 *mm*. Допустимое количество установочных мест интерфейса - не более 32. Допускается реализация объединительной платы для нерезервированного варианта интерфейса ИР с одним рядом соединителей (одна магистраль). Требования к шагу установки и количеству мест при этом не изменяются. Предельная суммарная длина информационных связей интерфейса – не более 1500 *mm*.

А.4 Дисциплина и алгоритмы сеансов обмена

Сеансы обмена магистрالي. Сеансом обмена называют пересылку одной порции данных. Минимальной порцией данных является 16(8)-разрядное слово. Сеанс, при котором осуществляется пересылка одного слова, называют одиночным. Сеанс, при котором осуществляется пересылка массива слов, называют страничным. Размер страницы не ограничен и определяется типом абонента, поддерживающего страничный обмен. Сеанс обмена всегда инициируется ведущим абонентом по отношению к одному из ведомых и состоит из трех этапов: инициализация, пересылка, завершение (рис. А.1). Ведущий абонент выдает на шины данных адресное слово, а затем устанавливает в активное состояние сигнал ВЫБОРКА абонента. После этого анализирует состояние сигнала ПОДТВЕРЖДЕНИЕ ВЫБОРКИ. Появление активного уровня данного сигнала завершает этап инициализации сеанса обмена (ведомый абонент воспринял начало сеанса и готов к пересылке данных).

Организация этапа пересылки зависит от направления передачи порции данных и типа сеанса - одиночного или страничного. При пересылке из ведущего в ведомый, ведущий абонент выдает слово данных на шины данных, затем устанавливает активный уровень сигнала НАЛИЧИЕ ДАННЫХ. После этого ожидает активный уровень сигнала ПОДТВЕРЖДЕНИЕ ОПЕРАЦИИ, завершающий этап пересылки. При пересылке из ведомого в ведущий абонент устанавливает активный уровень сигнала ЗАПРОС ДАННЫХ и ожидает активный уровень сигнала ПОДТВЕРЖДЕНИЕ ОПЕРАЦИИ, свидетельствующий о выдаче ведомым абонентом слова данных на шину данных. При выполнении этапа пересылки, время распространения сигналов на шинах данных учитывает ведущий абонент – сигнал НАЛИЧИЕ ДАННЫХ устанавливается после выдачи слова данных через интервал времени, необходимый для их распространения. Прием слова данных с шин после установления ведомым абонентом сигнала ПОДТВЕРЖДЕНИЕ ОПЕРАЦИИ в ответ на ЗАПРОС ДАННЫХ выполняется через такой же интервал времени. Установка активного уровня сигнала ПОДТВЕРЖДЕНИЕ ОПЕРАЦИИ завершает этап пересылки слова данных.

При выполнении страничного сеанса ведущий абонент, восприняв активный уровень сигнала ПОДТВЕРЖДЕНИЕ ОПЕРАЦИИ, устанавливает пассивный уровень сигнала НАЛИЧИЕ ДАННЫХ (или ЗАПРОС ДАННЫХ) и ожидает установления пассивного уровня сигнала ПОДТВЕРЖДЕНИЕ ОПЕРАЦИИ ведомым абонентом. После этого ведущий абонент повторяет последовательность сигналов для пересылки очередного слова данных. Переход сигналов НАЛИЧИЕ ДАННЫХ или ЗАПРОС ДАННЫХ в пассивное состояние инициирует подготовку интерфейсного узла к приему (выдаче) следующего слова данных (инкремент/декремент текущего адреса, сдвиг вершины стека и т.п.). По завершении этого процесса ведомый абонент устанавливает в пассивное состояние сигнал ПОДТВЕРЖДЕНИЕ ОПЕРАЦИИ, тем самым, сообщая ведущему абоненту о готовности к продолжению пересылок.

Этап завершения сеанса инициируется ведущим абонентом установлением пассивного уровня сигнала ВЫБОРКА. Ведомый абонент непрерывно анализирует состояние сигнала ВЫБОРКА и при появлении пассивного уровня сигнала перевести интерфейсные схемы в состояние готовности к новому сеансу обмена. Дисциплина сеанса магистрали. Шины адреса/данных (Д15–Д00) магистральные, мультиплексированные. В первой части сеанса служат для пересылки адресной информации в ведомый абонент, во второй части сеанса - для пересылки слова данных. Младший разряд слова данных пересылается по шине Д00. Адресная информация используется для дешифрации обращения к минимальной единице информации внутри БСО (байт, слово) в соответствии с таблицей А.2. Отдельные субполя могут использоваться для адресации ресурсов и их структурных элементов в многоресурсных БСО. БСО обеспечивает защелкивание адресной информации до формирования сигнала ПОДТВЕРЖДЕНИЕ ВЫБОРКИ и хранение до окончания сеанса обмена.

Формат адресного слова магистрали

Д1 5	Д1 4	Д1 3	Д1 2	Д1 1	Д1 0	Д9	Д8	Д7	Д6	Д5	Д4	Д3	Д2	Д1	Д0
Поле адреса						Неопределенное состояние					Поле адреса				

Шина выборки ВЫБОРКА предназначена для адресации ведущим абонентом одного из ведомых абонентов во время сеанса. Шины выборки – радиальные. Интерфейсный порт ведущего абонента имеет не менее 32 шин выборки. Ведущий абонент формирует сигнал по шине выборки и удерживает его до конца сеанса обмена. Переход сигнала шины в активное состояние осуществляется не ранее установления активных сигналов на шинах данных с учетом времени распространения. Переход сигнала на шине выборки в пассивное состояние осуществляется не ранее перехода в пассивное состояние сигналов на шинах НАЛИЧИЕ ДАННЫХ и ЗАПРОС ДАННЫХ. Шина подтверждения выборки ПОДТВЕЖДЕНИЕ ВЫБОРКИ информирует ведущего абонента о том, что адресуемый ведомый абонент готов к началу сеанса обмена. Сигнал на шине формируется ведомым абонентом, переход сигнала в активное состояние осуществляется не ранее запоминания адресной информации для продолжения сеанса обмена. Шина ЗАПРОС ДАННЫХ информирует ведомого абонента о необходимости выдать заказанное слово данных на шины данных. Сигнал на шине формируется ведущим абонентом, переход сигнала в активное состояние осуществляется не ранее появления активного сигнала на шине ПОДТВЕЖДЕНИЕ ВЫБОРКИ. Шина НАЛИЧИЕ ДАННЫХ информирует ведомого абонента о наличии на шинах данных информации. Сигнал на шине формируется ведущим абонентом, переход сигнала в активное состояние осуществляется не ранее установления активных сигналов на шинах данных с учетом времени распространения. Шина ПОДТВЕЖДЕНИЕ ОПЕРАЦИИ информирует ведущего абонента о том, что ведомый абонент готов завершить текущий сеанс. Сигнал на шине формируется ведомым абонентом в ответ на сигналы НАЛИЧИЕ ДАННЫХ или ЗАПРОС ДАННЫХ. Переход сигнала по шине в активное состояние осуществляется не ранее полного завершения всех операций, предписанных сигналами НАЛИЧИЕ ДАННЫХ и ЗАПРОС ДАННЫХ. Сигнал формируется за время не более 200 мкс после поступления сигнала ВЫБОРКА. Шина ПЕРЕРЫВАНИЕ предназначена для передачи сигнала прерывания от ведомого абонента ведущему. Алгоритм формирования прерывания определяется назначением и типом конкретного абонента и оговорен в технической документации. Физическая реализация выхода прерывания в ведомых абонентах имеет тип "открытый коллектор". Физическая реализация входа прерывания в ведущем абоненте обеспечивает прием сигнала типа "открытый коллектор". Шина РАБОТА предназначена для информирования ведомых абонентов о наличии и работоспособности ведущего абонента на магистрали. Ведомый абонент имеет не более одного входа сигнала РАБОТА с каждой магистралью. Реализация входа обеспечивает формирование пассивного уровня сигнала РАБОТА при отсутствии источника сигнала (ведущего абонента).

Синхросигналы СС1, СС2 предназначены для синхронизации работы специализированных БСО.

Алгоритмы обмена. Два типа БСО по способу доступа определяют два вида алгоритмов обмена: обмен с произвольным доступом и обмен с протокольным доступом.

Алгоритм обмена с произвольным доступом предполагает пересылку данных в течение одного или нескольких сеансов. Алгоритм обмена с произвольным доступом предполагает прямую адресацию каждой порции данных в БСО полем адресного слова. Алгоритм обмена с протокольным доступом предполагает диалоговый обмен сообщениями между ведущим и ведомым абонентом.

А.5 Логический протокол обмена

Область обмена БСО. Под областью обмена подразумевается логическая область оперативной памяти ведомого абонента (БСО), предназначенная для временного размещения получаемой из ИР информации (команд, данных) для последующего анализа, а также для временного размещения данных и служебной информации (для последующей выдачи в ИР). В зависимости от типа и назначения БСО некоторые компоненты могут отсутствовать, некоторые поля - не использоваться. Область обмена БСО (согласно рисунку А.2) состоит из нескольких структур с определенным назначением. Каждая структура состоит из полей, приведенных на рис. А.2.

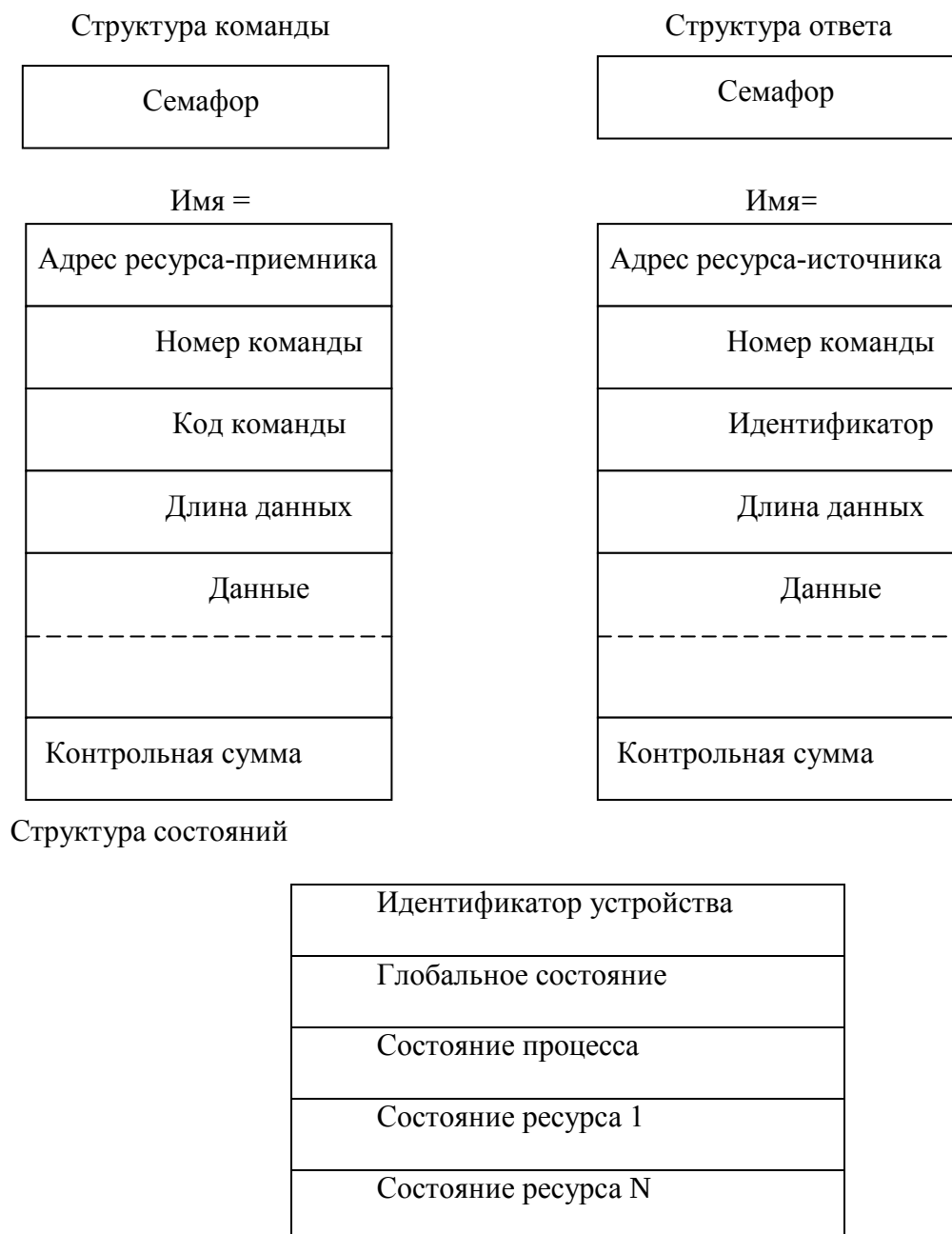


Рис. А.2 - Структуры области обмена БСО

Поле структуры команд. Адрес ресурса-приемника (1 байт) указывает, какому из ресурсов БСО предназначена команда (если ресурсов несколько). Нулевое значение означает глобальную команду (касающуюся всех ресурсов). Номер команды (1 байт) определяет

порядковый номер команды для данного БСО. Используется в алгоритмах обработки команд. Формируется циклически путем наращивания на единицу после формирования каждой команды. Код команды (1 байт) предписывает БСО выполнение определенных действий над ресурсом. Содержит субполя, определяющие способ выполнения и обработки команды (мажорирование, время жизни и т.п.). Формат поля определяется конкретным типом БСО. Длина данных (1 байт) и данные (N байт), определяют длину поля данных в байтах. Нулевое значение означает отсутствие данных. Контрольная сумма (1 байт) содержит значение контрольной суммы всех полей структуры.

Поле структуры ответа. Адрес ресурса-источника (1 байт) определяет ресурс БСО, ответ от которого содержит структура. Номер команды (1 байт) содержит номер команды, ответ на которую содержит структура. Идентификатор данных (1 байт) характеризует тип порции данных, содержащихся в структуре (назначение, количество порций, номер порции и т.п.). Длина данных (1 байт), данные (N байт) и контрольная сумма (1 байт) содержат значения аналогично структуре команды.

Поле структуры состояний. Идентификатор устройства (4 байта). Содержит информацию, однозначно характеризующую тип БСО. Может содержать субполя, характеризующие отдельные параметры БСО (в зависимости от типа БСО): уникальный цифровой шифр БСО; байта индивидуальных характеристик БСО (типы и количество ресурсов (каналов), режимы работы и т.п.). В БСО идентификатор располагается в начальных адресах области обмена.

Глобальное состояние (2 байта). Содержит информацию, характеризующую состояние модуля: результаты стартовой диагностики, признаки работоспособности узлов и т.п. Формат поля определяется типом конкретного БСО: обобщенный признак работоспособности и признаки работоспособности внутренних узлов (ресурсов); признак работоспособности интерфейсных узлов (текущий способ обработки команд (степень резервирования), состояние сигналов «РАБОТА» и т.п.).

Состояние процесса (5 байт). Поле содержит динамическую информацию о текущем состоянии БСО: код выполняемого в данный момент процесса (внешняя команда или один из внутренних процессов); код последней внешней команды, взятой на выполнение; номер команды; состояние выполнения последней внешней команды (текущий этап обработки); код завершения выполнения последней внешней команды.

Состояние ресурса. Поле содержит динамическую информацию о текущем состоянии каждого из ресурсов БСО, если таковые имеются. Формат поля определяется типом конкретного БСО и типом ресурса: код типа ресурса; текущий режим работы; динамически изменяющиеся свойства ресурса.

Структура команды и структура ответа являются структурами с синхронным доступом. Поле семафора в этих структурах служит для синхронизации доступа. Форматы управляющих полей всех структур оговариваются в технической документации конкретного БСО.

Доступ к структурам области обмена в режиме прямого доступа в память. Структуры команды и ответа имеют жесткую дисциплину доступа. Структура команды записывается со стороны интерфейса ИР, считывается со стороны БСО. Структура ответа записывается со стороны БСО, считывается со стороны интерфейса ИР. Структура состояний записывается со стороны БСО, считывается со стороны интерфейса ИР. Содержимое полей структуры состояний носит информационный характер, доступ к структуре - произвольный. Для БСО, предназначенных для работы с магистралью в режиме прямого доступа в память, алгоритм доступа к структурам команд и данных обеспечивают бесконфликтный доступ ко всем полям структуры. Дисциплина доступа обеспечивается полем семафора. Семафор имеет два состояния - "полон" и "пуст". Состояние "полон" означает: для записи в структуру: не обработана предыдущая запись, доступ невозможен; для чтения структуры: произведена новая запись после предыдущего чтения, доступ возможен. Состояние "пуст" означает: для записи в структуру: предыдущая запись обработана, доступ возможен; для чтения структуры: не произведена новая запись, доступ невозможен.

Алгоритм семафорного доступа к структуре представлен схемами алгоритмов, приведенными на рис. А.3.

Организация протокольного обмена. Типы кадров. Протокольный обмен определяет дисциплину взаимодействия абонентов магистрали и структуру кадров, пересылаемых в течение сеанса обмена. Все кадры делятся на следующие типы: командный кадр, инициативный кадр, информационный кадр, квитирующий кадр.

Командный кадр, информационный и квитирующий кадр являются составными кадрами любого сеанса обмена. Для ИР протокольный обмен реализуется через область обмена БСО. При пересылке данных в БСО командный и информационный кадры составляют поле структуры команд. Квитирующий кадр управляет установкой семафора доступа к соответствующей структуре. При пересылке данных из БСО командный кадр аналогичен анализу готовности данных для чтения, информационный кадр составляет поле структуры ответа, квитирующий кадр - установка семафора доступа к структуре данных. Сигнал прерывания от БСО аналогичен инициативному кадру.

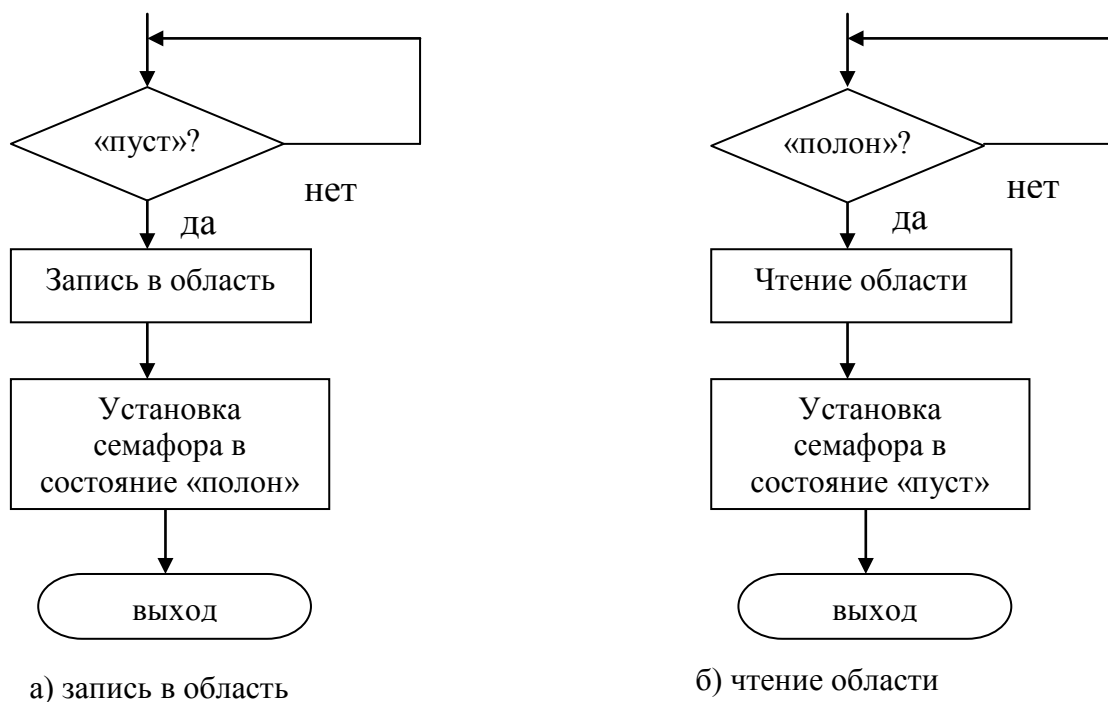


Рис. А.3. Алгоритмы семафорного доступа.

А.6 Дисциплина формирования и обработки команд. Резервирование

Формирование команд. Формирование любых команд выполняет только ведущий абонент по отношению к ведомым. Ведущий абонент не формирует очередную команду ведомому абоненту (БСО) до завершения ведомым абонентом выполнения предыдущей команды. Если ведомый абонент содержит несколько функционально независимых ресурсов, данное требование применяется к каждому из ресурсов, а не к абоненту в целом. Ведущий абонент присваивает каждой команде индивидуальный порядковый номер в диапазоне 0–255. Одна и та же команда, формируемая резервированными абонентами, имеет один и тот же номер.

Перед формированием очередной команды ведущий анализирует состояние ведомого абонента.

Прием команд. Прием и передача на обработку команд осуществляется ведомым абонентом в соответствии с функциональным назначением. Прием очередной команды, как правило, квитируется. Исключение составляют вспомогательные команды (опрос состояния и т.п.). Для обеспечения детерминированного процесса приема и обработки команд БСО используются функционально независимые области в соответствии с рис. А.4: область обмена (ОО); область временного хранения команд (ОВХ); область обработки команд (ООК).

Команда, поступившая в область обмена ООх, анализируется на достоверность (контрольная сумма и т.п.) и переносится в область временного хранения (если она свободна). Область временного хранения индивидуальная для каждой из магистралей. Состоит из структуры команды и структуры состояния. Структура команды содержит все поля структуры команды области обмена, кроме контрольной суммы. Структура состояния содержит семафор, состоящий из поля, квитирующего передачу команды в область обработки (ОВХ свободно для очередной команды) и поля, квитирующего поступление очередной команды из области обмена в ОВХ (очередная команда ожидает обработку). Указанные поля семафора используются в алгоритмах обработки команды.

БСО не принимает на обработку и квитируание очередную команду до завершения обработки предыдущей.

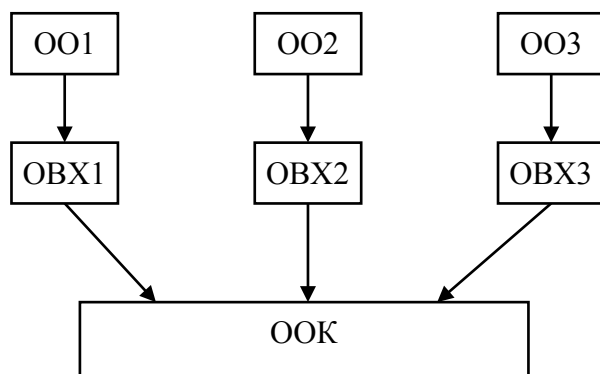


Рис. А.4. Организация областей для приема и обработки команд.

Абонент (БСО), имеющий несколько функционально независимых ресурсов, принимает и обрабатывает команды для каждого из ресурсов в отдельности. При этом БСО имеет одну область обмена с интерфейсом ИР и отдельные области обработки команд для каждого из ресурсов. Перемещение очередной команды из области обмена в область обработки соответствующего ресурса является началом обработки команды, отражается в структуре состояний и квитируется (при необходимости). Структура состояний абонентов с несколькими ресурсами содержит отдельные идентичные поля для каждого из ресурсов. Процедура приема команды завершается переносом команды в область временного хранения. Область обмена при этом освобождается для очередных команд.

Обработка команд. Резервирование. Для БСО с произвольным доступом процесс обработки команд отсутствует, так как при обращении со стороны интерфейса ИР данные помещаются непосредственно в исполнительные узлы или выбираются из исполнительных узлов. При этом исполнительный узел, формирующий выходное воздействие, обеспечивает побитную обработку помещаемых в него порций данных из каждой магистрали по принципу «два из трех». Активный сигнал «РАБ» каждой из магистралей разрешает участие каждого бита в конкурсе. Пассивное состояние сигнала «РАБ» всех магистралей вызывает формирование предопределенного состояния каждого бита. Значения предопределенного состояния устанавливается программно или аппаратно.

Для БСО с протокольным доступом все этапы обработки каждой команды и результат выполнения динамически отображается в соответствующих полях структуры состояния области обмена. Обработка команды состоит из четырех этапов: 1 - пересылку команды из ОВХ

в область обработки команд (ООК); 2 - анализ достоверности; 3 - выполнение предписаний команды; 4 - завершение обработки.

Этап анализа достоверности выполняется только при резервированной текущей конфигурации ИР и заключается в сравнении команд с одним номером, поступивших на обработку из разных магистралей ИР. Этап завершения обработки заключается в формировании признаков результата выполнения команды в соответствующих полях области состояния (квитирования выполнения команды).

Алгоритм обработки команд зависит от следующих факторов: текущей конфигурации интерфейса ИР (наличие признака работоспособности каждой из магистралей интерфейса) и предписания способа обработки в соответствующем поле самой команды.

Текущая конфигурация интерфейса определяется по состоянию сигнала «РАБ» каждой из магистралей. Сигнал «РАБ» всех магистралей контролируется при получении любой команды по любой из магистралей ИР. При отсутствии сигналов «РАБ» всех магистралей ведомый абонент принимает на обработку только команды, имеющие информационный характер. Любые команды, предписывающие формирование выходных воздействий, игнорируются.

Алгоритм анализа при обработке команд не опирается на регламентируемые временные интервалы (тайм-ауты). Все ветвления алгоритма анализа носят событийный характер. Процесс обработки команды начинается переносом очередной команды с номером N из области временного хранения (ОВХ) любой из магистралей, в которой команда с номером N появилась первой. Процесс обработки очередной команды не начинается до завершения процесса обработки предыдущей команды. После переноса команды N в область обработки, ОВХ освобождается для приема очередной команды (N+1). Этап анализа любой команды при троированной текущей конфигурации интерфейса ИР осуществляется по принципу «два из трех». Алгоритм анализа при троированной конфигурации интерфейса:

- после приема команды по какой-либо магистрали ИР с активным сигналом «РАБ» выполняется анализ текущей конфигурации и при наличии еще хотя бы одной работоспособной магистрали – выполняется ожидание команды с таким же номером из других работоспособных магистралей;

- после приема второй команды с тем же номером, выполняется процедура полного сравнения полей двух команд. При полном сравнении команда передается на выполнение. При несравнении выполняется ожидание третьей команды с тем же номером;

- при получении третьей команды выполняется взаимное сравнение полей всех трех команд и при совпадении двух, команда передается на выполнение;

- при несовпадении трех команд с одинаковым номером формируется признак ошибки приема команды с этим номером;

- при поступлении в область обмена двух направлений ИР команды с очередным номером этап анализа предыдущей команды завершается формированием признака ошибки приема команды;

- если при поступлении трех команд с одним номером одна из команд не совпадает с остальными, формируется признак недостоверности команды с этим номером в соответствующей области структуры состояния соответствующей магистрали.

Этап анализа для дублированной текущей конфигурации интерфейса ИР выполняется по принципу приоритетной магистрали. По умолчанию приоритетной считается магистраль с меньшим номером (первая из первой и второй, вторая из второй и третьей, третья из первой и третьей). Команда с номером N, поступившая из приоритетной магистрали всегда выполняется. При этом процедура ожидания команды с номером N из второй магистрали выполняется до поступления в ОВХ приоритетной магистрали команды с номером N+1. При поступлении команды с номером N из второй магистрали выполняется процедура полного сравнения и при несравнении формируется признак нарушения передачи команды с номером N в соответствующих полях структуры состояний. Несмотря на это, выполняются предписания команды N, поступившей из приоритетной магистрали.

Этап анализа для нерезервированной текущей конфигурации ИР заключается в проверке текущей конфигурации, и если она не изменилась – выполняются предписания поступившей команды.

Для некоторых типов БСО допускается командное предписание режима обработки. При этом ведущий абонент указывает данному БСО номер приоритетной магистрали. БСО выполняет все команды из приоритетной магистрали по алгоритму нерезервированного интерфейса.

Выполнение команд осуществляется согласно функциональному назначению абонента (БСО).

А.7 Диагностирование ИР

Диагностирование магистрали. Функция диагностирования поддерживается всеми типами ведущих абонентов. Диагностирование может выполняться как в процессе функционирования (фоновое), так и в виде специальной процедуры. Обязательным является выполнение процедуры диагностирования при включении в работу ведущего абонента магистрали до установки в активное состояние сигнала «РАБ».

Узел управления ИР ведущих абонентов обеспечивает автоматическую (без подключения дополнительного оборудования) проверку шин адреса-данных и управления (за исключением шин выборок) и распознавание неисправностей следующих типов: замыкание на «землю»; замыкание на «+ 5 V»; замыкание между шинами в любых сочетаниях. Для шины «РАБ» обеспечивается контроль нарушений следующих типов:

- активный уровень сигнала на шине «РАБ» до логического включения;
- отсутствие сигнала на шине «РАБ» после логического включения.

При выполнении фоновой диагностики шин магистрали обеспечивается пассивный уровень всех сигналов выборок. При обнаружении любых неисправностей шин магистрали ведущий абонент устанавливает пассивный уровень сигнала «РАБ», при этом не допускается обмен с абонентами магистрали.

Специальная процедура диагностирования ИР включает следующие виды проверок:

- проверку шин адреса-данных и управления;
- проверку времени срабатывания тайм-аута на ожидание ответа от БСО (длительность сеанса при обращении к неисправному или отсутствующему БСО);
- измерение времени сеанса обмена с любым БСО.

АРХИТЕКТУРА СЕТИ МАПС

В данном приложении рассмотрены особенности архитектуры сети МАПС – промышленной локальной сети МСКУ М.

Б.1 Основные определения

Б.1.1 Сеть МАПС и ее компоненты

Магистралью МАПС называется совокупность программно-аппаратных средств, включающая в себя КСв - контроллеры связи (с внутренним программным обеспечением), обеспечивающие управление магистралью передачи данных и взаимодействующие между собой абоненты: МСКУ - микропроцессорные субкомплексы контроля и управления, рабочие станции ПС 5110, персональные ЭВМ и т.п.

Сетью МАПС называется одна, две или три магистрали МАПС, абоненты которых взаимодействуют между собой. Магистрали в составе сети нумеруются числами 1, 2 и 3.

Адресом абонента в МАПС является целое число в пределах от 0 до 255 включительно, причем, адрес, равный 255, означает одновременную адресацию всех абонентов магистрали. Допустимыми значениями адреса являются 0-62 и 255; таким образом, максимальное количество абонентов на магистрали - 63.

Любой абонент магистрали может иметь, помимо выхода на данную магистраль, до двух дополнительных портов, предназначенных для связи с другими магистралями сети (например, в дублированных и троированных комплексах). Эти порты, включая выход на данную магистраль, называются *"шлюзами"* и нумеруются числами 1, 2, 3, помещаемыми в соответствующие двухразрядные поля. Значение 00 в этих Полях означает, что сообщение относится непосредственно к данному абоненту. Исходное сообщение в этом случае может быть передано через любой шлюз абонента. Исходные сообщения с ненулевым значением в данных разрядах передаются по магистрали, к которой подключен соответствующий шлюз абонента, но могут при необходимости передаваться по другой магистрали (например, при отказе первой). Ответные сообщения всегда передаются по той магистрали, по которой было принято исходное сообщение. Как правило, шлюзы обеспечивают выход на разные магистрали одной сети, при этом адреса их на магистралях должны совпадать. Кроме того, если абонент подключен к магистралям сети и имеет адрес, равный i , то на магистралях сети, к которым он не подключен, значение i не должно использоваться в качестве адреса подключения какого-либо абонента. Таким образом, максимальное количество абонентов в сети совпадает с максимальным количеством абонентов на магистрали, равным 63. Количество шлюзов у абонента и количество магистралей, к которым подключен абонент, может не совпадать. Так, абонент с тремя шлюзами может быть подключен к одной, либо двум магистралям. В этом случае сообщение любому шлюзу абонента передается по одной из магистралей, к которым он подключен, а передача сообщения тому шлюзу, которому оно адресовано, возлагается на самого абонента.

Основным (активным) арбитром на магистрали называется КСв, который фактически выполняет управление магистралью.

Резервным (пассивным) арбитром на магистрали называется КСв, который не выполняет управление магистралью, а анализирует работоспособность основного арбитра и, в случае выявления неработоспособности последнего, заменяет его.

Приоритетными абонентами называются абоненты, которые могут выдавать приоритетные (срочные) сообщения или запросы. Таким абонентам магистраль предоставляется в первую очередь.

Неприоритетными абонентами называются абоненты, которые выдают только неприоритетные сообщения и запросы. Им магистраль предоставляется в случае, когда осталось время после обслуживания всех заявок от приоритетных абонентов.

Инициативным абонентом называется абонент, которому в любой момент времени может потребоваться магистраль для передачи сообщений.

Пассивным абонентом называется абонент, который осуществляет только прием сообщений и выдачу ответных сообщений на запрос без разрыва связи.

Активизируемый абонент - это абонент, который в исходном состоянии является пассивным, но переходит в активное состояние (идентичное состоянию инициативного абонента) при получении различных команд (сообщений, предусматривающих вывод ответных сообщений) и в некоторых других случаях. Таким образом, абоненты МАПС разделяются на следующие типы: арбитр сети; приоритетные инициативные абоненты; инициативные неприоритетные абоненты; приоритетные активизируемые абоненты; неприоритетные активизируемые абоненты и пассивные абоненты.

Возможно динамическое изменение типа абонента. Тип абонента определяется в процессе проектирования соответствующих технических средств. Так, для МСКУ в настоящее время допустимыми являются два типа: инициативный приоритетный и инициативный неприоритетный, что определяется при генерации управляющей системы МСКУ.

Если в сети МАПС имеется более одной магистрали и необходимо обеспечивать взаимодействие между абонентами, подключенными к разным магистралям сети, то в состав сети должны входить так называемые **межмагистральные адаптеры (ММА)**. ММА - это устройство, которое, во-первых, подключено к двум или трем магистралям сети и имеет возможность передавать данные между своими шлюзами, во-вторых, осуществляет прием и обработку сообщений, предназначенных для взаимодействия абонентов разных магистралей. Функции ММА могут выполнять только арбитры магистралей, передавая сообщения из одной магистрали в другую.

Б.1.2. Общая структура передаваемых по сети МАПС сообщений

Взаимодействие между абонентами сети осуществляется посредством передачи различных сообщений, описанных ниже. Сообщением называется последовательность байтов (рис. Б.1), передаваемая между абонентами в одном сеансе обмена. Преамбула сообщения состоит из байтов, содержащих: коды синхронизации, равные 01111110 (один или более); адрес абонента, которому передается данное сообщение (адрес абонента-приемника); если в этом байте указан код 11111111, то данное сообщение адресуется всем абонентам магистрали и называется циркулярным; управляющий байт, определяющий тип и логическую структуру данного сообщения; значения этого байта имеют структуру, описанную ниже; код, содержащий в разрядах 5-0 адрес абонента, от которого поступило данное сообщение (адрес абонента-источника), а в разрядах 7-6 - номер шлюза, из которого принято сообщение.

преамбула сообщения
Информационная часть сообщения (от 1 до 256 байтов)
Постамбула сообщения (не менее 3 байтов)

Рис. Б.1. Общая структура сообщения.

Количество и назначение последующих байтов преамбулы определяется значением соответствующих полей в управляющем байте. Если седьмой разряд управляющего байта равен 0 (такое сообщение называется коротким), то преамбула может не содержать более ни одного байта, либо содержать еще 4 или 6 дополнительных байтов, наличие и назначение которых определяется при описании данных сообщений. В коротких сообщениях отсутствует информа-

ционная часть. Если седьмой разряд управляющего байта равен 1, то далее в преамбуле идут байты, содержащие длину информационной части сообщения в байтах, если в информационной части меньше 256 байтов, или код 0, если информационная часть содержит 256 байтов и четырехбайтный идентификатор сообщения, структура которого описана ниже; это поле имеет место в преамбуле только при наличии соответствующего признака в управляющем байте.

Постамбула сообщения состоит как минимум из трех байтов, первые два из которых обозначаются КБ1 и КБ2 соответственно и содержат контрольную сумму байтов сообщения. КБ1 и КБ2 формируются в абонентах МАПС, которые посылают сообщения, и проверяются в абонентах-приемниках сообщений. Значения КБ1 и КБ2 формируются на основании циклического кода с производящим полиномом $X^{16}+X^{12}+X^5+1$. остальные байты постамбулы - коды синхронизации, равные 01111110.

Б.1.3. Структура управляющего байта

Для коротких сообщений управляющий байт имеет структуру, приведенную на рис. Б. 2.

0	1	2	3	4	7
О	ПП	НШ	КФ		

Рис. Б.2. Структура управляющего байта для коротких сообщений.

Таблица Б.1

Функциональное назначение коротких сообщений

Мнемоническое обозначение функции	Функциональное назначение коротких сообщений	Двоичное значение КФ
ПМ	Предоставление магистральной абоненту-адресату	000000
ВМ	Возврат магистральной	000001
* ВМД	Возврат магистральной с деактивизацией	000011
ЗН	Отказ от приема сообщения из-за занятости абонента	001000
* ПТЖ	Подтверждение приема сообщения	000101
* ПТЖА	Подтверждение приема сообщения и начала его обработки с указанием, что абонент, пославший данное короткое сообщение, должен перейти в активное состояние	001101
* ОШ	Подтверждение приема сообщения, но отказ от его обработки из-за наличия логической ошибки в нем	000110
* ОШН	Подтверждение приема сообщения, но отказ от его обработки из-за неработоспособности адресата	000010
* ЗП	Запрос сообщения, на которое ранее было выдано сообщение ЗН, от абонента-адресата	000111
ЗС	Запрос состояния абонента (в преамбуле данного сообщения обязательно присутствуют 6 дополнительных байтов, формат которых описан в п.Б.2.2 данного документа)	001000
* ОЗС	Сообщение, передаваемое в ответ на сообщение типа ЗС (в преамбуле данного сообщения обязательно присутствуют 4 дополнительных байта, формат которых описан в п.2.2 данного документа)	001001

ПП - признак повторной передачи сообщения (ПП=0 - сообщение передается в первый раз, ПП=1 - сообщение передается повторно), КФ - код функции (табл. Б.1).

0	1	2	3	4	5	6	7
1	ПП	НШ	Ф	ТС	ПР		

Рис. Б.3. Структура управляющего байта для длинных сообщений.

На рис. Б.3 обозначено: ПП - признак повторной передачи сообщения (ПП=0 - сообщение передается в первый раз, ПП=1 - сообщение передается повторно); НШ - номер шлюза, в который должно быть выдано сообщение; Ф - признак наличия идентификатора в преамбуле сообщения (Ф=0 - идентификатор считается опущенным, Ф=1 - идентификатор имеет место); ТС - код, определяющий тип сообщения (ТС=00 - сообщение, не требующее ответа и подтверждения, если Ф=0, то значение идентификатора предполагается равным 10100000, 10100000, 00000000, 00000000; ТС=01 - сообщение, требующее подтверждения, но не требующее ответа, если Ф=0, то значение идентификатора предполагается равным 10100000, 10100000, 00000000, 00000000; ТС=10 - сообщение, не требующее ответа и подтверждения, если Ф=0, то значение идентификатора предполагается равным 10100001, 10100000, 00000000, 00000000; ТС=11 - сообщение, содержащее последнюю порцию данных и требующее ответ, если Ф=0, то значение идентификатора предполагается равным 10100000, 10100000, 00000000, 00000000); ПР - признак приоритетности сообщения (ПР=0 - сообщение не приоритетное, ПР=1 - сообщение приоритетное).

Примечания:

1. Активизируемый абонент переводится в активное состояние в случаях, если он выдал короткое сообщение типа ПТЖА, а также если в ответ на любое сообщение он выдал сообщение типа ЗН. Отмена состояния активности выполняется тогда, когда у абонента не осталось ни одной невыполненной заявки на выдачу ответа. Об этом он извещает активного арбитра магистральной, выдавая в конце сеанса сообщение типа ВМД.

2. Для циркулярных сообщений ТС=00, ПП=00, НШ=00.

3. Короткие сообщения типа ПМ, ЗП и ЗС требуют выдачи ответного сообщения допустимого типа в том же сеансе связи, остальные короткие сообщения, приведенные в табл. Б.1, не требуют выдачи ответа или подтверждения.

Б.1.4 Структура идентификатора

Идентификатор в сообщениях состоит из четырех байтов, значения которых определяют особенности приема и обработки сообщений. Структура идентификатора приведена на рис. Б.4.

7	6	5	4	3	2	1	0				
Код уровня приемника (КУП)				Код вида сообщения (КВС)							
1	Код уровня источника (КУИ)			Код модификации (КМ)							
2	Адрес второго уровня абонента приемника (АВУП)										
3	Адрес второго уровня абонента источника (АВУИ)										

Рис. Б.4. Структура идентификатора.

Код уровня приемника (КУП) определяет, какому из объектов абонента адресуется данное сообщение. Значения КУП, варьируемые в пределах 0 - 15, разбиваются на следующие группы:

- 0 - сообщение адресуется связному оборудованию, обеспечивающему выход абонента на магистраль МАПС (как правило, КСв);

- 1 - сообщение адресуется программным средствам, обеспечивающим обмен данными между связным оборудованием и функциональными программами абонента (например, драйверу, управляющему работой КСв);

- 2 - 3 - резерв;

- 4 - 5 - сообщение адресуется программным средствам, предназначенным для удаленной отладки, настройки, функционального тестирования и других подобных операций по отношению к абоненту;

- 6 - 8 - резерв;

- 9 - сообщение адресуется программным средствам вычислительных сетей;

- 10 - 15 - сообщение адресуется программным средствам сбора и обработки информации и управления.

Аналогичные смысл и возможные значения имеет и код уровня источника, но в отличие от КУП он определяет объект, который является источником передаваемого сообщения.

Код вида сообщения (КВС) определяет, к какому из видов относится данное сообщение. Его возможные значения приведены в табл. Б.2.

Формат информационного поля длинных сообщений определяется отдельными соглашениями для каждого типа абонента в процессе его проектирования. Код модификации (КМ) для инициативных сообщений содержит так называемый счетчик инициативных сообщений (СИС). СИС в состав идентификаторов этих сообщений введен для обеспечения контроля их приема. А именно, при передаче инициативных сообщений для каждого значения КУП должна вестись их нумерация по модулю 16, что обеспечивает на приемной стороне возможность определения потери сообщений. Для других видов сообщений КМ может содержать дополнительные данные, указывающие на различные особенности обработки сообщений.

Таблица Б.2

Значение поля КВС	
Значение КВС	Вид передаваемого сообщения
0000	Исходное сообщение, содержащее последнюю порцию данных
0001	Ответное сообщение, содержащее последнюю порцию данных
0010	Инициативное сообщение, содержащее единственную порцию данных; здесь и в дальнейшем под инициативным сообщением понимается длинное сообщение, предполагающее передачу подтверждения о его приеме (но не предполагающее передачу ответного длинного сообщения) и содержащее, как правило, информацию о различного рода изменениях в управляемом объекте, аварийных ситуациях и т.п.
0100	Исходное сообщение, содержащее не последнюю порцию данных
0101	Ответное сообщение, содержащее не последнюю порцию данных
1000	Исходное сообщение, содержащее данные, которые передаются в режиме ЭХО (в ответ на это сообщение от абонента-адресата должно поступить сообщение, содержащее полученные данные)
1001	Ответное сообщение, содержащее данные, переданные в режиме ЭХО
1010	Исходное сообщение запроса состояния (в ответ на это сообщение от абонента-адресата должно быть передано сообщение, содержащее данные о состоянии объекта абонента, определенных значениями КУП и АВУП)
1011	Сообщение, содержащее данные о состоянии объекта абонента
1100	Сообщение, содержащее время суток и дату
1101	Сообщение, содержащее информацию об изменении состояния абонента

Код модификации (КМ) для сообщений с КВС=0000, 0001, 0100, 0101 определяет номер порции передаваемых данных (0-15). Таким образом, максимальная длина передаваемого сообщения составляет $16 \times 256 = 4096$ байт. Номер порции данных, указываемый в байте 1 идентификатора, используется для передачи массивов данных, включающих в себя более 256 байтов. Такие массивы передаются в исходных или ответных сообщениях в виде последовательностей сообщений, каждое из которых содержит очередную порцию данных. Порции нумеруются последовательными числами, начиная с нуля; при передаче последней порции КВС должен быть равен 0000 или 0001, а не последней - 0100 или 0101. Следует заметить, что пассивные

абоненты не могут передавать последовательности, включающие в себя более одной порции. Они не могут также отвечать на передачу длинного сообщения коротким сообщением типа ЗН.

Не могут передаваться последовательности, содержащие более одного сообщения, и связному оборудованию или от связного оборудования любых абонентов сети МАПС, т.е. сообщения с КУП=0 или КУИ=0 всегда содержат только одну порцию данных. Ответ на сообщение, адресованное связному оборудованию абонента (КУП=0), должен передаваться в том же сеансе связи, т.е. не допускается выдача в ответ на данное сообщение сообщений типа ЗН или ПТЖА.

Код модификации (КМ) для сообщений с КВС=1010, 1011, 1101 определяет характер запрашиваемой/передаваемой информации. Адреса второго уровня, содержащиеся в байтах 2 и 3 идентификатора, используются для указания абонентов второго уровня, подключенных к абоненту данной сети любым допустимым для данного абонента способом (подключение сетей второго уровня, радиальное подключение абонентов второго уровня по стыку С2 или др.). Все сообщения для/от абонентов второго уровня требуют наличия идентификатора. Значение 00000000 в этом поле означает, что сообщение относится к аппаратуре или программным средствам собственно абонента данной сети МАПС.

Б.2 Взаимодействие между абонентами магистрали МАПС

Б.2.1 Типы и форматы сообщений

Взаимодействие между абонентами магистрали сети МАПС осуществляется путем передачи в соответствии с определенными регламентами нескольких взаимосвязанных сообщений, которые в сочетании с операциями их обработки в дальнейшем именуются процедурами. Предоставление магистрали абоненту осуществляется путем передачи ему короткого сообщения ПМ. Это сообщение может передаваться только активным арбитром.

Возврат магистрали (отказ от использования магистрали) осуществляется абонентом путем передачи активному арбитру сообщения типа ВМ или ВМД. Адрес, по которому передается это сообщение, - адрес активного арбитра - равен обратному адресу в сообщении ПМ, по которому предоставлена магистраль. Сообщение ВМ передается, во-первых, инициативными абонентами в любом случае, во-вторых, активизируемыми абонентами в случае, когда состояние активности не должно измениться. Если же абонент должен перейти в пассивное состояние, передается сообщение ВМД. Получив магистраль, абонент использует ее для передачи различных сообщений другим абонентам в соответствии с регламентами, описанными ниже в данном разделе. При этом возможны следующие разновидности сообщений: сообщения, не требующие подтверждений о приеме и ответов от абонентов-адресатов; требующие только подтверждения о приеме от абонентов-адресатов; требующие ответных сообщений от абонентов-адресатов; ответные сообщения.

Сообщения ЗН передаются в том же сеансе связи в ответ на сообщения, требующие ответов или подтверждений. Эти сообщения свидетельствуют о невозможности приема исходных сообщений, например, из-за отсутствия области памяти нужного размера для буферирования сообщений. Когда же возникнет возможность приема сообщений, абонент в своем сеансе передает сообщение ЗП. На сообщение ЗП абонент должен ответить подтверждением (ПТЖ или ПТЖА) в том же сеансе связи, а выдачу не принятых ранее сообщений повторить в своем сеансе связи. Сообщение ПТЖА выдается в том случае, когда необходимость передачи не принятых ранее сообщений еще не отпала, и ПТЖ в противном случае. Сообщения ПТЖ, ПТЖА, ОШН и ОШ - это подтверждения о приеме сообщений. Они выдаются в том же сеансе связи в ответ на сообщения, требующие подтверждения или ответа. Сообщения ПТЖ, ОШН и ОШ не изменяют состояния активности абонента, а сообщение типа ПТЖА указывает активному арбитру, что абонент должен быть переведен в активное состояние. Сообщения типа ПТЖ и ПТЖА свидетельствуют о начале обработки исходного сообщения, а сообщения ОШН и ОШ - об ошибке. В последнем случае исходное сообщение не обрабатывается. Сообщения ЗП предназначены для запроса исходного сообщения в случае, когда в ответ на

первоначальную его передачу от абонента-адресата поступило сообщение ЗН, свидетельствующее о занятости. Сообщения ЗС и ОЗС предназначены для запроса состояния абонентов магистральной (например, в процессе опроса арбитром состояния абонентов, подключенных к магистральной).

Б.2.2 Порядок обмена сообщениями на магистральной между абонентами магистральной

Основной процедурой является так называемая процедура опроса, выполняемая активным арбитром. Суть этой процедуры заключается в предоставлении магистральной абонентам в соответствии с описанным ниже правилом. А именно, в процессе выполнения процедуры опроса магистраль предоставляется последовательно всем работоспособным приоритетным инициативным абонентам и всем приоритетным активизируемым абонентам, находящимся в активном состоянии. Если суммарное время предоставления магистральной этим абонентам не превысило заданное для МАПС время цикла T_0 , то в течение оставшегося промежутка выполняется предоставление магистральной всем или части работоспособных неприоритетных инициативных абонентов и неприоритетных активизируемых абонентов, находящихся в активном состоянии. При этом в каждом цикле первому предоставляется магистраль абоненту, номер которого в списке на единицу больше номера последнего абонента, которому предоставлена магистраль в предыдущем цикле.

При предоставлении магистральной абоненту в процессе выполнения процедуры опроса активный арбитр посылает короткое сообщение типа ПМ. При этом возможны следующие ответные реакции:

- от абонента активному арбитру передается короткое сообщение типа ВМ, означающее отказ от предоставления магистральной, арбитр переходит к предоставлению магистральной следующему абоненту в соответствии с описанным выше порядком;

- в течение промежутка времени $3 \cdot t$ с момента предоставления магистральной абонент не передает по ней никаких данных, арбитр повторяет попытку предоставления магистральной вновь: если эта попытка не увенчалась успехом, отмечается неработоспособность абонента и осуществляется переход к предоставлению магистральной следующему абоненту;

- абонентом выполняются процедуры обмена данными. Активный арбитр следит за ходом их выполнения, фиксируя следующие моменты:

- в ответ на длинное сообщение от абонента-адресата поступило сообщение ПТЖА. Арбитр переводит в активное состояние абонента, от которого поступило данное сообщение, если он не был переведен в это состояние ранее. При этом приоритетный абонент включается в список приоритетных абонентов, а неприоритетный - неприоритетных. Такие же действия выполняются по передаче короткого сообщения ЗН;

- от абонента активному арбитру передается короткое сообщение ВМ, означающее отказ от предоставления магистральной. Арбитр переходит к предоставлению магистральной следующему абоненту в соответствии с описанным выше порядком;

- от абонента активному арбитру передается короткое сообщение ВМД, означающее отказ от предоставления магистральной и отмену состояния активности. Арбитр исключает из списка инициативных абонента и переходит к предоставлению магистральной следующему абоненту в соответствии с описанным выше порядком;

- в течение промежутка времени $3 \cdot t$ с момента последнего обмена по магистральной не передается по ней никаких данных. Арбитр осуществляет переход к предоставлению магистральной следующему абоненту;

- абонент превысил отведенную квоту владения магистралью (квота владения магистралью абонентом описана ниже). Арбитр фиксирует неработоспособность магистральной, сообщая об этом в ВК, к которому он подключен (если такой ВК имеется).

Возобновление работы магистральной осуществляется после выполнения ремонтных работ и инициализации функционирования КСв и абонентов. Далее рассматриваются действия

абонента при предоставлении ему магистрали. Абонент может либо не иметь никаких запросов на магистраль (в этом случае он передает активному арбитру короткое сообщение ВМ), либо иметь один или несколько запросов на выдачу сообщений. Все эти запросы делятся на две группы - приоритетные и неприоритетные. С другой стороны, по происхождению запросы делятся на исходные и ответные. И те, и другие могут быть приоритетными и неприоритетными (приоритетность ответных сообщений определяется равной приоритетности исходного сообщения). Приоритетные ответные сообщения должны выдаваться раньше приоритетных исходных, а неприоритетные ответные - раньше неприоритетных исходных. Для приоритетных абонентов, если промежуток времени, прошедший после последнего отказа данным абонентом от магистрали, меньше уставки нормальной длительности цикла T_0 т.е. магистраль загружена слабо), абонент начинает удовлетворять свои запросы в порядке убывания приоритета до исчерпания заявок либо до исчерпания длительности цикла. При этом количество длинных (исходных и ответных) сообщений не должно превысить величины N_0 . После этого могут удовлетворяться только приоритетные запросы, причем количество длинных (исходных и ответных) сообщений не должно превысить числа N_1 . Если магистраль абоненту была предоставлена тогда, когда нормальная длительность цикла T_0 уже была исчерпана, он может передавать приоритетные сообщения в количестве не более N_1 , а при отсутствии таковых имеет право передать одно неприоритетное сообщение. Для неприоритетных абонентов в одном сеансе предоставления магистрали могут передаваться длинные сообщения в количестве не более N_2 .

Передача активным арбитром сообщений, полученных им из ВК, к которому он подключен, выполняется также в соответствии с описанным выше порядком.

Величины T_0 , N_0 , N_1 , N_2 и t являются настроечными параметрами работы магистрали. Они формируются в памяти КСв в процессе инициализации работы, могут быть изменены в процессе функционирования и передаются абонентам в коротком сообщении ЗС, формат которого описан в п.Б.1.3. По окончании использования магистрали абонент выдает короткое сообщение типа ВМ или ВМД.

Для нормального функционирования сети МАПС арбитр и абоненты должны придерживаться следующих правил:

1. Передача абоненту-адресату очередного длинного сообщения любого вида (исходного, инициативного, циркулярного, ответного), предоставление абоненту магистрали (ПМ), запрос данных от абонента (ЗП), запрос состояния абонента (ЗС), если предыдущее сообщение такого же класса выдавалось данному абоненту, осуществляются не ранее чем через 1 ms (но не позже чем через время t) после завершения передачи или приема очередного сообщения любого вида;

2. Передача абоненту-адресату очередного длинного сообщения любого вида (исходного, инициативного, циркулярного, ответного), предоставление абоненту магистрали (ПМ), запрос данных от абонента (ЗП), запрос состояния абонента (ЗС), если предыдущее сообщение такого же класса выдавалось не данному абоненту, осуществляются не ранее чем через $200\text{ }\mu\text{s}$ (но не позже чем через время t) после завершения передачи или приема очередного сообщения любого вида;

3. Сообщения ВМ, ВМД, ЗН, ПТЖ, ПТЖА, ОШ, ОШН, ОЗС передаются абонентом-источником не ранее, чем через $200\text{ }\mu\text{s}$ (но не позже чем через время t) после получения исходного сообщения;

4. Величина цикла T_0 определяется исходя из количества и типов абонентов на магистрали.

Примечание: Если абонент, выдав сообщение, должен согласно протоколу повторить его в этом же сеансе предоставления магистрали, а временная или количественная квоты исчерпались, абонент выполняет повторение выдачи и только после этого отказывается от магистрали. Аналогично в случае исчерпания временной квоты после выдачи длинного сообщения отказ от магистрали выполняется только после передачи всех необходимых по регламенту коротких сообщений и циркулярных сообщений о выявленной неработоспособности других абонентов и изменении состояния данного абонента.

В следующих пунктах данного раздела описывается порядок выполнения различных процедур абонентами после предоставления им магистральной.

Б.2.3 Процедура "Выдача без подтверждения"

Абонент посылает длинное сообщение (или последовательность длинных сообщений, передача которых описана ниже), в управляющем байте которого ТС равен 00 или 10 в зависимости от того, данное сообщение исходное или ответное, и переходит к следующей процедуре или посылает сообщение типа ВМ либо ВМД. Адресат принимает сообщение. В случае искажения данных, обнаруженного аппаратом контрольного суммирования, сообщение игнорируется. Сообщение игнорируется также в случае, когда оно не может быть немедленно обработано и отсутствует возможность его сохранения для последующей обработки. Кроме этого, сообщение игнорируется в случае, когда оно содержит логическую ошибку или неработоспособен адресат, которому оно предназначено (шлюз абонента, оборудование для заданного КУП, абонент второго уровня). Данная процедура используется для вывода различного рода информационных сообщений, в том числе и циркулярных.

Если массив данных, который необходимо передать, по длине превышает 256 байтов, то, как было упомянуто выше, он передается в нескольких последовательных сообщениях. Следует заметить, что нельзя "разрывать" передачу такой последовательности сообщений, т.е. передавать тому же самому абоненту-адресату какие-либо другие длинные сообщения между передачей сообщений с порциями данных массива, кроме циркулярных сообщений и сообщений, адресованных связному оборудованию (КУП=0).

Примечания: 1. В процессе выполнения процедуры "Выдача без подтверждения" при передаче циркулярных сообщений массив данных должен включать в себя не более 256 байтов и должен передаваться в одном сообщении.

2. Поскольку в процессе выполнения процедуры "Выдача без подтверждения" потеря сообщений не фиксируется передающей стороной, то эта процедура может применяться для передачи "не важных" данных или в случае, когда имеется возможность на потребительском уровне обнаружить отсутствие адекватной реакции на прием сообщения адресатом.

Б.2.4 Процедура "Выдача с подтверждением"

Если в процессе выполнения процедуры необходимо передать абоненту-адресату массив данных, длина которого не превышает 256 байтов, последовательность действий должна быть следующей. Абонент посылает длинное сообщение, в управляющем байте которого ТС равен 01, и переходит к ожиданию ответного сообщения. При этом возможны следующие ситуации:

- в ответ от адресата поступило короткое сообщение ПТЖ. Абонент переходит к следующей процедуре или отказывается от использования магистральной;

- в ответ от адресата поступило короткое сообщение типа ОШ или ОШН. Абонент фиксирует ошибочную ситуацию и переходит к следующей процедуре или отказывается от использования магистральной;

- в ответ от адресата поступило короткое сообщение ЗН. Абонент переходит к следующей процедуре или отказывается от использования магистральной, но исходная процедура считается не выполненной, продолжение ее выполнения (начиная с выдачи исходного сообщения) осуществляется после поступления от адресата сообщения типа ЗП. Время до поступления этого сообщения контролируется по тайм-ауту. Сообщение ЗН не может быть выдано пассивным абонентом;

- в течение промежутка t от адресата не поступило никаких сообщений. Абонент повторяет вывод исходного сообщения (указывая 1 в разряде ПП управляющего слова) и, если вновь повторилась та же ситуация, фиксируется неработоспособность адресата.

Адресат, получив сообщение (без искажений в данных, обнаруженных аппаратом контрольного суммирования), анализирует его допустимость и возможность его обработки. Ес-

ли запрашиваемая операция не может быть выполнена из-за неработоспособности оборудования (неработоспособность оборудования для соответствующего КУП; неработоспособность абонента второго уровня; неработоспособность шлюза абонента, которому адресовано сообщение), то адресат, получив сообщение, отвечает коротким сообщением ОШН. Принятые данные при этом игнорируются.

Если запрашиваемая операция не может быть выполнена из-за логической ошибки в сообщении (в абоненте отсутствует обработка данного вида сообщения), то адресат отвечает коротким сообщением ОШ. Принятые данные при этом игнорируются. Если сообщение допустимо, но отсутствует возможность начать его обработку немедленно или сохранить его для последующей обработки адресатом (инициативным, активизируемым), выдается сообщение ЗН. Когда же появится возможность обработки операции, адресат должен послать абоненту-источнику короткое сообщение ЗП (в свой сеанс предоставления магистральной связи) и в ответ (в том же сеансе) получить сообщение ПТЖА или ПТЖ. Абонент, инициировавший процедуру, в свой сеанс предоставления магистральной связи продолжит выполнение процедуры, начиная с передачи исходного сообщения. Если запрашиваемая операция может быть выполнена, то адресат, получив сообщение, отвечает коротким сообщением ПТЖ или ПТЖА и приступает к выполнению обработки. Если в прибывшем сообщении искажены данные и это искажение обнаружено аппаратом контрольного суммирования, то адресат игнорирует принятые данные и в ответ не передает никаких сообщений.

Примечания: 1 - после получения от адресата сообщения ЗН абонент должен придержать выдачу данному адресату какие-либо сообщения, кроме "Выдачи без подтверждения", до получения сообщения ЗП; 2 - отслеживая передачу сообщений, активный арбитр в случае выдачи сообщения ЗН активизируемым абонентом должен переводить последний в активное состояние; 3 - пассивный абонент не имеет права в ответ выдавать сообщение ЗН.

Если массив данных, который необходимо передать в начале выполнения процедуры, по длине превышает 256 байтов, то он передается в нескольких последовательных сообщениях, в которых ТС должно быть равно 01. В ответ на каждое сообщение абонент-адресат должен выдать сообщение типа ЗН, ПТЖ, ПТЖА, ОШН или ОШ. Каждое из этих сообщений выдается и обрабатывается в соответствии с описанным выше алгоритмом с тем лишь исключением, что в случае выдачи короткого сообщения ЗН продолжение выполнения процедуры осуществляется не с начала, а с передачи сообщения, в ответ на которое поступило сообщение ЗН.

Абоненту-источнику нельзя "разрывать" передачу последовательности сообщений, т.е. нельзя передавать тому же самому абоненту-адресату какие-либо другие длинные сообщения между передачей сообщений с порциями данных массива, кроме исходных сообщений, не предполагающих ответов и подтверждений (например, циркулярных), и сообщений, адресованных связанному оборудованию (КУП=0).

Б.2.5 Процедура "Выдача с подтверждением или ответом"

Если в процессе выполнения процедуры необходимо передать абоненту-адресату массив данных, длина которого не превышает 256 байтов, последовательность действий должна быть следующей. Абонент посылает длинное сообщение, в управляющем байте которого ТС равен 11, и переходит к ожиданию ответного сообщения. При этом возможны следующие ситуации:

- в ответ от адресата поступило короткое сообщение ПТЖА, означающее, что ответ от адресата, содержащий результат запрашиваемой операции, поступит позже в отдельном сеансе предоставления магистральной связи, абонент переходит к следующей процедуре или отказывается от использования магистральной связи. Сообщение ПТЖА не может быть выдано пассивным абонентом;

- в ответ от адресата поступило короткое сообщение ОШ, означающее недопустимость исходного сообщения. Абонент фиксирует ошибочную ситуацию и переходит к следующей процедуре или отказывается от использования магистральной связи;

- в ответ от адресата поступило короткое сообщение ОШН, означающее неработоспособность оборудования адресата, которому предназначено сообщение. Абонент фиксирует ошибочную ситуацию и переходит к следующей процедуре или отказывается от магистрали;

- в ответ от адресата поступило короткое сообщение ЗН. Абонент переходит к следующей процедуре или отказывается от использования магистрали, но исходная процедура считается не выполненной, продолжение ее выполнения (начиная с выдачи исходного сообщения) осуществляется после поступления от адресата сообщения ЗП. Время до поступления этого сообщения контролируется по тайм-ауту. Сообщение ЗН не может быть выдано пассивным абонентом;

- в ответ от адресата поступило ответное длинное сообщение. Это сообщение содержит результат выполненной операции. Абонент переходит к следующей процедуре. Пассивный абонент всегда должен передавать запрашиваемые у него данные в этом же сеансе связи;

- в ответ от адресата не поступило никаких сообщений в течение промежутка времени t. Осуществляется попытка повторить процедуру (аналогично описанной в предыдущем пункте) и, если вновь выявлена ошибка, фиксируется неработоспособность адресата.

Адресат, получив сообщение (без искажений в данных, обнаруженных аппаратом контрольного суммирования), анализирует его допустимость и возможность его обработки. Если сообщение недопустимо, в ответ адресатом выдается короткое сообщение ОШ. Принятые данные при этом игнорируются. Если же оно допустимо, но неработоспособно оборудование, которому предназначено сообщение (неработоспособен шлюз, адресат второго уровня, оборудование для заданного КУП), выдается короткое сообщение ОШН. Принятые данные при этом игнорируются. Если же оно допустимо, но отсутствует возможность начать его обработку немедленно или запомнить для последующей обработки, выдается сообщение ЗН. Когда же появится возможность обработки операции, адресат должен послать абоненту короткое сообщение ЗП (в свой сеанс предоставления магистрали) и в ответ (в том же сеансе) получить сообщение ПТЖА, если необходимость передачи исходного сообщения еще не отпала, и сообщение ПТЖ в противном случае. Если подтверждение на сообщение ЗП не получено, то выдается повторное сообщение ЗП. Если и на него не получено подтверждение, то фиксируется неработоспособность абонента. Абонент, инициировавший процедуру, в свой сеанс предоставления магистрали повторит передачу исходного сообщения, в ответ на которое было передано сообщение ЗН. Если в прибывшем сообщении искажены данные и это искажение обнаружено аппаратом контрольного суммирования, то адресат игнорирует принятые данные и в ответ не передает никаких сообщений. Если запрашиваемая операция может быть выполнена немедленно и длина передаваемого в ответ массива данных, содержащего результат выполненной операции, не превышает 256 байтов, то адресат, получив сообщение, в этом же сеансе связи посылает ответное сообщение. В этом ответном сообщении ТС должен быть равным 10. Пассивный абонент имеет право передавать только такого вида ответные сообщения. Если запрашиваемая операция не может быть выполнена немедленно или длина передаваемого в ответ массива данных, содержащего результат выполненной операции, превышает 256 байтов, то адресат, получив сообщение, в этом же сеансе связи посылает ответное короткое сообщение ПТЖА. Затем в свои сеансы связи он посылает ответные сообщения (одно или последовательность из нескольких сообщений). Во всех сообщениях ответной последовательности ТС должен быть равным 01.

В случае, когда в начале выполнения процедуры (в исходном сообщении) абоненту-адресату следует передавать массив, длина которого превышает 256 байтов, вместо одного исходного сообщения передается последовательность сообщений так, как это описано в предыдущем пункте. В этой последовательности все сообщения, кроме последнего, должны содержать в управляющем байте ТС, равный 01, а в последнем сообщении ТС должен быть равным 11.

НЕКОТОРЫЕ АСПЕКТЫ ПЕРВИЧНОЙ ОБРАБОТКИ ИЗМЕРИТЕЛЬНОЙ ИНФОРМАЦИИ

В данном приложении приведены методы фильтрации измеренных значений сигналов, методы линеаризации значений сигналов термопар и термопреобразователей сопротивления, компенсации влияния термоэлектродвижущей силы свободных концов термопар, а также методы масштабирования измерений.

В.1 Фильтрация измеренных значений сигналов

В.1.1 Назначение и основные свойства цифровых фильтров

Каждое измерение параметра технологического процесса в цифровой измерительной системе, для получения которого используются, по крайней мере, коммутатор и аналого-цифровой преобразователь, сопряжено с влиянием помех. Источником помех в ряде случаев могут быть также датчики сигналов, объект (например, его ионизирующее излучение), характер материального потока (например, турбулентность среды) и др.

Помехам присущ более высокий частотный спектр, чем спектр измеряемого параметра (не менее чем на порядок). Вид сигнала на входе в фильтр, до устранения помехи, и сигнала на выходе фильтра показан на рис. В.1.



Рис. В.1. Вид сигналов на входе и выходе фильтров.

Под фильтрацией понимается процесс изменения частотного спектра сигнала с целью усиления или выделения одних частот и ослабления или подавления других. По принципам реализации различают фильтры аналоговые (непрерывные) и цифровые. Аналоговые фильтры преобразуют непрерывный во времени t аналоговый сигнал $x(t)$ на входе в аналоговый сигнал $y(t)$ на выходе. Цифровые фильтры обрабатывают равномерно распределенные, квантованные во времени, последовательности входных чисел $x(i \cdot \Delta T)$, где ΔT - постоянная величина (шаг квантования), а i - целое число. В дискретные моменты времени $t = iT$ дискретный сигнал $x(i\Delta T)$ может достаточно точно, с погрешностью преобразования, воспроизводить свой аналоговый образ $x(t)$. Результатом обработки входных в фильтр данных $x(i\Delta T)$ является последовательность выходных данных $y = (i\Delta T)$ - реакция фильтра на входное воздействие.

Свойства аналоговых фильтров, за исключением некоторых специальных, описываются во временной области обыкновенными дифференциальными уравнениями, а в частотной - частотными характеристиками. Для описания цифровых фильтров применяются разностные уравнения.

В последнее время существенное развитие получили теория, методы проектирования и программное обеспечение цифровой фильтрации сигналов. В практике обработки данных широко применяются фильтры Баттерворта, Чебышева, Бесселя, Калмана и др. [77, 78]. Основными характеристиками фильтров являются: полоса пропускания — диапазон частот (ω_1, ω_2) в котором выходные сигналы, по крайней мере, не ослаблены (усилены или слегка ослаблены); полоса задержания - диапазон частот (ω_1, ω_2) , в котором выходные сигналы имеют нулевые или близкие к нулевым, значения; переходная полоса - диапазон частот между полосой пропускания и полосой задержания.

Для применения в информационно-измерительных и других автоматизированных системах управления технологическими процессами, как правило, интерес представляют фильтры с заданной полосой пропускания нижних частот, узкой переходной полосой и заданной полосой задержания.

В.1.2 Цифровая фильтрация измеренных значений сигналов в ПТК

Свойства одного из часто применяемых в ПТК фильтров, простого по структуре, с ограниченным перечнем параметров настройки и минимальными требованиями к ресурсам времени и памяти процессора, описываются во временной области обыкновенным дифференциальным уравнением первого порядка вида:

$$T * \frac{dy(t)}{dt} + y(t) = x(t),$$

где $x(t)$ - сигнал на входе фильтра; $y(t)$ - сигнал на выходе фильтра; T - постоянная времени фильтра.

Структурная схема фильтра показана на рис. В.2.

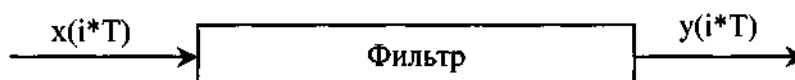


Рис. В.2. Структурная схема фильтра.

Свойства фильтров устанавливаются посредством анализа их уравнений в операторной форме.

$$(Tp + 1)y(p) = x(p).$$

Передаточная характеристика фильтра (отношение изображения сигнала, на выходе фильтра к его изображению на его входе) имеет вид:

$$W(p) = \frac{1}{Tp + 1}$$

Из уравнения передаточной характеристики находится зависимость изменения амплитуды выходного сигнала на выбранной частоте ω и зависимость запаздывания сигнала на выходе фильтра по отношению к сигналу на входе, т.е. амплитудная и фазовая характеристики фильтра. Модуль фильтра $A(\omega)$ первого порядка и сдвиг фаз $\varphi(\omega)$ описываются выражениями:

$$A(\omega) = \frac{1}{\sqrt{T^2 * \omega^2 + 1}}, \quad \varphi(\omega) = -\arctg(T * \omega).$$

Модуль $A(\omega)$ амплитудно-частотной характеристики показывает, во сколько раз увеличивается (или уменьшается) амплитуда выходного сигнала по сравнению с амплитудой входного сигнала при заданной частоте ω гармонического воздействия. Аргумент $\varphi(\omega)$ - угол сдвига выходного сигнала по отношению к входному. На рис. В.3 показано, как выглядят характеристики $A(\omega)$ и $\varphi(\omega)$ зависимости от изменения частот ω , при их наличии в спектре входного сигнала.

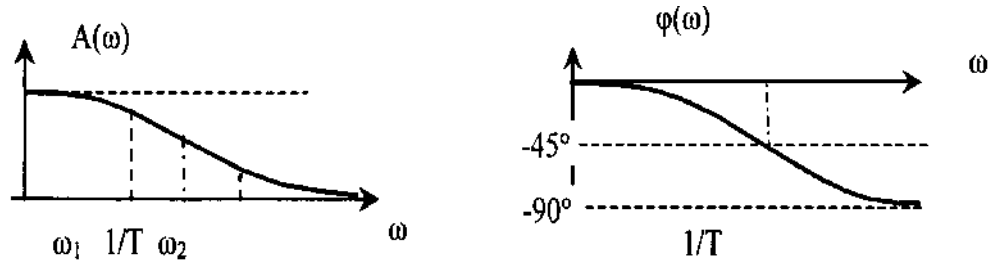


Рис. В.3. Зависимости $A(\omega)$ и $\varphi(\omega)$ от ω .

С увеличением частоты ω амплитуда $A(\omega)$ убывает и тем быстрее, чем больше T . Постоянная T характеризует инерционность фильтра. Амплитудная характеристика имеет перегиб в точке $\omega = 1/T$. Считается, что фильтр первого порядка пропускает все частоты ω от 0 до $1/T$ и подавляет частоты выше $\omega = 1/T$. Частота $\omega = \omega_c = 1/T$ называется частотой среза фильтра. Другими словами, если бы фильтруемый сигнал $x(t)$ представить суммой гармонических сигналов (рядом Фурье) и к такому сигналу применить фильтр, то на его выходе частоты выше $\omega_c = 1/T$ практически отсутствовали бы. И если бы из каких-либо соображений удалось задать значение $A(\omega)$ для частоты ω - из уравнения амплитудной характеристики можно было бы получить значение постоянной времени T .

Рассмотрим цифровой эквивалент аналогового фильтра первого порядка. В разностной форме приведенное выше дифференциальное уравнение имеет вид:

$$T * \frac{y_i - y_{i-1}}{\Delta T} + y_i = x_i.$$

Индексом i в уравнении обозначен момент времени $t = i\Delta T$, где ΔT - шаг квантования входного сигнала по времени, величина постоянная; y_i - значение сигнала (в цифровом виде) на выходе фильтра, если на вход фильтра в этот же момент времени подан сигнал x_i . Решением уравнения относительно выхода фильтра (относительно y_i) будет:

$$y_i = \frac{T}{T + \Delta T} y_{i-1} + \frac{\Delta T}{T + \Delta T} x_i = k y_{i-1} + (1 - k) x_i.$$

Очевидными свойствами коэффициентов k и $(1 - k)$ уравнения цифрового фильтра являются:

$$\frac{T}{T + \Delta T} + \frac{\Delta T}{T + \Delta T} = k + (1 - k) = 1; \quad 0 \leq k \leq 1.$$

При $k = 0$ фильтр отключен - пропускает все частоты. При $k = 1$ все частоты подавляются. Важной особенностью цифрового фильтра является то, что его выход зависит не только от постоянной времени T (как это имеет место у аналогового фильтра), но и от шага ΔT квантования входного сигнала по времени. Так как постоянная времени T фильтра определяется частотой среза ω_c (и, наоборот), для k можно принять

$$k = \frac{T}{T + \Delta T} = \frac{1}{1 + \omega_c \Delta T}.$$

Для получения оценки частоты среза ω_c должен быть исследован спектральный состав сигнала $x(t)$, выявлены частоты, свойственные собственно технологическому процессу, и частоты помехи, источником которой технологический процесс не является.

Важным с точки зрения реализации цифрового фильтра является поддержание постоянства шага ΔT квантования входного сигнала во времени. Непостоянство ΔT , его случайные значения, могут стать источником существенных дополнительных погрешностей последовательности данных $y(i \cdot \Delta T)$ на выходе фильтра.

В.1.3 Фильтрация входных сигналов методом скользящего усреднения

В предыдущем пункте рассмотрен применяемый в ПТК для обработки входных сигналов один из классических цифровых фильтров - линейный фильтр первого порядка. Другим, часто применяемым методом фильтрации измеренных значений параметров, является усреднение измерений на скользящем интервале. В основе метода лежит замена измеренного значения параметра (сигнала) средним значением параметра, взятым по заданному (постоянному) множеству измерений. Для получения скользящего среднего значения $y(i \cdot \Delta T)$ на интервале $m \cdot \Delta T$ на каждом шаге i измерений вычисляется скользящая сумма

$$S(i\Delta T) = \sum_{m=0}^{k-1} x((i-m)\Delta T).$$

Более эффективной с вычислительной точки зрения является схема счета, в которой $S(i\Delta T) = S((i-1)\Delta T) + x(i\Delta T) - X((i-m)\Delta T)$. Среднее значение $y(i \cdot \Delta T)$ измеряемой величины в момент времени $t = i \cdot \Delta T$ равно

$$y(i\Delta T) = S(i\Delta T) / m.$$

С вычислительной точки зрения алгоритм эффективен при $m = 2^n$, где n - целое.

В2 Линеаризация измеренных значений сигналов термопар и термопреобразователей сопротивления.

Компенсация влияния термоэлектродвижущей силы свободных концов термопар

В.2.1 Измерения температуры и вычислительные проблемы

Для измерения температуры в промышленных условиях применяются термоэлектрические преобразователи (термопары) и термопреобразователи сопротивления. Термоэлектроды, применяемые для изготовления термопар, и терморезисторы термометров сопротивления генерируют выходной сигнал (электродвижущую силу или сопротивление), связанные, как правило, нелинейной зависимостью с измеряемым параметром — температурой T . Соответствие (номинальные статические характеристики преобразования — НСХ) между температурой и выходными сигналами датчиков - термоэлектродвижущей силой E и, для термометров сопротивления, сопротивлением R) установлено межгосударственными стандартами.

НСХ как термопар, так и термометров сопротивления имеют существенные нелинейности. Для обеспечения точности аппроксимация НСХ выполнена полиномами вида:

$$t = \sum_{k=0}^N d_k * E^k,$$

где: N изменяется в пределах от 8 до 12 и зависит от типа преобразователя, d_k - коэффициенты полинома.

Для получения по измеренному значению E (или R) температуры t расчет должен осуществляться с применением вещественной арифметики двойной точности и, очевидно, требует значительных ресурсов процессорного времени. Если принять во внимание тот факт, что в информационно-измерительных и управляющих системах основным видом измерений являются температурные (до 70 % от общего числа параметров), а количество контролируемых параметров температуры достигает нескольких сотен и что измерения параметров должны выполняться с периодом от 100 ms до 1 s - поиск и применение в АСУ ТП эффективных методов расчета температуры являются одними из важнейших задач математического обеспечения.

Одним из широко применяемых на практике методов расчета температуры является понижение порядка аппроксимирующего НСХ полинома до второго или третьего, так что

$$t = \sum_{k=0}^2 c_k E^k,$$

где c_k - вещественные коэффициенты аппроксимирующего полинома второй степени.

Возможность понижения порядка достигается при этом за счет сужения области определения полинома посредством ограничения ее (области) пределами измерения температуры в конкретных условиях. Так, например, при измерении температуры в компенсационной коробке, установленной в отапливаемом помещении, достаточно измерять температуру в диапазоне от 0 до 100 °С и, следовательно, НСХ датчика температуры должна быть аппроксимирована только в данном диапазоне.

Несмотря на возможность минимизации требований к ресурсам процессорного времени, предлагаемый путь не лишен существенного недостатка, а именно: аппроксимирующие полиномы должны быть созданы персонально для каждого канала измерения температуры. По этой же причине программно-технический комплекс теряет свойство автоматической адаптации при его переносе на другой объект, возрастают трудозатраты разработки математического, программного и других видов обеспечения.

В.2.2 Линеаризация номинальных статических характеристик термопреобразователей и расчет температур

По сравнению с описанным выше подходом со свойственными ему недостатками, метод расчета температур продвинул в следующих основных двух направлениях. Во-первых, в качестве интервала аппроксимации НСХ датчика принят не рабочий диапазон изменения температур конкретного объектового параметра, а диапазон изменения сигналов низкого уровня на входе измерительных каналов. Во-вторых, в диапазоне приема каналом входного сигнала аппроксимация НСХ выполнена не квадратичными полиномами, а кусочно-линейными функциями. Погрешность аппроксимации двусторонняя. На концах диапазона (в начале и конце шкалы) погрешность равна нулю - аппроксимируемая и аппроксимирующая функции совпадают. Наибольшая погрешность достигается, как правило, в узловых точках. Величина предела допускаемой погрешности аппроксимации является параметром настройки метода аппроксимации и является функцией предела инструментальной погрешности измерительного канала (принята меньшей ее в два - три раза). Для прецизионных измерительных каналов с пределом допускаемой приведенной погрешности ± 0.04 % погрешность принятого метода аппроксимации не превышает ± 0.04 °С.

Поддержание двухсторонней погрешности позволило сократить количество узлов, ориентировочно, в два раза. В среднем, количество узлов колеблется от двух до десяти и зависит как от предъявленных требований к точности аппроксимации и характера нелинейности НСХ, так и от диапазона принимаемых измерительным каналом входных сигналов. В промежутке между соседними узлами (E_i, T_i) и (E_{i+1}, T_{i+1}) температура рассчитывается по линейному уравнению связи (рис. В.4), так что

$$T = \frac{T_{i+1} - T_i}{E_{i+1} - E_i} * (E - E_i) + T_i$$

Для упрощения вычислений процедуре расчета температуры предоставляются не значения температуры T_i , напряжений E и E_i (сопротивления R и R_i) в узлах i , а их относительные значения (коды)

$$N_i = \text{Ent}[(E_i + 0,5h) / h], \quad i = 1, 2, \dots,$$

где N_i - цифровое значение (код аналого-цифрового преобразователя); $\text{Ent}[\]$ - символ, определяющий целую часть числа; E_i - значение напряжения (или сопротивления в узле); h - номинальная ступень квантования, в единицах преобразуемой величины.

Расчет значений температуры осуществляется в ПТК, таким образом, посредством применения целочисленной арифметики 16-разрядного формата. Линеаризация НСХ датчиков темпе-

ратуры не только создала условия для эффективного использования процессорного времени, но и памяти.

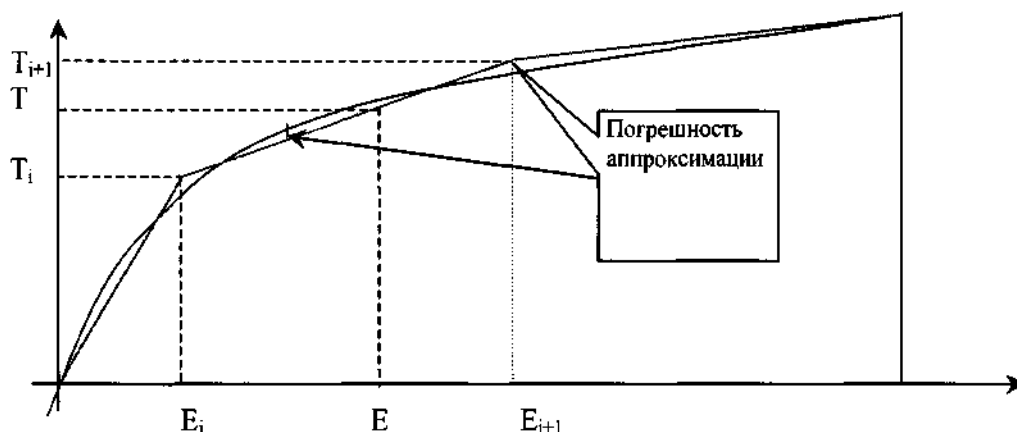


Рис. В.4. График линейного уравнения связи.

В.2.3 Компенсация влияния термоэлектродвижущей силы свободных концов термопар

Необходимость компенсации термоэлектродвижущей силы свободных концов термопар вызвана физическим явлением, при котором в спае рабочего конца, помещенного в измеряемую (рабочую) среду с температурой t_i и спае свободного конца с температурой t_0 возникают термоэлектродвижущие силы E_{i1} и E_{i0} соответственно, направленные встречно. В цепи термопары действует при этом результирующая термоэлектродвижущая сила $E_p = E_{i1} - E_{i0}$.

Из уравнения следует метод компенсации влияния электродвижущей силы свободного конца: для получения термоэлектродвижущей силы рабочего конца термопары к результирующей термоэлектродвижущей силе E_p следует прибавить величину термоэлектродвижущей силы E_{i0} ее свободного конца, то есть $E_{i1} = E_p + E_{i0}$.

Заметим, что подменить приведенное выше выражение простой разностью температур рабочего и свободного конца термопары с целью получения действительной температуры рабочей среды $T_p = T_1 - T_0$, в общем случае, нельзя из-за нелинейной зависимости термоэлектродвижущей силы от температуры. Такой способ, хотя он и довольно широко применяется на практике, может привести к значительным погрешностям определения температуры T_p .

К сожалению, прямые методы измерения E_{i0} отсутствуют. Для ее оценивания применяется, в этой связи, метод косвенных измерений. Сущность метода состоит в измерении T_0 и последующем расчете по известной температуре T_0 величины термоЭДС, которая была бы развита данной термопарой при данной температуре. Для измерения температуры T_0 применяются медные и, реже, платиновые термометры сопротивления.

При известном значении T_0 расчет E_{i0} осуществляется в порядке, обратном порядку, описанному в п. В.2.2: по значению температуры T_0 по номинальной статической характеристике термопары вычисляется значение E_{i0} , которое преобразуется затем в относительные единицы $N_{E_{i0}}$ - масштаб кодов аналого-цифрового преобразования канала измерения результирующей термоЭДС. Полученный таким образом код $N_{E_{i0}}$ и есть та величина компенсирующей поправки, на которую ослаблена (или усилена, в зависимости от знака E_0) результирующая термо-

ЭДС, а величиной, эквивалентной электродвижущей силе $E_{t1} = E_p + E_{t0}$ является ее (электродвижущей силы) код $N_{E_{t1}} = N_{E_p} + N_{E_{t0}}$.

В.3 Масштабирование результатов измерений

Под масштабированием результатов измерений понимается процедура преобразования измеренных значений параметров технологического процесса из одной системы физических единиц в другую. Чаще всего необходимость в масштабировании возникает в связи с преобразованиями кодов сигналов аналого-цифрового преобразования в формат физических величин выбранной размерности - давление (в kg/cm^2), расход (в m^3/H), температура (в $^{\circ}C$), напряжение (V или mV) и др. Такие же преобразования, но в обратном порядке, выполняются при выводе управляющих воздействий на регулирующие органы исполнительных механизмов.

Так как измерительные каналы ПТК обладают линейными характеристиками преобразования, то вид функций масштабных преобразований зависит только от вида статических характеристик датчиков. Измерительные каналы, в состав которых входят датчики сигналов с линейными номинальными статическими характеристиками имеют линейные уравнения масштабных преобразований. Каналы с нелинейными характеристиками датчиков, как, например, рассмотренные выше термопары или термометры сопротивления, требуют нелинейных, учитывающих нелинейность датчиков, уравнений для масштабных преобразований.

Масштабные преобразования каналов с линейными компонентами (датчиками, преобразователями) специфицируются верхним Y_{max} и нижним Y_{min} пределами шкалы измерения параметра и верхним N_{max} и нижним N_{min} пределами кода преобразования выходного сигнала датчика. Пределы шкалы приводятся, как правило, в проектной документации на АСУ ТП, а именно, в документе "Перечень выходных сигналов и данных". Пределы диапазона изменения кода являются характеристикой измерительного канала, на вход которого подан сигнал датчика, одна из его конструктивных характеристик. В случае линейной связи между входом в измерительный канал и его выходом или после линеаризации измерений масштабные преобразования кодов осуществляются в соответствии с уравнением:

$$Y = \frac{Y_{max} - Y_{min}}{N_{max} - N_{min}} * (N - N_{max}) + Y_{min}$$

При наличии нелинейных компонент в измерительном канале масштабные преобразования выполняются с учетом нелинейности. При этом если нелинейность описывается уравнением $Y = f(x)$, где x - выходной сигнал датчика и x преобразуется измерительными каналами в код

$$N_i = Ent[(x_i + 0,5h / h)], \quad i = 1, 2, \dots,$$

то после масштабирования имеем $Y_i = f(i)$, где $x_i = N_i h$.

Другой областью применения масштабирования в информационно-измерительных и управляющих системах являются преобразования результатов измерения, представленных в масштабе кодов аналого-цифрового преобразования, во внутренний, выбранный в качестве стандарта внутримашинного межпрограммного интерфейса передачи данных. Иллюстрацией необходимости выполнения таких преобразований может служить следующая ситуация.

Стандарты на нормированные выходные сигналы датчиков разрешают применение как сигналов 0, ..., 5 и 1, ..., 5, так и сигналов 0, ..., 20 и 4, ..., 20 mA. Для целей же преобразования сигналов тока в код применяются в ПТК, как правило, аналого-цифровые преобразователи с диапазоном входных сигналов 0, ..., 5 или 0, ..., 20 mA. При таком техническом решении теряется возможность контроля зашкаливания сигналов за нижний предел диапазона допустимых изменений, что снижает эксплуатационные свойства ПТК. Очевидно также, что таким образом спроектированный измерительный канал не использует 1/5 часть диапазона представления кодов (кодов, предназначенных для представления токов от 0 до 1 или от 0 до 4 mA). Восстанов-

ление свойств каналов приема сигналов 1, ..., 5 или 4, ..., 20 mA достигается посредством применения следующей процедуры масштабирования:

$$N'_i = \begin{cases} \frac{N'_k - N'_H}{N_{Yk} - N_{YH}}(N_i - N_{YH}) + N'_H, & \text{если } N_{YH} \leq N_i \leq N_{Yk}. \end{cases}$$

Данные достоверны, если $N_i < N_{YH}$ или $N_i > N_{Yk}$.

Соответствие между диапазонами представления данных показано на рис. В.5.

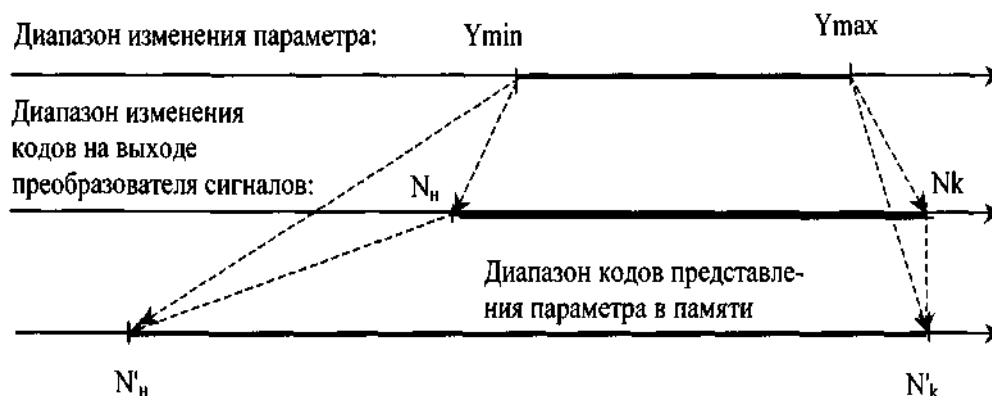


Рис. В.5. Соответствие между диапазонами представления данных.

Для одного из наиболее часто встречающихся случаев измерения сигналов тока 1, ..., 5 mA на каналах приема сигналов 0, ..., 5 mA (с диапазоном кодов преобразователя от 0 до 32000) рабочий диапазон составляют коды диапазона $(N_H, N_k) = (6400, 32000)$. Для преобразования кодов диапазона $(6400, 32000)$ в диапазон $(N'_H, N'_k) = (-32752, 31752)$ масштабное преобразование приобретает вид:

$$N' = \frac{2 \cdot 32752}{25600}(N - 6400) - 32752.$$

Очевидно, в частности, что токи $I = 1 \text{ mA}$ (начало диапазона) в масштабе нового диапазона будут представляться кодами $N' = -32752$, а токи $I = 5 \text{ mA}$ - кодами $+32752$. В случае приема тока $I < 1 \text{ mA}$ будет выдаваться предупреждающее диагностическое сообщение.

АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ IBM PC-КОНТРОЛЛЕРОВ

Архитектура и структура процессоров семейства INTEL достаточно полно представлены в различной литературе. Далее рассмотрены лишь те архитектурные особенности, которые позволяют реализовывать на их основе многозадачные операционные системы реального времени промышленного применения.

В процессоре 80286 был реализован *защищенный режим работы*. В защищенном режиме программа может записывать данные только в те области памяти, которые ей выделены операционной системой. Это создало предпосылки для разработки многозадачных многопользовательских операционных систем. В таких системах изолирование адресных пространств задач различных пользователей полностью исключает некорректные взаимодействия задач. В последующих моделях процессоров Intel реализованы расширение адресного пространства до 4 GB и концепция страничной виртуальной памяти. Механизм страничной виртуальной памяти позволяет поместить основную часть оперативной памяти задач на диске. При этом размер оперативной (виртуальной) памяти, предоставляемой программам, ограничен размером свободного пространства на диске.

Помимо страничной виртуальной памяти в процессорах Intel реализован режим виртуального процессора 8086. Этот режим действует в рамках защищенного режима (процессор может переключиться в виртуальный режим только из защищенного режима). В виртуальном режиме процессор способен выполнять программы, составленные для MS-DOS, находясь в защищенном режиме и используя аппаратные средства защищенного режима: мультизадачность, изолирование адресных пространств отдельных задач друг от друга, страничную виртуальную память. Наличие виртуального режима значительно облегчает проектирование операционных систем, способных выполнять одновременно в мультизадачном режиме несколько программ.

Основные особенности защищенного режима: возможность непосредственной адресации памяти за пределами первого мегабайта; реализация механизма страничной виртуальной памяти; аппаратная поддержка многозадачности, позволяющая создавать высокопроизводительные многозадачные, многопользовательские ОС; работа нескольких программ, составленных для MS-DOS, основанная на использовании виртуального режима работы процессора.

Защищенный режим процессора I80286 доступен и на всех более поздних процессорах (80386, 80486, Pentium). Основные отличия процессора 80386 от 80286 - это увеличение разрядности данных и адресов до 32 b и страничная организация памяти как еще один уровень преобразования между логической схемой (сегмент: смещение). Архитектура процессоров 80486 и Pentium принципиально не отличается от архитектуры 80386. Процессоры 80486 и Pentium работают в тех же режимах, что и 80386 процессор. Основное отличие - это более высокое быстродействие.

Механизм адресации памяти в защищенном режиме описан в литературе [79].

Г.1 Адресация памяти

Сущность организации памяти в защищенном режиме состоит в том, что содержимое регистра сегментов не определяет базовое значение физического адреса сегмента памяти, как это было в реальном режиме работы. Напротив, значение регистра сегментов в защищенном режиме является так называемым *селектором*. Селектор в данном случае является указателем для специальной таблицы. Эта таблица содержит полную информацию о распределении сегментов памяти в системе. Элемент таблицы, относящийся к одному сегменту, называется *дескриптором сегмента*. Дескриптор сегмента содержит всю необходимую информацию для указания раздела физической памяти, к которому система будет обращаться.

В системе имеются две различные таблицы для дескрипторов сегментов. Это *таблица глобальных дескрипторов* (GDT) и *таблица локальных дескрипторов* (LDT). Вторым битом значения селектора определяется выбор таблицы дескрипторов LDT или GDT. После того как выбрана соответствующая таблица, по значению 3 - 15 битов селектора выбирается один из 8192 возможных сегментов. Длина дескриптора сегмента составляет 8 байт. Содержимым дескриптора сегмента являются:

- базовый адрес физического сегмента - 24 бит;
- размер сегмента - 16 бит;
- наличие (отсутствие) сегмента в физической памяти;
- флаг обращения к данному селектору;
- вид сегмента - программный, данные или специальный;
- разрешение (или запрет) считывания содержимого программного сегмента;
- разрешение (или запрет) записи в сегмент данных;
- информация об уровнях приоритета.

Таблица глобальных дескрипторов (GDT) - единственная в системе. Обычно в ней находится описание сегментов операционной системы. Таблиц LDT может быть много. Эти таблицы содержат описания сегментов программ, работающих под управлением ОС. В каждый данный момент времени процессор может использовать только одну таблицу LDT. Операционная система должна создавать отдельные таблицы локальных дескрипторов для каждого набора программных сегментов, который необходимо изолировать от других наборов программных сегментов. 24-разрядный базовый адрес сегмента указывает адрес начала сегмента в физической памяти. Значение 16-разрядного смещения, являющегося второй составляющей полного 24-разрядного физического адреса, формируемого из текущего значения дескриптора, служит для определения окончательного реального (физического) адреса в памяти. В защищенном режиме сегменты могут быть различных размеров - размер каждого сегмента определяется в программе. Это позволяет экономно использовать реальную память системы и виртуальную память на диске.

Микропроцессор в защищенном режиме позволяет организовать защиту памяти и установить приоритеты. Формирование отдельной таблицы локальных дескрипторов для каждой программы позволяет операционной системе обеспечить полную защиту любой программы от других программ. Какое бы значение селектора ни сформировала бы программа, ей все равно не удастся обратиться к дескриптору сегмента, находящемуся в чужой таблице LDT. Содержимое дескриптора сегмента используется для проверки корректности каждого обращения в память. Если какое-либо из условий защиты памяти не выполняется, процессор генерирует особую ситуацию. Кроме того, если процессор пытается записать в регистр сегментов полученное с помощью селектора значение дескриптора сегмента, соответствующее отсутствующему в реальной памяти сегменту, то снова возникает особая ситуация. В этом случае операционная система должна отыскать сегмент на диске, перекачать его в память, изменить соответствующий ему дескриптор так, чтобы он содержал информацию об имеющемся в памяти сегменте, и завершить обслуживание особой ситуации, что позволяет продолжить выполнение программы. Для запрета доступа программы к таблице глобальных дескрипторов существует механизм приоритетов. Основная его идея состоит в том, что программы могут обращаться только к тем данным, которые соответствуют их уровню приоритета или имеют более низкий уровень. Для этой цели в микропроцессоре существуют четыре уровня приоритета. Они имеют значения от 0 до 3, где 0 соответствует высшему приоритету, а 3 - низшему. Устанавливать приоритеты должна операционная система. Обычно ядру операционной системы присваивается приоритет 0, а прикладным программам - приоритет - 3. Уровень 1 присваивается программам обслуживания аппаратуры, драйверам, уровень 2 системам управления файлами и расширениям ОС.

Механизм приоритетов предохраняет ОС от работающих программ и позволяет ОС обратиться к любой информации более низкого уровня приоритетного уровня.

Процессоры 80386/80486/Pentium в защищенном режиме используют трехступенчатую схему преобразования адресов.

Программы используют логический адрес, состоящий из селектора и смещения (аналогично процессору 80286). Селектор полностью аналогичен используемому в процессоре 80286. Компонента смещения является 32-разрядной, т.к. допустимый размер сегмента значительно превышает 64 Кбайта. Уровень логического адреса - это первая ступень в схеме преобразования адресов. Вторая ступень - получение из логического адреса 32-разрядного линейного адреса. Линейный адрес берется из глобальной или локальной таблицы дескрипторов (GDT или LDT) в зависимости от соответствующего бита селектора (бит 2). Механизм получения линейного адреса напоминает механизм получения 24-разрядного физического адреса в процессоре 80286. Однако линейный адрес не отображается непосредственно на адресную шину памяти, т.е. он не является физическим адресом. Для получения из линейного адреса физического адреса используется третья ступень - механизм страничной адресации. С помощью этого механизма 20 старших бит линейного адреса используются для выбора блока памяти размером 4 Кбайта. Такой блок называется страницей физической памяти. Оставшиеся 12 бит линейного адреса представляют собой смещение внутри страницы.

Процесс вычисления адреса страницы часто называют трансляцией страниц. Старшие 10 бит линейного адреса используются как индекс в таблице, называемой каталогом таблиц страниц. Расположения каталога таблиц страниц в физической памяти определяется содержимым системного регистра процессора CR3. Каталог таблиц страниц содержит дескрипторы таблиц страниц, определяющие физический адрес таблиц страниц. В каталоге всего может быть 1024 дескриптора. Самых же каталогов может быть сколько угодно, но в каждый момент времени используется только один - тот, на который указывает регистр CR3. Следующие 10 бит линейного адреса предназначены для индексации таблицы страниц, выбранной с помощью старших 10 бит адреса. Таблица страниц содержит 1024 дескриптора, определяющие физические адреса страниц памяти. Размер одной страницы составляет 4 Кбайта, т.е. 4096 байт. Младшие 12 бит линейного адреса указывают смещение к адресуемому байту внутри страницы. Для представления старших 20 битов физического адреса таблицы страниц в дескрипторе используются биты 12-31. Младшие 12 битов адреса таблицы всегда равны нулю, таким образом, таблица страниц должна быть выровнена в памяти на границу 4096 байт (на границу страницы). Для процессора 80386 добавилось еще два типа таблиц, содержащих дескрипторы и соответственно два типа дескрипторов - дескриптор таблиц страниц и дескриптор страниц.

Для использования механизма трансляции страниц операционная система должна установить в 1 старший бит системного регистра CR0. Если этот бит не установлен в 1, физический адрес будет равен линейному, содержимое регистра адреса каталога таблиц страниц CR3 при этом для преобразования адреса использоваться не будет. Форматы дескрипторов, располагающихся в таблицах GDT, LDT, IDT, претерпели изменения по сравнению с используемыми в процессоре 80286. Например, вместо 24-битового физического адреса в дескрипторах должен находиться 32-битовый линейный адрес.

Процессор 80386 использует 32-разрядный базовый адрес сегмента и 20-разрядное поле предела. В зарезервированном для процессора 80286 поле в битах 24-31 находится старший байт 32-разрядного базового адреса сегмента. Биты 16-19 используются для хранения старших четырех битов предела.

В процессоре 80286 поле предела указывало размер сегмента в байтах. Для процессора 80486 интерпретация поля предела зависит от установки бита G - бита гранулярности. Если бит G установлен в 1, поле предела содержит размер сегмента в страницах (размером 4096 байт). Если бит G сброшен, размер сегмента вычисляется в байтах. Бит гранулярности G также находится в поле, которое в процессоре 80286 было отмечено как зарезервированное.

Поле X указывает разрядность выполняемых команд, принятых по умолчанию. Если этот бит установлен в 1, используются 32-разрядные команды, если сброшен в ноль - 16-разрядные. Бит AVL используется системным программным обеспечением. Бит S-признак системного сегмента. Если этот бит сброшен в ноль, то сегмент системный. Назначение остальных полей дескриптора аналогично используемому в процессоре 80286.

Г2. Обработка прерываний

В защищенном режиме все прерывания разделяют на два типа - обычные прерывания и исключения (особые случаи). Обычное прерывание инициируется командой INT (программное прерывание) или внешним событием (аппаратное прерывание). Перед передачей управления процедуре обработки обычного прерывания флаг разрешения прерываний IF сбрасывается и прерывания запрещаются. Обработка прерываний базируется на таблице прерываний. Таблица прерываний защищенного режима называется *таблицей дескрипторов прерываний* (IDT). Так же как и таблицы GDT и LDT, таблица IDT содержит 8-байтовые дескрипторы. Эта таблица может содержать до 256 элементов. Дескриптор прерываний называется *шлюзом* или *вентилем*. Каждый шлюз содержит ссылку на соответствующий обработчик прерываний, который связан с номером прерывания. Шлюзы позволяют также осуществить передачу управления между различными уровнями приоритетов. Расположение таблицы IDT определяется содержимым 5-байтового внутреннего регистра процессора IDTR. Формат регистра IDTR полностью аналогичен формату регистра GDTR, для его загрузки используется команда LIDT. Регистр IDTR содержит 24-битовый физический адрес дескрипторной таблицы прерываний IDT и ее предел.

Для обработки исключений зарезервирован 31 номер прерывания. Перед тем, как передать управление обработчику исключения, процессор помещает в стек 16-битовый код ошибки. Помещение кода ошибки производится только для следующих исключений: двойная ошибка; недействительный TSS; отсутствие сегмента в памяти; исключение при работе со стеком; исключение по защите памяти. Обработку прерываний от внешних устройств, как правило, выполняют драйверы этих устройств.

Процессоры 80386, 80486, Pentium. в защищенном режиме обрабатывают прерывания точно так же, как и процессор 80286. Есть отличия в формате дескрипторов, располагающихся в дескрипторной таблице прерываний IDT. Эти отличия заключаются в использовании других значений в поле и в том, что два байта, зарезервированные в вентилях прерывания и исключения процессора 80286, используются процессорами для хранения битов 16-31 32- разрядного смещения.

Г.3 Средства мультизадачности

Для организации мультизадачного режима предусмотрено автоматическое сохранение контекста задачи в специально выделенной для каждой программы области памяти. Под контекстом программы понимается содержимое регистров процессора. Для хранения контекста неактивной в настоящий момент задачи процессор использует специальную область памяти, называемую сегментом состояния задачи TSS (Task State Segment). Сегмент TSS адресуется процессором при помощи 16-битного регистра TR (Task Register), содержащего селектор дескриптора TSS, находящегося в глобальной таблице дескрипторов GDT. Поле доступа содержит бит В - бит занятости. Если задача активна, этот бит устанавливается процессором в 1.

Операционная система для каждой задачи создает свой TSS. Перед тем как переключиться на выполнение новой задачи, процессор сохраняет контекст старой задачи в ее сегменте TSS. В TSS записывается содержимое регистров общего назначения AX, BX, CX, DX, регистров SP, BP, SI, DI, сегментных регистров ES, CS, SS, DS, содержимое указателя команд IP и регистра флажков FLAGS. Кроме того, сохраняется содержимое регистра LDTR, определяющего локальное адресное пространство задачи. Дополнительно при переключении задачи в область TSS со смещением 44 операционная система может записать любую информацию, которая относится к данной задаче.

Поле Link представляет собой поле обратной связи и используется для организации вложенных вызовов задач. Поля Stack 0, Stack 1, Stack 2, хранят логические адреса (селектор: смещение) стеков, отдельных для каждого уровня приоритета. Эти поля используются при межсегментных вызовах через шлюзы.

Для обеспечения защиты данных процессор назначает отдельные стеки для каждого уровня приоритетов. Когда задача вызывает подпрограмму из другого уровня через шлюз вызова, процессор вначале загружает указатель стека SS:SP адресом нового стека, взятого из соответствующего поля TSS. Затем в новый стек копируется содержимое регистров SS:SP задачи (т.е. адрес вершины старого стека задачи). После этого в новый стек копируются параметры, количество которых задано в шлюзе вызова, и адрес возврата. Таким образом, при вызове привилегированного модуля через шлюз вызова менее привилегированная программа не может передать в стек больше параметров, чем это определено операционной системой для данного модуля. Включение адресов стеков в TSS позволяет разделить стеки задач и обеспечивает их автоматическое переключение при переключении задач.

Для переключения задач имеются следующие возможности: переключение по командам JMP или CALL и переключение по прерыванию. В первом случае используются обычные команды JMP или CALL, но в качестве операнда в этих командах указывается адрес сегмента TSS задачи, на которую необходимо переключиться. Если произошло переключение при помощи команды JMP, то для возврата к выполнению предыдущей задачи необходимо вновь использовать команду JMP, указав в качестве операнда адрес TSS этой задачи.

Команда CALL позволяет организовать вызов вложенных задач. Возврат к предыдущей задаче осуществляется по команде IRET. Адрес TSS для возврата команда IRET берет из поля обратной связи LINK текущего сегмента TSS, куда он был записан командой CALL при первом переключении задач. Для реализации переключения по прерыванию необходимо поместить шлюз задачи в дескрипторную таблицу прерываний IDT.

В процессорах 80386, 80486, Pentium получила дальнейшее развитие аппаратная поддержка мультизадачности, впервые введенная в процессоре 80286. На рис. Г.1 приведен формат сегмента TSS для процессоров 80386, 80486, Pentium. Из рисунка видно, что в TSS предусмотрены поля для хранения сегментных регистров GS, FS, SS, DS, CS. Имеется поле для хранения содержимого регистра LDTR, указывающего на локальную таблицу дескрипторов, распределенную данной задаче. Для хранения содержимого 32-разрядных регистров используются поля TSS, обозначенные на рисунке как EDI, ESI, EBF, ESP, EBX, EDX, ECX, EAX, EFLAGS, EIP. Поле CR3 хранит содержимое системного регистра CR3. Этот регистр является указателем на каталог таблиц страниц. Таким образом, каждая задача может иметь свой собственный каталог таблиц страниц, что позволяет выполнить изоляцию задач не только на уровне сегментов, но и на уровне страниц.

TSS процессора содержит указатели на стеки для второго, первого и нулевого приоритетных колец. Это поля SS2:ESP2, SS1:ESP1, SS0:ESP0. Поле LINK используется для ссылки на TSS вызвавшей задачи при вложенном вызове задач, аналогично тому, как это было в процессоре 80286. Бит Т используется для отладки. Если он установлен в 1, при переключении на задачу возникает отладочное исключение, которое может быть использовано системным отладчиком. Еще одно новшество - битовая карта ввода/вывода. Для обеспечения безопасной работы системы необходимо ограничить доступ программ пользователя ко всем или, по крайней мере, некоторым портам ввода/вывода.

Некорректная программа, имеющая доступ к портам контроллера прямого доступа к памяти, может выполнить с помощью этого контроллера чтение и запись информации по любым физическим адресам. Процессор 80286 хранит в регистре флагов уровень привилегий IOPL, на котором разрешено выполнять команды ввода/вывода. С помощью этого механизма можно запретить непривилегированным программам выполнять команды ввода/вывода. Однако, такой способ защиты не слишком удобен. Некоторые порты ввода/вывода не только безопасны для использования, но и весьма полезны для обычных программ.

Битовая карта ввода-вывода процессора 80386 позволяет для каждой задачи определить порты, которые эта задача может использовать. То есть, операционная система имеет возможность санкционировать любую задачу для использования любого набора адресов портов ввода/вывода. Если задача пытается обратиться к несанкционированному порту ввода/вывода, произойдет особый случай.

БАЗА КАРТЫ ВВОДА - ВЫВОДА	0	T
0	LDTR	
0	GS	
0	FS	
0	DS	
0	SS	
0	CS	
0	ES	
EDI		
ESI		
EBP		
ESP		
EBX		
EDX		
ECX		
EAX		
EFLAGX		
EIP		
CR3		
0	SS2	
ESP2		
0	SS1	
ESP1		
0	SS0	
ESP0		
0	LINK	

Рис.Г.1. Сегмент TSS процессоров 80386, 80486, Pentium.

Поле базы карты ввода/вывода указывает 16-разрядное смещение начала битовой карты ввода/вывода относительно TSS. Предел TSS должен определяться с учетом карты. Каждый бит в карте ввода/вывода соответствует адресу байта порта ввода/вывода. После битовой карты должен располагаться байт 0FFH. При выполнении 16- или 32- разрядных операций ввода/вывода процессор проверяет все биты (2 или 4), соответствующие адресу порта. Если проверяемый бит установлен в 1, происходит особый случай.

Для привилегированных программ есть уровень привилегий меньше или равный уровню IOPL, процессор не выполняет проверку битовой карты ввода/вывода. Чтобы полностью запретить задаче обращаться к портам ввода/вывода, достаточно установить базу карты ввода/вывода большей или равной пределу TSS. В этом случае любая команда ввода/вывода приведет к генерации особого случая.

ХАРАКТЕРИСТИКИ БЛОКОВ МСКУ

Д.1 КОНТРОЛЛЕР МИКРОПРОЦЕССОРНЫЙ КМП-20

Контроллер микропроцессорный КМП-20 является дальнейшим развитием ряда контроллеров, используемых в МСКУ. Основой КМП-20 служит набор стандартных модулей PC/104, устанавливаемых на пользовательской несущей плате типа E2. Это обеспечивает гибкость комплексов и возможность модернизации путём простой замены модулей PC/104.

Стандарт PC/104 имеет полную совместимость с IBM PC по архитектуре, аппаратной и программной части, но в компактном (90x96mm) варианте модулей. Шина PC/104 полностью соответствует 16-разрядной шине ISA.

КМП-20 состоит из следующих узлов: процессорного модуля (ПМ), выполняющего обработку информации и управляющего другими узлами; узла связи с другими контроллерами; узла управления устройствами связи с объектом по интерфейсу ИР; узла сетевого интерфейса (блока связи интерфейсного – БСИ). Структурная схема КМП-20 представлена на рис. Д.1.

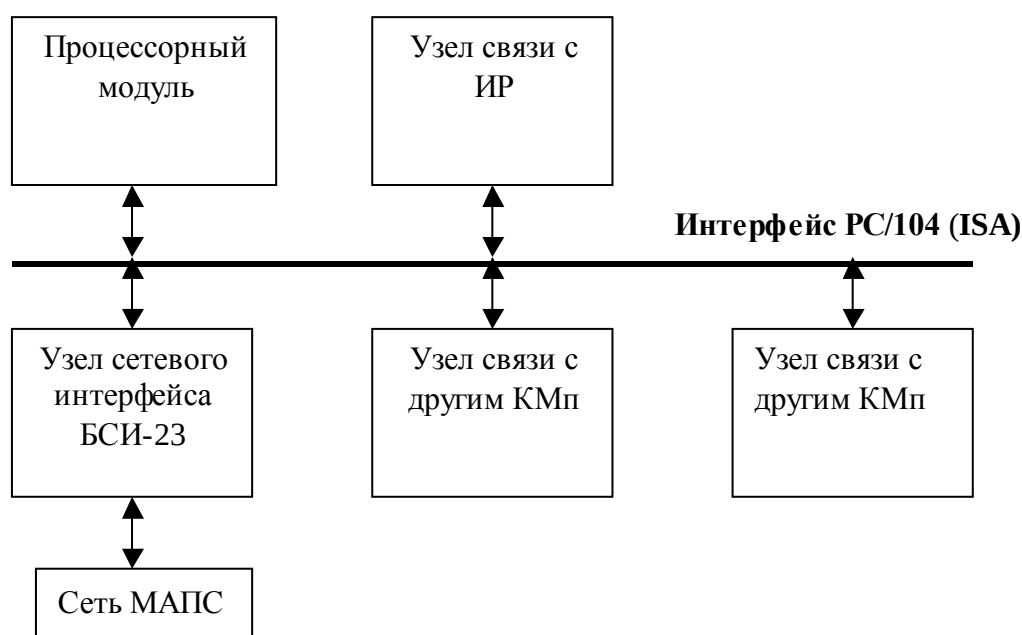
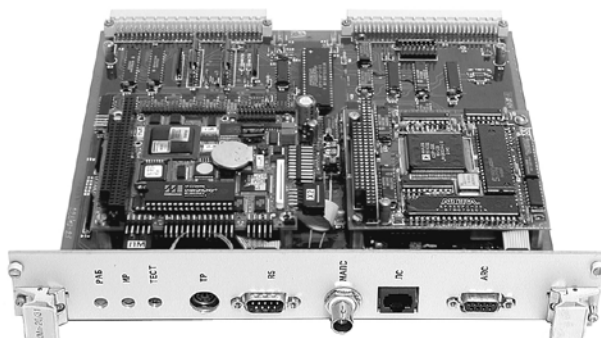


Рис. Д.1. Структурная схема КМП-20.

Рис. Д.2. Внешний вид устройства КМП-20.



В основных конфигурациях КМП-20 используются процессорные модули MSM486SL2 и MSM586SN фирмы DIGITAL LOGIC. ПМ на основе процессорного модуля MSM486SL2 фирмы DIGITAL LOGIC имеет следующие технические характеристики: процессор – 486 с частотой 66 MHz; BIOS ROM; динамическое ОЗУ емкостью 2 MB; три

последовательных (RS-485, RS-232C, RS-232C) и один параллельный порты; электронный диск: Disk on Chip до 72 MB; часы реального времени.

ПМ на основе процессорного модуля MSM586SN имеет следующие характеристики: процессор AMDELAN520 133 MHz; BIOS ROM; динамическое ОЗУ емкостью 32 MB; сопроцессор; четыре последовательных RS-232C и один параллельный порты; электронный диск: Disk On Chip до 72 MB; программируемый сторожевой таймер и часы реального времени.

Д.1.1 БЛОК СВЯЗИ С МАПС

МСКУ через блок связи интерфейсный БСИ (далее – БСИ) имеет выход в сеть МАПС. БСИ – это устройство на основе цифрового сигнального процессора ADSP-2181 фирмы Analog Devices и многофункционального связного контроллера SCN2652 фирмы Philips Electronics. Для взаимодействия БСИ с ПМ используется системный интерфейс промышленного стандарта ISA. Функционально БСИ состоит из четырёх основных узлов: узел сопряжения с ISA; узел сигнального процессора ADSP-2181; узел связного контроллера SCN2652 и узел сопряжения с МАПС (модем). Структурная схема БСИ приведена на рис. Д.3.

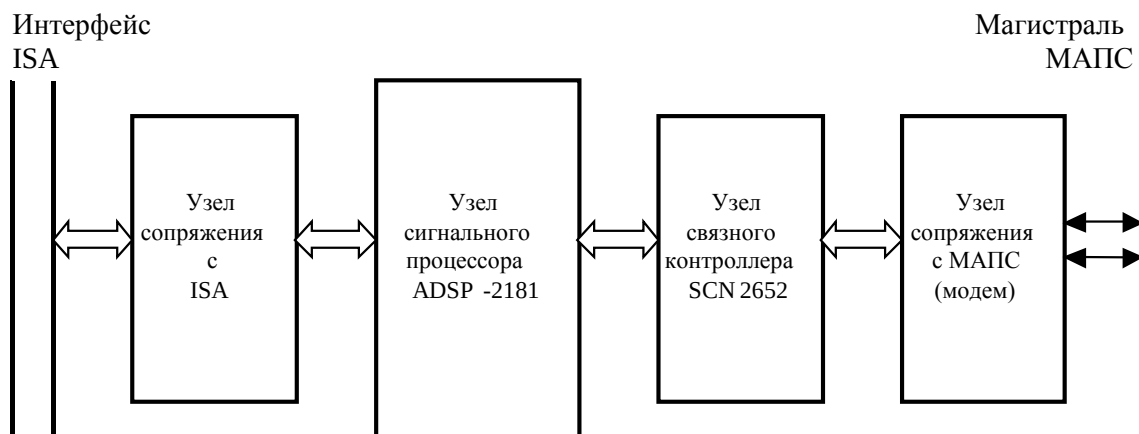


Рис. Д.3. Структура БСИ.

Центральным узлом БСИ является узел сигнального процессора ADSP-2181, который управляет работой всего устройства. ADSP-2181 – HARC-микропроцессор (Harvard Architecture Computer) фирмы Analog Devices, который обладает производительностью 32 MIPS при тактовой частоте 32 MHz и оптимизирован для высокоскоростной цифровой обработки сигналов. ADSP-2181 имеет базовую архитектуру семейства ADSP-2100 (три вычислительных устройства, два генератора адресов данных и блок управления последовательностью выполнения программы), дополненную двумя последовательными портами, 16-разрядным внутренним портом прямого доступа к памяти, 8-разрядным внешним портом прямого доступа к памяти, программируемым таймером, флаговыми входами-выходами, расширенными возможностями по обработке прерываний и встроенной памятью программ и данных.

В БСИ используются следующие аппаратные ресурсы сигнального процессора ADSP-2181:

- память программ ёмкостью 48 KB (16384 24-разрядных слова) – применяется для хранения программы функционирования БСИ;
- память данных ёмкостью 32 KB (16384 16-разрядных слова) – применяется для организации интерфейсной области памяти, обеспечивающей взаимодействие БСИ с ПМ, а также для буферирования сообщений, выдаваемых в магистраль и принимаемых из магистрали;
- три вычислительных устройства (арифметико-логическое устройство, умножитель-аккумулятор и сдвигатель) – применяются для обработки и анализа данных в процессе функционирования БСИ;

- внутренний порт прямого доступа к памяти – применяется для сопряжения БСИ с интерфейсом ISA;

- внешний параллельный порт – применяется для сопряжения ADSP-2181 со связным контроллером SCN2652;

- два последовательных порта, флаговые входы-выходы и входы запросов на прерывание – применяются для управления работой связного контроллера SCN2652 и модема.

В узел сигнального процессора ADSP-2181 также входят кварцевый генератор H5C-2E 16MHz, задающий тактовую частоту процессора, и супервизор ADM699, обеспечивающий формирование сигнала первоначального сброса процессора в течение 140 ms после подачи напряжения питания на БСИ. Узел сопряжения с ISA обеспечивает взаимодействие БСИ с ПМ и выполняет следующие функции: дешифрация адреса БСИ и кода операции (запись-чтение данных, запись адреса данных, прерывание или сброс БСИ); согласование шин данных интерфейса ISA и внутреннего порта прямого доступа к памяти процессора ADSP-2181; формирование запроса прерывания (IRQ) ПМ; формирование других необходимых сигналов интерфейса ISA.

Узел связного контроллера SCN2652 обеспечивает работу БСИ на магистрали МАПС в соответствии с принятым в сети МАПС бит-ориентированным протоколом обмена. SCN2652 – многопротокольный программируемый связной контроллер фирмы Philips Electronics, который обеспечивает передачу и приём последовательных синхронных данных со скоростью до 2 Mbit/s и поддерживает несколько бит- и байт-ориентированных протоколов обмена.

В БСИ связной контроллер SCN2652 работает в следующем режиме: бит-ориентированный протокол обмена SDLC (Synchronous Data Link Control) со скоростью приёма-передачи данных 1 Mbit/s; автоматическое генерирование и распознавание символов начала и конца сообщения; автоматическое распознавание адреса БСИ на магистрали МАПС; автоматическая проверка контрольной суммы сообщения с использованием циклического полинома деления.

Узел сопряжения с МАПС (модем) выполняет модуляцию передаваемых последовательных данных и синхронизирующего сигнала в двухполярный сигнал с фазоразностным кодированием, выдаваемый в магистраль МАПС, и демодуляцию принимаемых последовательных данных и синхронизирующего сигнала из двухполярного сигнала с фазоразностным кодированием, принимаемого из магистрали МАПС. Модем состоит из кодировщика-декодировщика (кодека) сигналов и блока приёма-передатчиков. БСИ подключается к ISA как 16-разрядное устройство ввода-вывода.

В адресном поле устройств ввода-вывода ПМ БСИ использует четыре 16-разрядных порта со смежными адресами. Базовый адрес ввода-вывода БСИ равен 3a0h. БСИ использует одну линию запроса прерывания ПМ – 5. Порты ввода-вывода БСИ имеют следующее назначение:

- 3a0h – порт записи данных в память БСИ и чтения данных из памяти БСИ;
- 3a0h+2 – порт записи адреса данных в памяти БСИ;
- 3a0h+4 – порт прерывания процессора БСИ;
- 3a0h+6 – порт сброса БСИ.

Обмен данными между БСИ и ПМ осуществляется через память БСИ, интегрированную в сигнальный процессор ADSP-2181. Память БСИ состоит из памяти программ (16384 24-разрядных слова) и памяти данных (16384 16-разрядных слова).

Д.2 МОДУЛЬ ПОСЛЕДОВАТЕЛЬНОЙ СВЯЗИ МПСВ-16

Модуль связи в составе МСКУ используется для подключения устройств ввода-вывода по 16 независимым последовательным интерфейсам типа: RS-485, полудуплекс; RS-485, полный дуплекс; RS-232C, полный дуплекс; ИРПС, токовая петля 20 mA, полный дуплекс; активный приемник и передатчик; пассивный приемник и передатчик; активный приемник и пассивный передатчик; активный передатчик и пассивный приемник.

Структурная схема МПСВ-16 приведена на рис. Д.4.

Модуль связи подключен к троированному интерфейсу ИР. Объем буферной памяти каждого порта ИР – не менее 4 КВ. Конструктивно модуль представляет собой печатную плату типа Е2, на которой размещаются узел интерфейса ИР, реализованный в ПЛИС, два 8-канальных универсальных микроконтроллера, flash-память для исполнительного программного модуля и установочные места для установки и крепления субмодулей ввода-вывода.

Все типы каналов модуля выполнены в виде одноканальных субмодулей ввода-вывода, соединенных с базовой платой через одно или двухрядные штыревые соединители. Установка соответствующих субмодулей на плату осуществляется в соответствии с картой заказа на МСКУ.

Вычислительный субмодуль с базовой платой также соединяется с помощью двухрядным штыревым соединителем. Построен на базе INTEL 386 EX, 33 MHz.

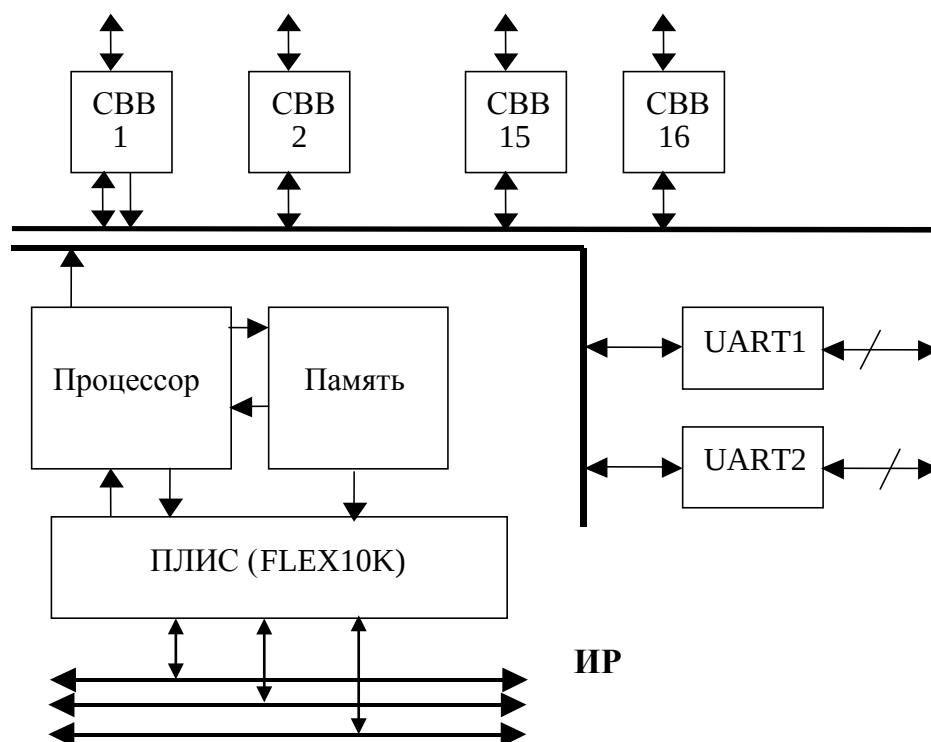


Рис. Д.4. Структурная схема МПСв-16: СВВ – субмодуль ввода-вывода.

Уровень передачи данных реализован на интерфейсном микроконтроллере SC28L198. Интерфейсный микроконтроллер содержит:

- восемь полно дуплексных UART каналов (промышленный стандарт Philips);
- буфер на 16 байтов в передатчике каждого канала;
- буфер на 16 байтов в приемнике каждого канала;
- встроенный аппарат формирования прерывания;
- сторожевой таймер приемника каждого канала;
- типовые режимы программной настройки параметров приемника и передатчика.
- шина передачи данных (30 ns время освобождения шины передачи данных, 125 ns) - цикл шины;
- аппаратуру приоритетных прерываний;
- следающий таймер для каждого приемника (64 счетчика синхросигналов приемника);
- 22 фиксированных диапазона: 50 - 230.4 Kbd или 100 - 460.8 Kbd;
- дополнительные ненормативные диапазоны до 500 Kbd с внутренними генераторами;
- два перезагружаемых счетчика, обеспечивающих генерацию дополнительных программируемых диапазонов;

- обнаружение ошибок: четность, кадрирование, переполнение;
- обнаружение обрыва на линии;
- локальный возврат к началу цикла.

Структурная схема восьмиканального приемопередатчика (UART) приведена на рис. Д.5.

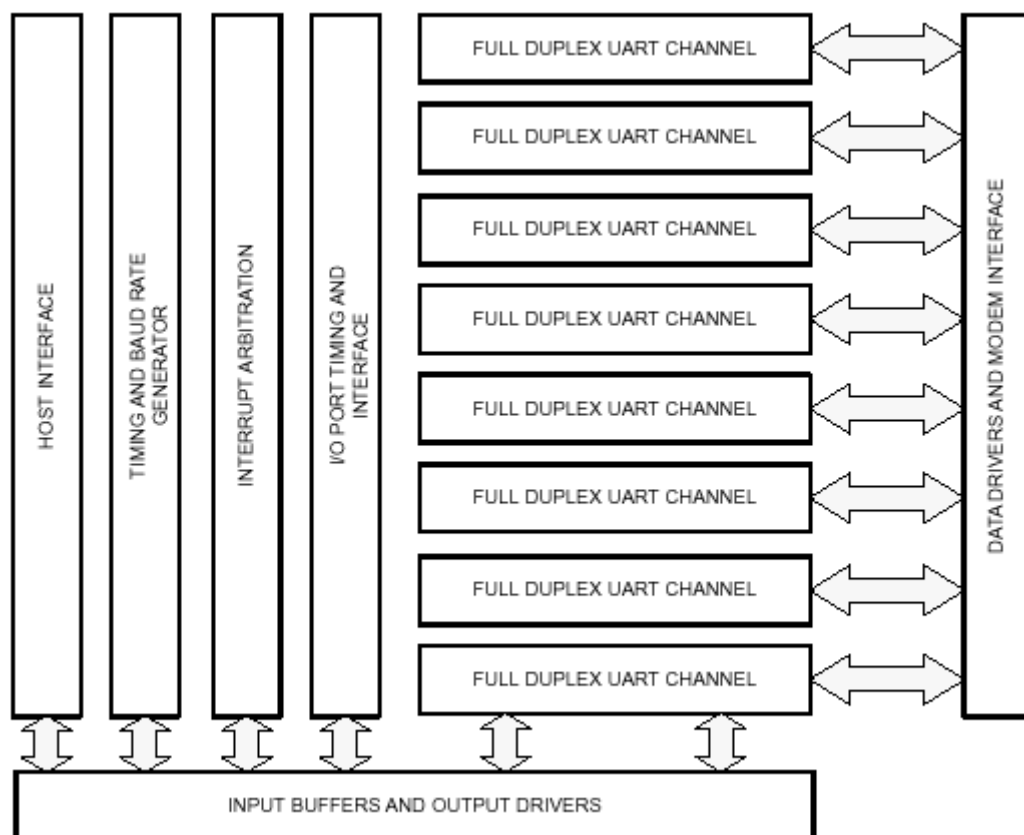


Рис. Д.5. Структурная схема восьмиканального приемопередатчика.

В состав UART входят: арбитр прерывания, ведущий (host) интерфейс, блоки синхронизации и восемь каналов UART. Восемь каналов работают независимо, взаимодействуя только с центральным процессором (host), I/F и блоками прерываний.

SC28L198 состоит из нескольких функциональных блоков:

- синхронный ведущий интерфейсный блок;
- блок синхронизации, состоящий из общего генератора скорости, обеспечивающего 22 скорости и два 16-разрядных счетчика, используемых для генерации нестандартной скорости;
- 4 идентичных независимых полно дуплексных UART канала;
- система арбитража прерываний, определяющая 24 конкурирующих запроса;
- блок управления портом ввода - вывода и детектор изменения состояния.

Д.3 БЛОКИ СВЯЗИ С ОБЪЕКТОМ

Ввод данных от объекта управления и вывод управляющих воздействий на объект управления осуществляется управляющей системой МСКУ через функциональные блоки связи с объектом. Их характеристики приведены в табл. Д.1.

Конструктивно каждый блок выполнен на плате типа Е2 с двумя 96-контактными соединителями Х1, Х2 для подключения его к контроллерам и одним 64-контактным соединителем Х3 для подключения сигналов датчиков. Внешний вид блоков связи с объектом показан на рис.Д.6. Далее рассмотрены особенности реализации блоков из каждой функциональной группы.

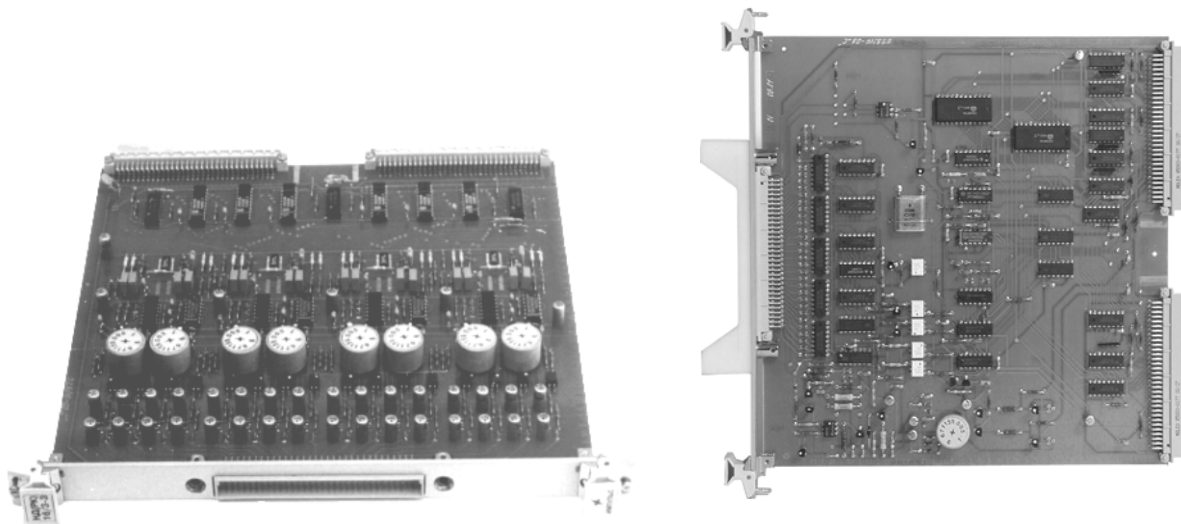


Рис. Д.6. Внешний вид блоков связи с объектом.

Таблица Д.1

Основные характеристики блоков связи с объектом

Функция, единица измерения	Диапазон изменения, значения параметра	Погрешность %	Шифр блока	Число каналов	Гальваническая развязка
1 Ввод аналоговых сигналов среднего уровня: напряжения, V тока, mA	$\pm 2,5; \pm 5; \pm 10$ $0-2,5; 0-5; 0-10$ $0-5; 0-20$	$\pm 0,10$	ПНК(И)-30/3	30	групповая
		$\pm 0,10$	ПНК(Р)-8/3	8	поканальная
		$\pm 0,10$	ПНК(Р)-1	16	поканальная
		$\pm 0,04/0,003$ $\pm 0,04/0,003$	ПНК(Р)-1М1 ПНК(Р)-1М2	16 16	поканальная поканальная
2 Ввод сигналов от преобразователей термоэлектрических, mV	$\pm 10; \pm 20; \pm 30; \pm 40; \pm 50; \pm 80; \pm 100$	$\pm 0,25$	ПТП-4/3 ПТП(В)-16/3	4 16	поканальная групповая
	$\pm 10; \pm 15; \pm 20; \pm 30; \pm 40; \pm 50; \pm 80;$	$\pm 0,04$	ПТП(П)-8/3	8	групповая
3 Ввод сигналов термопреобразователей сопротивления, Ω	0-50-100; 0-50-150 0-50-250, 0-100-200, 0-100-300, 0-100-500	$\pm 0,25$	ПТС-4/3 ПТС(В)-16/3	4 16	поканальная групповая
	25-50-75; 0-50-100 50-100-150; 0-100-200 100-150-200; 50-150-200 100-200-300; 0-200-400 100-300-500	$\pm 0,04$	ПТС(П)-8/3	8	групповая
4 Ввод аналоговых сигналов сопротивления, Ω	0-100; 0-200; 0-400	$\pm 0,02$	ПСК-2	8	поканальная
5 Ввод аналоговых сигналов низкого уровня: - напряжения, mV ; - тока, μA ; - проводимости, $1/\Omega$	минус 20-0-20; минус 40-0-40; минус 80-0-80 (0-5); $\pm 0,5$ От 0 до $5 \cdot 10^{-6}$	$\pm 0,02$	ПНК(Р)-2	16	групповая
		$\pm 0,05$ $\pm 0,20$ $\pm 2,0$	ПКИ-2/3	7	поканальная
6 Ввод аналоговых сигналов - напряжения низкого уровня, mV - сопротивления, Ω	$\pm 15;$ ± 30 30-50-70; 10-50-90	$\pm 0,04$ $\pm 0,1$ $\pm 0,04$ $\pm 0,1$	ПКИ-3/3	6 2	групповая

Продолжение табл. Д.1

7 Ввод сигналов от сельсинов, V	0-20	$\pm 0,2$	ПСС-1	8	поканальная
8 Ввод частотных сигналов, kHz	0-32; 0-16; 0-8; 0-4; 0-2; 0-1; 0-0,5; 0-0,25		ПЧ(РТ)-8/3 ПЧ1(РТ)-8/3	8 8	поканальная поканальная
9 Ввод время-импульсных сигналов		$\pm 0,1$	ПВИ(РТ)-8/3	8	поканальная
10 Ввод дискретных сигналов: - напряжения (уровни "0"/"1"), V - тока (уровни "0"/"1"), mA - напряжения (уровни 0"/"1"), V	(0-1,2)/(4,2-7,8); (0-2,4)/(9,6-14,4); (0-4,8)/(19,2-28,8); (0-9,6)/(38,4-57,6); (0-1)/(4-20) (0-4,8)/(19,2-28,8)		НД(РТ)-32/3, НД(РТФ)-32/3 НД(Р)-48 НД(РФ)-48	32 48	поканальная поканальная
11 Ввод сигналов от датчиков типа «сухой контакт»: - замкнутый контакт, Ω - разомкнутый контакт, $k\Omega$	менее 200 более 50		НД(РК)-16/3, НД(РКФ)-16/3 НД(РК)-24 НД(РКФ)-24	16 24	поканальная
12 Ввод сигналов от датчиков типа «сухой контакт» с контролем линии связи с датчиком: - замкнутый контакт, Ω - разомкнутый контакт, $k\Omega$ - разрыв связи, $k\Omega$	менее 200 от 1,3 до 2,2 более 16		НД(РК)-28/3	28	Групповая
13 Вывод аналоговых сигналов тока, mA	0-5; 0-20	$\pm 0,2$	ФТ-16	16	поканальная
14 Вывод дискретных сигналов: напряжение нагрузки, V/ток нагрузки, A	не более 48/0,2 не более 50/1,0		ФД(Р)-16/3 М Бесконтактный ключ с защитой; ФК-16/3 М Контакт реле	16 16	поканальная поканальная
15 Последовательные порты	RS-232C; RS-485; ИРПС (20 mA)		МПСв16	16	Поканальная

Д.3.1 Блоки ввода аналоговых сигналов напряжения и тока среднего уровня

Преобразователь напряжение-код ПНК(И)-30/3 осуществляет преобразование сигналов напряжения постоянного тока в цифровой код, а также хранение полученных результатов и передачу в три КМп по интерфейсу ИР. Значения кодов по контрольным каналам могут быть использованы для контроля работоспособности преобразователя в качестве эталона, относительно которого проверяются выходные коды по контрольному каналу. Характеристики каналов ввода при использовании ПНК(И)-30/3 приведены в таблице Д.2.

Принцип работы. Преобразователь блока ввода аналоговых сигналов (рис. Д.7) содержит:

- коммутаторы К1, К2, подключающие входные сигналы напряжения постоянного тока (К1 - с первого по пятнадцатый канал и сигнал с входа контрольного канала, К2 - с шестнадцатого по тридцатый канал и сигнал с входа контрольного канала) на входы масштабирующих усилителей;

- масштабирующие усилители МУ1, МУ2, служащие для усиления сигналов напряжения постоянного тока;

- источники опорных напряжений ИОН1, ИОН2, формирующие опорные напряжения положительной и отрицательной полярности;

- каналы преобразования КП1-КП4, преобразующие сигналы напряжения в интервал времени;

- генератор тактовых импульсов ГТИ, формирующий непрерывную последовательность тактовых импульсов;

- счетчики каналов СчК1-СчК3, формирующие непрерывную последовательность адресов входных каналов преобразователя;

- счетчики импульсов СчИ1, СчИ2, суммирующие импульсы ГТИ в течение интервалов времени, формируемых КП1-КП4;

- элементы оптоэлектронной развязки ЭОР1 - ЭОР8 для оптоэлектронной развязки цифровых сигналов измерительного узла преобразователя;

- ОЗУ, служащее для хранения результата преобразования по каждому каналу;

- передатчик данных ПД и передатчик адреса ПА, осуществляющие подключение порта ввода-вывода ОЗУ, соответственно, к СчИ1, СчИ2 и СчК3;

- каналы связи с КМп - КСК1-КСК3, реализующие обмена преобразователя с КМп по ИР;

- узел управления УУ, осуществляющий синхронизацию операций чтения данных из ОЗУ с процессом записи результатов преобразования;

- преобразователь напряжения ПН, формирующий напряжения питания +5 V, +15 и минус 15 V и осуществляющий гальваническое разделение цепей питания аналого-цифрового преобразователя.

Каждый из КП1-КП4 содержит (рис. Д.8):

- коммутатор напряжения КН, подключающий источники напряжений к входу интегратора;

- интегрирующий усилитель ИУ для интегрирования выходных сигналов масштабирующих усилителей и опорных напряжений;

- компаратор нулевого уровня КНУ, осуществляющий сравнение выходного сигнала ИУ с нулевым потенциалом;

- триггер знака ТгЗн, фиксирующий знак выходного сигнала ИУ в момент окончания интегрирования сигнала МУ;

- схему сравнения СС для сравнения сигналов КНУ и ТгЗн и определяющую момент перехода выходным сигналом ИУ нулевого уровня;

- триггер конца преобразования ТгКПр, фиксирующий под управлением СС момент окончания интегрирования ИУ опорного напряжения;

- схему управления СУ, осуществляющую выбор опорного напряжения в соответствии с знаком сигнала ИУ.

Таблица Д.2

Характеристики каналов ввода при использовании ПНК(И)-30/3

Наименование показателя и единица измерения	Значение параметра				
Диапазоны изменения входных сигналов:	$\pm 2,5V$	$\pm 5V$	$\pm 10V$	$\pm 5mA$	$\pm 20mA$
Входной ток не более, μA	0,03			-	-
Входное сопротивление не более, Ω	-			499	249
Число подключаемых входных сигналов, шт.	30				
Период преобразования входных сигналов, ms	320±0,32				
Пределы допускаемой приведенной погрешности преобразования, %: -для рабочих условий эксплуатации -для предельных условий эксплуатации	$\pm 0,10$ от U_D $\pm 0,20$ от U_D U_D – верхнее значение напряжения входного сигнала, V			$\pm 0,15$ от I_D $\pm 0,30$ от I_D I_D - верхнее значение тока входного сигнала, mA	
Номинальная функция преобразования	$N_{вых} = U_{вх} \cdot \frac{32000}{U_D}$ где $N_{вых}$ – десятичное значение выходного кода; $U_{вх}, I_{вх}$ – значение входного сигнала, $V; mA$			$N_{вых} = I_{вх} \cdot \frac{32000}{I_D}$	
Формат выходного кода: -знак -информативные разряды	15 0-14				
Коэффициент подавления помех частотой (50±1) Hz не менее, dB : -нормального вида -общего вида	34 80				
Вид гальванической развязки	Групповая				
Испытательное напряжение гальванической развязки, не менее, V	1000				
Время опроса одного входа по ИР не более, μs	10				

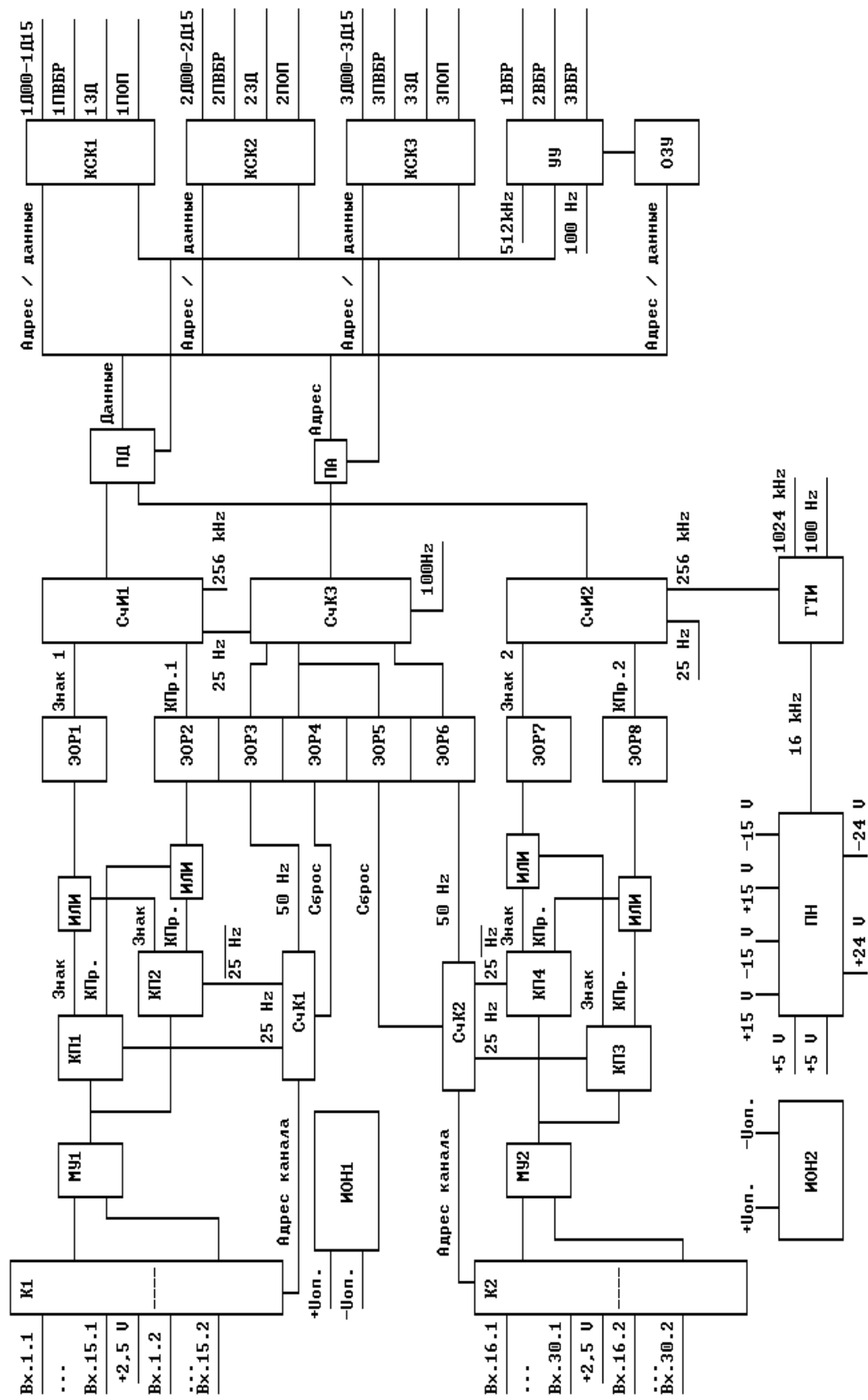


Рис. Д.7. Схема электрическая структурная преобразователя.

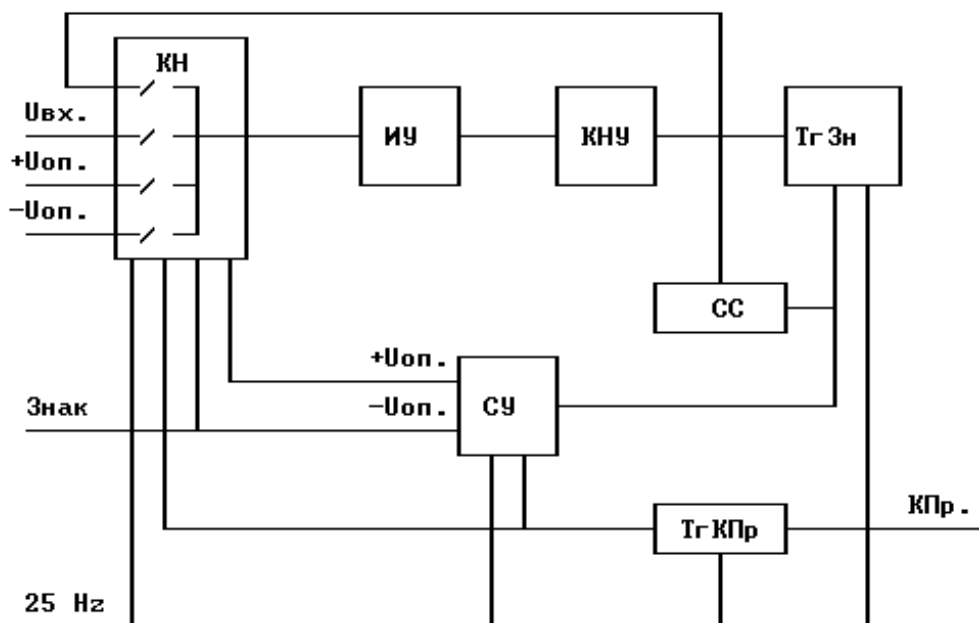


Рис. Д.8. Схема электрическая структурная канала преобразования КП1-КП2.

Четыре канала преобразования КП1-КП4 образуют вместе с СчИ1 и СчИ2 четыре аналого-цифровых преобразователя, работающих по принципу двухтактного интегрирования входного сигнала и опорного напряжения. Время интегрирования входного сигнала равно номинальному значению периода напряжения промышленной сети переменного тока, т.е. 20 ms . При этом обеспечивается 50-кратное подавление помех нормального вида частоты $(50 \pm 1)\text{ Hz}$. Каждый из КП1-КП4 обеспечивает преобразование по восьми входным каналам преобразователя (1-8, 9-15, 16-23 и 24-30) и двум контрольным каналам, на входах которых установлено напряжение $+2,5\text{ V}$ (КП2 и КП4). Фазы интегрирования входных сигналов в КП1 и КП2 сдвинуты относительно друг друга на 20 ms (рис. Д.9) и на 10 ms относительно КП3 и КП4.

Синхронная работа КП1-КП4, К1 и К2 обеспечивается за счет СчК3, формирующего тактовые сигналы с частотой 50 Hz , сдвинутые относительно друг друга на 10 ms , а также импульсом синхронизации, формируемым СчК3 в момент t_1 (рис. Д.10), повторяющимся каждые 320 ms . Данный импульс синхронизации через ЭОР4 и ЭОР5 устанавливает СчК1 и СчК2 в нулевое состояние, соответствующее адресу восьмого, пятнадцатого, двадцать третьего и тридцатого каналов преобразователя. В этот момент к входу МУ1 подключается сигнал восьмого канала преобразователя и начинается его интегрирование в КП1, а в КП3 продолжается интегрирование сигнала по двадцать третьему каналу, заканчивающееся в момент t_3 . Интегрирование сигнала восьмого канала в КП1 длится 20 ms и заканчивается в момент t_4 , в который под действием тактового сигнала 25 Hz в ТгЗн КП1 запоминается знак выходного сигнала ИУ, к входу которого в соответствии с сигналом ТгЗн с помощью СУ подключается опорное напряжение противоположной полярности. В этот же момент к входу МУ1 подключается сигнал пятнадцатого канала преобразователя и начинается его интегрирование в КП2.

Под действием опорного напряжения выходной сигнал ИУ КП1 возвращается к нулевому уровню, момент достижения которого (t_7) фиксируется по изменению уровня выходного сигнала КНУ схемой сравнения СС, устанавливающей ТгКПр в состояние, при котором КН отключает от ИУ опорное напряжение и подключает выход КНУ, обеспечивая таким образом стабилизацию выходного сигнала ИУ в зоне чувствительности КНУ.

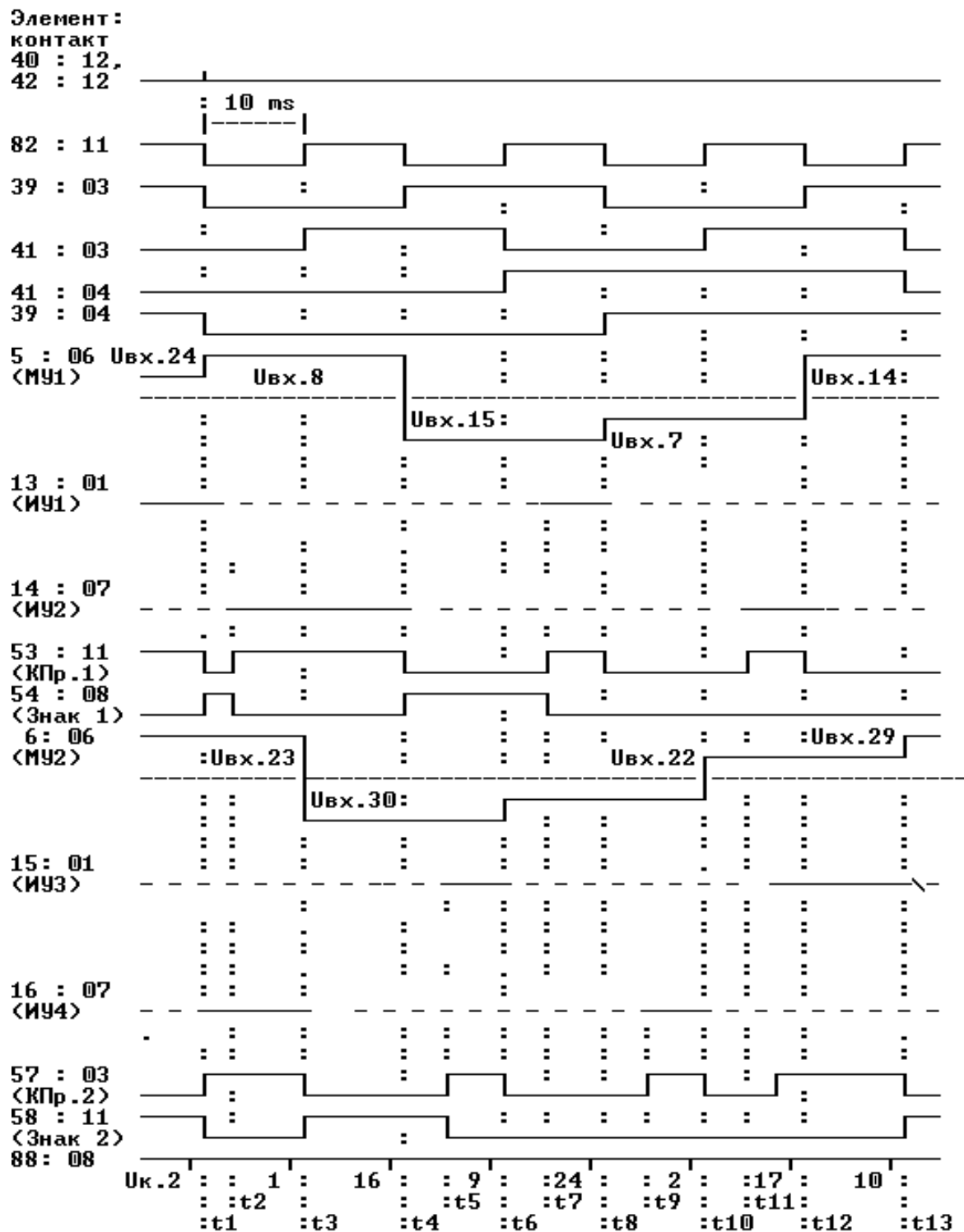


Рис. Д.9. Временная диаграмма сигналов преобразования.

В течение времени обратного хода ИУ КП1 ($t_4 - t_7$) производится суммирование высокочастотных импульсов ГТИ в СЧИ1, в котором формируется число, пропорциональное входному сигналу по восьмому каналу преобразователя. Данное число записывается в ОЗУ по сигналу УУ, формируемому за 1 ms до начала обратного хода интегратора в КП2 (t_8).

Перед началом суммирования в моменты t_4 , t_8 и т. д. СЧИ1 устанавливается в исходное состояние импульсом, формируемым по фронтам сигнала тактовой частоты 25 Hz. В момент t_8 заканчивается интегрирование сигнала по пятнадцатому каналу преобразователя в КП2 и к вхо-

ду МУ1 подключается следующий по порядку в первой группе седьмой канал. Преобразование сигнала по пятнадцатому каналу в двоичный код осуществляется в период $t_8 - t_{12}$ аналогично описанному выше методу. Таким же образом происходит преобразование входных сигналов по каналам 16-30 в КПЗ и КП4, которые работают синхронно с КП1 и КП2 со сдвигом фаз интегрирования на 10 ms .

Преобразователь напряжение-код ПНК(Р)-1. Преобразователь предназначен для преобразования в цифровой код сигналов напряжения постоянного тока или сигналов постоянного тока и ввода их в контроллеры. Он является 16-канальным аналого-цифровым интегрирующим преобразователем, осуществляющим преобразование сигналов напряжения постоянного тока или сигналов постоянного тока в цифровой код, хранение полученных результатов и ввод их в КМп по трем каналам связи с интерфейсом ИР.

Таблица Д.3

Характеристики каналов ввода при использовании ПНК(Р)-1

Наименование показателя и единица измерения	Значение параметра				
Диапазоны изменения входных сигналов	$\pm 25V$	$\pm 5V$	$\pm 10V$	$\pm 5mA$	$\pm 20mA$
Входной ток не более, mA	0,01			-	-
Входное сопротивление, Ω , не более	-			499	249
Число подключаемых входных сигналов, шт.	16				
Период преобразования входных сигналов, ms	30±0,03				
Пределы допускаемой приведенной погрешности преобразования, %: -в рабочих условиях эксплуатации -в предельных условиях эксплуатации	$\pm 0,10$ от U_D $\pm 0,20$ от U_D U_D – верхнее значение напряжения входного сигнала, V			$\pm 0,15$ от I_D $\pm 0,30$ от I_D I_D - верхнее значение тока входного сигнала, mA	
Номинальная функция преобразования	$N_{\text{вых}} = U_{\text{вх}} \cdot \frac{32000}{U_D}$ где $N_{\text{вых}}$ – десятичное значение выходного кода $U_{\text{вх}}$, $I_{\text{вх}}$ – значение входного сигнала, V ; mA			$N_{\text{вых}} = I_{\text{вх}} \cdot \frac{32000}{I_D}$	
Формат выходного кода: -знак -информативные разряды	15 0-14				
Коэффициент подавления помех частотой (50±1) Hz , не менее, dB : -нормального вида -общего вида	80 120				
Вид гальванической развязки	Покаканальная				
Испытательное напряжение гальванической развязки, не менее, V	1000				
Время опроса одного входа по ИР, μs , не более	10				

Преобразователь по интерфейсу ИР осуществляет: прием адреса входного канала; вывод на шины данных результата преобразования по адресованному каналу; вывод на шины данных кода по контрольному каналу. Характеристики каналов ввода при использовании ПНК(Р)-1 приведены в табл. Д.3.



285

Шестнадцатиканальный коммутатор КН управляется МК. Он обеспечивает подключение к входу АЦП выходных сигналов U1-U8, или контрольного напряжения, или датчика температуры ДТ. АЦП преобразует сигналы напряжения постоянного тока в цифровой код и передает результаты преобразования в МК.

Микроконтроллер МК выполняет функции интегрирования и масштабирования по каждому каналу цифровых значений входного сигнала и записывает результаты преобразования в ОЗУ. Источник опорного напряжения ИОН формирует опорное напряжение +2,5 V для работы АЦП и датчика температуры ДТ. Схема управления СУ формирует тактовые сигналы запуска АЦП и работы ПН. СУ управляет чтением и записью данных в ОЗУ, устанавливая очередность обслуживания запросов доступа к ОЗУ от БУ1-БУ3 и МК. ОЗУ служит для хранения результатов преобразования по каждому каналу. Преобразователь напряжения ПН формирует переменное напряжение для питания гальванически развязанных узлов U1-U8, а также постоянные напряжения +15 и минус 15 V. ПН работает по принципу мостового модулятора постоянного напряжения.

Датчик температуры ДТ служит для коррекции результатов преобразования входных сигналов при изменении температуры. Датчик температуры представляет собой микросхему с чувствительностью 10 mV/°C. При изменении температуры выходное напряжение микросхемы увеличивается пропорционально изменению температуры. В процессе работы МК периодически вводит значение температуры для коррекции данных, поступающих по входным каналам.

Буферные усилители БУ1-БУ3 осуществляют обмен преобразователя с ИР.

Д.3.2 БЛОКИ ВВОДА АНАЛОГОВЫХ СИГНАЛОВ ТОКА НИЗКОГО УРОВНЯ

Преобразователь каналов измерительных ПКИ-2/3 предназначен для преобразования токовых сигналов от 7 датчиков прямого заряда (ДПЗ) в цифровые коды и передачи их в КМп, а также для контроля проводимости сопротивления изоляции ДПЗ от корпуса реактора.

Для каждого датчика имеется четыре измерительных канала: канал токового сигнала от 0 до 5 μA ; канал фоновой составляющей токового сигнала от 0 до 5 μA ; канал измерения проводимости изоляции ДПЗ; канал измерения проводимости изоляции по каналу фоновой составляющей сигнала. Кроме того, преобразователь осуществляет формирование поправочных коэффициентов (коэффициентов передачи входных усилителей по каналам ДПЗ) Кн1 - Кн7 для программной коррекции результатов преобразования сигналов ДПЗ.

Характеристики каналов ввода при использовании ПКИ-2/3. Преобразование в ПКИ-2 имеет циклический характер, запуск очередного цикла преобразования осуществляется программным способом. В каждом цикле осуществляется преобразование основных сигналов (0-5) μA . Все остальные параметры являются дополнительными (ДП) и преобразовываются один раз в 40 циклов.

Входные усилители преобразователя не имеют индивидуальной подстройки передаточной характеристики, что ограничивает точность результатов преобразования сигналов ДПЗ. Более точное (действительное) значение сигнала ДПЗ может быть получено согласно формуле:

$$N_{нд} = N_n \times \frac{16000}{K_n},$$

где $N_{нд}$ - действительное значение сигнала ДПЗ; 16000 - верхний предел изменения выходного кода по каналам ДПЗ; K_n - коэффициент передачи входных усилителей; N_n - результат преобразования.

Характеристики каналов ввода при использовании ПКИ-2/3 приведены в табл.Д.4.

Характеристики каналов ввода при использовании ПКИ-2/3

Наименование показателя и единица измерения	Значение параметра		
Диапазоны изменения входных сигналов: -тока, μA -проводимости, $1/\Omega$	От 0 до 5 -	$\pm 0,5$ -	- От 0 до $5 \cdot 10^{-6}$
Число подключаемых входных сигналов (шт.)	7	7	14
Период преобразования входных сигналов, s	0,6 $\pm$ 0,0006	19,2 $\pm$ 0,02	
Пределы допускаемой приведенной погрешности преобразования, % от верхнего значения сигнала: -в рабочих условиях эксплуатации -в предельных условиях эксплуатации	$\pm 0,05$ от I_D $\pm 0,1$ от I_D	$\pm 0,20$ от I_D $\pm 0,40$ от I_D	$\pm 2,0$ от G_D $\pm 4,0$ от G_D
Номинальная функция преобразования	$N_{\text{вых}} = I_{\text{вх}} \cdot \frac{16000}{I_D}$$N_{\text{вых}} = G_{\text{вх}} \cdot \frac{1000}{5 \cdot 10^{-6}}$ где $N_{\text{вых}}$ – десятичное значение выходного кода; $I_{\text{вх}}, G_{\text{вх}}$ – значение входного сигнала, μA; $1/\Omega$; I_D, G_D – верхнее значение входного сигнала, μA; $1/\Omega$		
Формат выходного кода: -знак -информативные разряды	15 0-14	15 0-14	
Коэффициент подавления помех частотой (50 $\pm$ 1) Hz, не менее, dB: -нормального вида -общего вида	60 100		
Вид гальванической развязки	Групповая		
Испытательное напряжение гальванической развязки, не менее, V	1500		
Время опроса одного канала по ИР не более, μs	10		

Формирование поправочных коэффициентов $K_{H1} - K_{H7}$ производится в преобразователе на фоне сигналов ДПЗ. Преобразователь выполняет следующие функции: прием команды запуска; прием адреса входного канала; прием команды блокировки измерений проводимости изоляции, фонового сигнала (ФС) и корректирующих коэффициентов; прием команды на включение сигнализации о неисправности; вывод на шины данных ИР по адресованному каналу результата преобразования или поправочного коэффициента в виде 16-разрядного двоичного кода.

Принцип работы. Преобразователь содержит (рис. Д.11):

- усилители U1-U7 для преобразования токовых сигналов ДПЗ и фонового сигнала в напряжение;
- коммутатор К, подключающий выходы U1-U7 к масштабирующему усилителю;
- масштабирующий усилитель МУ для усиления сигналов U1-U7 по каналам ФС;
- преобразователь «напряжение - временной интервал» (ПНВИ), работающий по принципу двойного интегрирования и преобразующий входной сигнал во временной интервал;

- счетчики каналов СчК1, СчК2, формирующие синхронные последовательности адресов измерительных каналов;
- коммутаторы адреса КА1, КА2, коммутирующие разряды СчК1 и СчК2 под управлением Дш и УУ;
- источник опорных сигналов ИОС, формирующий стабилизированные значения опорных сигналов, необходимых для работы преобразователя;
- коммутатор опорного тока КОТ, подключающий калиброванный источник тока к входам токовых усилителей при контроле их коэффициента передачи;
- шифратор Дш, определяющий по старшим разрядам СчК1 выбор типа измерительного канала;
- счетчик импульсов СчИ, служащий для преобразования временного интервала, формируемого ПНВИ, в эквивалентный цифровой двоичный код;
- оперативное запоминающее устройство ОЗУ, предназначенное для хранения результатов преобразования;
- передатчик данных ПД, служащий для подключения выходов СчИ к магистрали обмена данными с ОЗУ;
- каналы связи с ИР - КСК1-КСК3;
- триггер запрета ТгЗ, предназначенный для приема от КМп признака запрета измерения дополнительных параметров;
- узел синхронизации УС, служащий для синхронизации сигнала запрета с циклом измерения;
- генератор тактовых импульсов ГТИ, формирующий непрерывную последовательность тактовых сигналов;
- делитель частоты ДЧ, формирующий тактовые сигналы низкой частоты для управления СчК1, СчК2 и ПНВИ;
- триггер запуска ТгЗп, служащий для запоминания сигнала запуска преобразователя и сбрасываемый СчК2 в конце 400-миллисекундного цикла преобразования сигналов ДПЗ;
- триггер ошибки ТгОш, предназначенный для приема от любого из трех КМп сигнала ошибки, обнаруженной в работе преобразователя, и управляющий режимом свечения индикатора, расположенного на его передней панели;
- узел управления УУ, реализующий алгоритм обмена данными ОЗУ с СчИ и КМп;
- преобразователь напряжения ПН, формирующий изолированные напряжения питания +12 и минус 12 V, +16 и минус 16 V, а также напряжение питания +5 V;
- элементы оптоэлектронной развязки ЭРО1-ЭРО5, обеспечивающие гальваническую развязку входов преобразователя.

ПНВИ, осуществляющий двойное интегрирование выходных сигналов $U_1 - U_7$, содержит (рис. Д.12): интегрирующий усилитель (ИУ), выполняющий функцию интегрирования; компаратор нулевого уровня (КНУ), фиксирующий момент перехода выходного сигнала ИУ через нулевой уровень; схему сравнения (СС), формирующую сигнал конца преобразования; триггер конца преобразования (ТгКПр), служащий для фиксации сигнала конца преобразования; триггер знака (ТгЗн), запоминающий знак выходного сигнала ИУ в конце прямого интегрирования преобразуемого сигнала; коммутатор напряжения (КН), подключающий под управлением тактового сигнала 20 Hz сигналов ТгКПр и СВП к входу ИУ преобразуемый сигнал, опорное напряжение, или выход КНУ; схему выбора полярности (СВП), определяющую полярность опорного напряжения, подключаемого к входу ИУ во втором такте интегрирования преобразуемого сигнала; формирователь весовой функции (ФВФ) и переключатель коэффициента весовой функции (ПКВФ), реализующие двухступенчатую весовую функцию интегрирования, обеспечивающую достаточную степень подавления помех нормального вида без подстройки времени интегрирования под частоту сети электроснабжения.

Принцип преобразования входных сигналов в цифровой код поясняется временной диаграммой, приведенной на рис. Д.13.

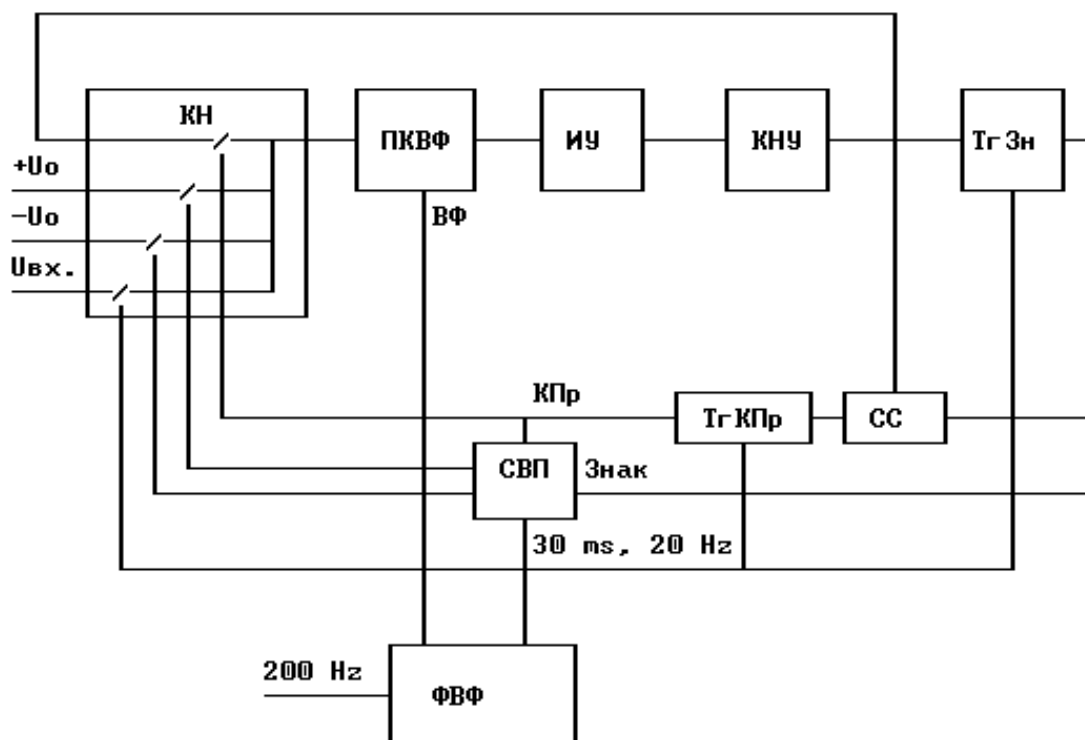


Рис. Д.12. Схема электрическая структурная ПНВИ.

До момента поступления от любого из трех КМп команды запуска преобразователь находится в режиме ожидания, в котором три младших разряда СчК2 находятся в состоянии сигнала высокого уровня (соответствующего выбору дополнительного параметра - ДП), а три младших разряда СчК1 - в состоянии сигнала низкого уровня (соответствующего выбору первого канала ДПЗ). В СчИ хранится результат преобразования ДП, а ИУ (в ПНВИ) находится в режиме стабилизации нуля, при котором к его входу подключен выход КНУ. Данный режим длится с момента поступления команды запуска (t_1) еще 20 ms (до t_2). В период $t_2 - t_5$ на выходе ФВФ (в ПНВИ) формируется 30-миллисекундный интервал, в течение которого к входу ИУ подключается сигнал первого канала ДПЗ. В момент t_5 на выходе ИУ формируется напряжение, пропорциональное среднему значению измеряемого параметра, знак которого запоминается в ТгЗн. В этот же момент происходит переключение СчК1, три младших разряда которого через КА1 подключают к входу ПНВИ сигнал следующего канала ДПЗ. В момент t_3 по сигналу ДЧ УУ производит запись результата преобразования ДП в ОЗУ из СчИ и в момент t_4 сбрасывает последний в нулевое состояние, подготавливая его к приему высокочастотных импульсов (1024 kHz). Адрес измерительного канала ДП формируется пятью старшими разрядами СчК2, из которых три разряда (с четвертого по шестой) определяют физический номер ДП, а два разряда (седьмой, восьмой) - тип параметра.

Переключение СчК2 происходит по сигналу ДЧ в момент t_5 , при этом три младших разряда (с первого по третий) переходят в нулевое состояние, определяющее адрес первого канала ДПЗ. В момент t_5 к входу ИУ ПНВИ (рис. Д.13) подключается опорное напряжение, противоположное по знаку выходному сигналу U_1 . Под действием опорного напряжения выходной сигнал ИУ возвращается к нулю, момент достижения которого (t_6) фиксирует КНУ. По изменению выходного сигнала КНУ СС формирует импульс, устанавливающий ТгКПр в состояние, в соответствии с которым СВП отключает опорное напряжение от ИУ. Выход ТгКПр, непосредственно управляющий одним из ключей КН, подключает к входу ИУ выход КНУ, который таким образом включается в цепь отрицательной обратной связи ИУ по постоянному току и стабилизирует его выходной сигнал до момента начала следующего цикла интегрирования. В те-

чение времени $t_5 - t_6$ производится заполнение СчИ импульсами высокой частоты, количество которых будет пропорционально измеряемому сигналу. Результат преобразования сигнала по первому каналу ДПЗ переписывается из СчИ в ОЗУ в момент t_8 по сигналу ДЧ. Аналогичным образом происходит преобразование по всем каналам ДПЗ, заканчивающееся в момент t_{10} . В момент t_9 три младших разряда СчК1 переключаются в состояние сигналов высокого уровня и устанавливается режим измерения ДП, адрес которого сформирован к тому времени в пяти старших разрядах СчК1.

При контроле коэффициента передачи $U_1 - U_7$ в период $t_9 - t_{12}$ на вход выбранного усилителя через коммутатор опорного тока КОТ подается калиброванный ток минус 5 мА. Следовательно, выходной сигнал усилителя будет определяться разностью тока ДПЗ и тока опорного, поэтому после записи в момент t_{11} значения тока ДПЗ по седьмому каналу в СчИ заносится из ОЗУ значение тока ДПЗ по контролируемому каналу. Таким образом, в момент окончания обратного хода ИУ (t_{13}) содержимое СчИ будет пропорционально коэффициенту передачи контролируемого усилителя, численное значение которого должно находиться в пределах 16000 ± 64 .

При контроле изоляции ДПЗ и ФС ДПЗ на опорные входы $U_1 - U_7$ подается напряжение контроля изоляции +0,0625 или минус 0,0625 В, вызывающее из-за отрицательных обратных связей в усилителях тока, аналогичное изменение потенциалов их входов относительно общего провода. Выходной сигнал усилителя изменяется при этом на величину, пропорциональную проводимости изоляции ДПЗ, и константу, определяемую напряжением контроля изоляции $U_{ки}$. Учет значения сигнала самого ДПЗ производится в СчИ так же, как и при контроле коэффициента передачи. При измерении ФС Дш подключает к входу ПНВИ выход усилителя U по фоновому сигналу контролируемого ДПЗ. В момент t_{12} ДЧ по конъюнкции младших разрядов (с первого по третий) СчК2 сбрасывает ТгЗп, который, в свою очередь, фиксирует состояние ДЧ до момента поступления от КМп очередной команды. Время выполнения данного цикла - 400 мс. Достоверная информация поступает на выход преобразователя через шестьдесят четыре цикла преобразования (25,6 с), необходимых для синхронизации работы СчК1 и СчК2 (12,8 с) и преобразования всех параметров.

Выдача информации на ИР производится через КСК1-КСК3 по запросу КМп. Синхронизация цикла обмена данными между контроллерами и преобразователем производится УУ, тактируемым частотой 1024 кГц.

При шумовой диагностике преобразователь переводится командой от любого из трех КМп в стационарный режим, при котором измерение ДП не производится. Блокировка измерения ДП осуществляется ТгЗ. УС служит для синхронизации команды запрета с циклом измерения и исключает искажение информации в ОЗУ.

Принцип интегрирования с двухступенчатой весовой функцией поясняется рис. Д.14. На участке $t_1 - t_2$ оба ключа ПКВФ замкнуты и интегрирование производится с удвоенной скоростью, что эквивалентно суммарному сигналу двух одинаковых ИУ, интегрирующих измеряемый сигнал в интервале 20 мс со сдвигом в 0,5 периода (10 мс) частоты помехи. Таким образом, действие помехи в двух интеграторах оказывается противоположным, что приводит к увеличению степени ее подавления.

Код ДП формируется двумя старшими разрядами СчК1 в следующем соответствии (первый знак - старший разряд): 00 - измерение коэффициента передачи усилителя U ; 01 - измерение проводимости изоляции канала ФС ДПЗ; 10 - измерение проводимости изоляции канала ДПЗ; 11 - измерение ФС ДПЗ.

Преобразователь сигналов термопреобразователей сопротивления ПТС(В)-16/3. Преобразователь является 16-канальным аналого-цифровым интегрирующим преобразователем, осуществляющим преобразование сигналов термопреобразователей сопротивления в цифровой код, хранение полученных результатов и передачу их в КМп по интерфейсу ИР, а также контроль линий связи с датчиками. Результат преобразования выводится на шины данных ИР 16-разрядным двоичным дополнительным кодом.

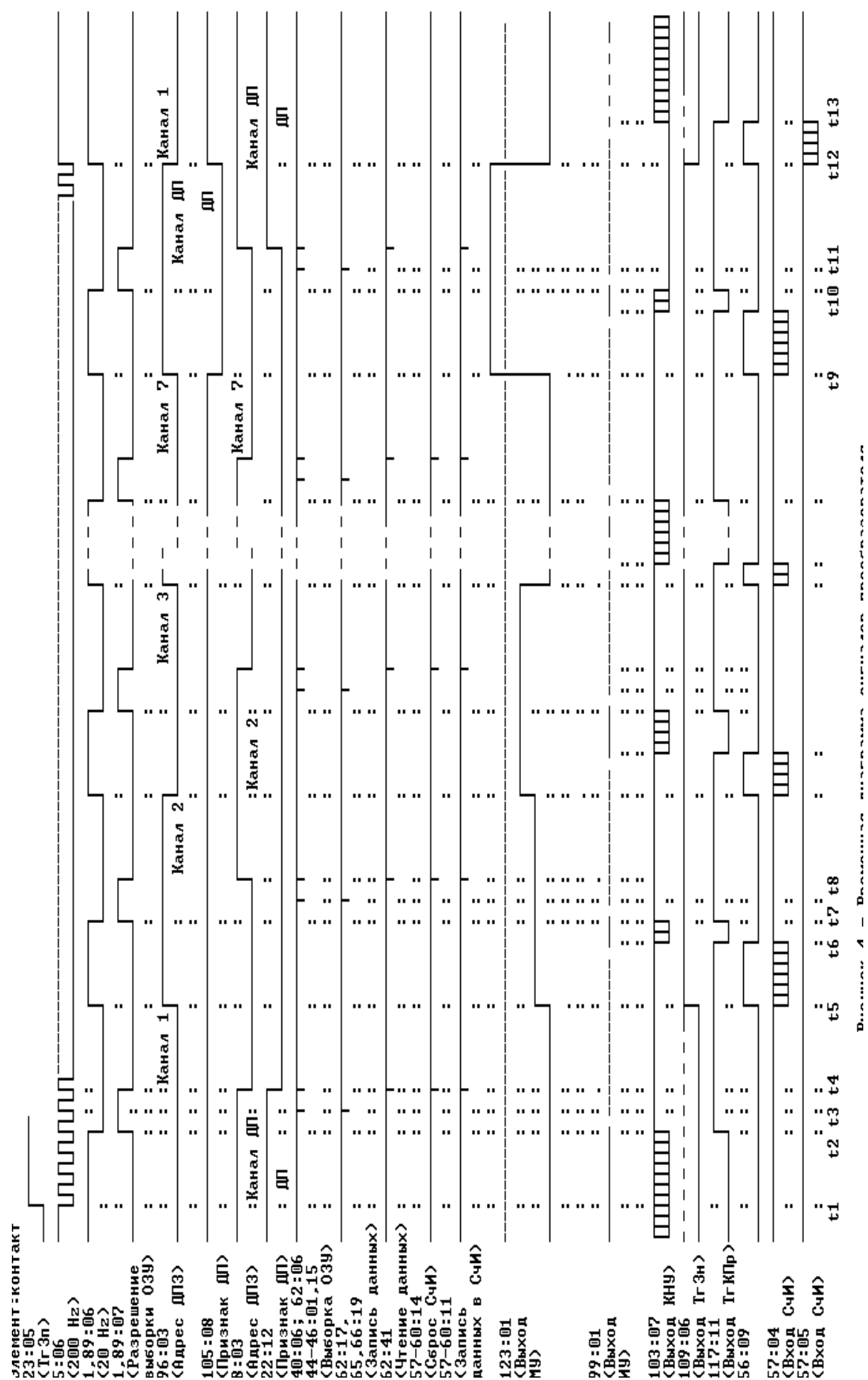


Рис. Д.13. Временная диаграмма сигналов ПКИ-2/3

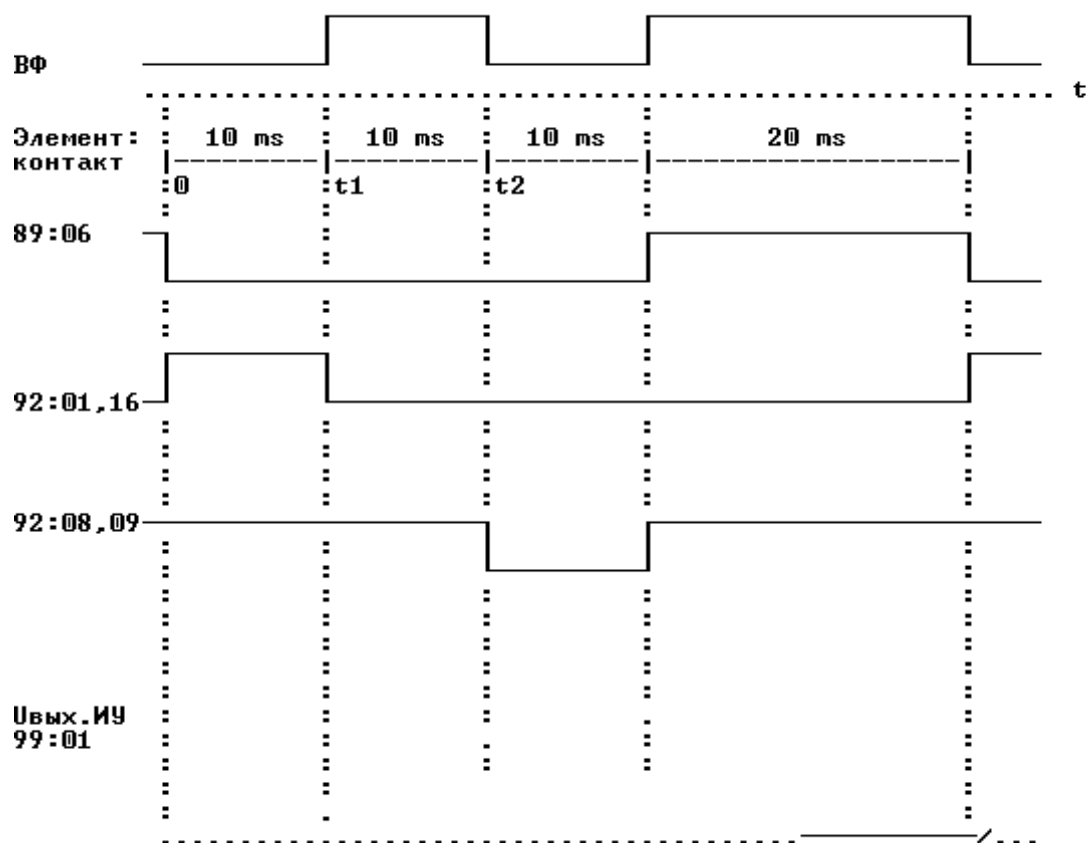


Рис. Д.14. Временная диаграмма сигналов ПНВИ.

Д.3.3 Блоки ввода сигналов термопреобразователей сопротивления

Преобразователь сигналов термопреобразователей сопротивления ПТС(В)-16/3 является 16-канальным аналого-цифровым интегрирующим преобразователем, осуществляющим преобразование сигналов термопреобразователей сопротивления в цифровой код, хранение полученных результатов и передачу их в КМп по интерфейсу ИР, а также контроль линий связи с датчиками. Результат преобразования выводится на шины данных ИР 16-разрядным двоичным дополнительным кодом. Характеристики каналов ввода с обеспечением взрывозащиты вида “искробезопасная электрическая цепь” при использовании ПТС(В)-16 приведены в табл. Д.5.

Принцип работы. Преобразователь содержит (рис. Д.15):

- коммутатор входной Квх., подключающий термопреобразователи сопротивления ТС к входам масштабирующего усилителя, усилителя компенсации и источнику тока питания;
- масштабирующий усилитель МУ, осуществляющий усиление напряжения, снимаемого с ТС;
- усилитель компенсации УК, служащий для компенсации начального сопротивления ТС и падения напряжения на линии связи с ТС, вызванного током питания ТС;
- источник опорных токов ИОТ, формирующий двухполярный ток питания ТС;
- источник компенсирующего напряжения ИКН, компенсирующий начальное сопротивление ТС;
- коммутатор опорных токов КОТ, переключающий направление тока питания ТС;
- источник опорных токов ИОН, формирующий опорные напряжения положительной и отрицательной полярности;

Таблица Д.7

Характеристики каналов ввода с обеспечением взрывозащиты вида “искробезопасная электрическая цепь” при использовании ПТС(В)-16.

Наименование показателя и единица измерения	Значение параметра	
Диапазоны изменения входных сигналов, Ω	0-50-100; 0-50-150; 0-50-250	0-100-200; 0-100-300; 0-100-500
Число подключаемых входных сигналов, шт.	16	
Период преобразования входных сигналов, s,	2,56±0,0026	
Пределы допускаемой приведенной погрешности преобразования, %: -в рабочих условиях эксплуатации -в предельных условиях эксплуатации	±0,25 от R_D ±0,5 от R_D , где $R_D = R_K - R_0$, R_K – максимальное сопротивление, Ω ; R_0 – сопротивление датчика при 0 °C, Ω (50 Ω , 100 Ω)	
Номинальная функция преобразования	$N_{вых} = \frac{R_{ex} - R_0}{R_D} \cdot 32000$, где $N_{вых}$ – десятичное значение выходного кода, R_{ex} – значение входного сигнала, Ω	
Формат выходного кода: -знак -информативные разряды	15 0-14	
Коэффициент подавления помех с частотой (50±1) Hz, dB, не менее: -нормального вида -общего вида	40 100	
Вид гальванической развязки	Групповая	
Испытательное напряжение гальванической развязки, не менее, V	1500	
Время опроса одного канала по ИР не более, μs	10	
Вид взрывозащищенности по ГОСТ 22782	Exia 11C	

- коммутатор напряжения КН, подключающий под управлением тактового сигнала и СУ к входу ИУ преобразуемый сигнал, или опорное напряжение, или выход КНУ;
 - интегрирующий усилитель ИУ, выполняющий функцию интегрирования усиленного МУ сигнала ТС и подавления помех нормального вида;
 - компаратор нулевого уровня КНУ, фиксирующий момент перехода выходного сигнала ИУ через нулевой уровень;
 - генератор тактовых импульсов ГТИ, формирующий непрерывную последовательность высокочастотных тактовых импульсов;
 - счетчики каналов СчК1, СчК2, формирующие синхронные последовательности адресов каналов преобразователя;
 - счетчик импульсов СчИ, преобразующий число-импульсные сигналы в цифровой код;
 - схему управления СУ, реализующую алгоритм аналого-цифрового преобразования;
 - ОЗУ, предназначенное для хранения результатов преобразования;
 - передатчик данных ПД для передачи данных из СчИ в ОЗУ;
 - передатчик адреса ПА, для передачи адреса канала из СчК2 в ОЗУ;
- каналы связи с контроллером КСК1-КСК3, образующие три канала связи с контроллерами;
- элементы оптоэлектронной развязки ЭРО1-ЭРО4, служащие для гальванической развязки цепей управления и выходного сигнала преобразователя;

- узел управления УУ, осуществляющий обмен преобразователя с ИР и управление ОЗУ;
- преобразователь напряжения ПН, осуществляющий гальваническое разделение цепей питания аналого-цифрового преобразователя.

Принцип преобразования входных сигналов в цифровой код поясняется временной диаграммой, приведенной на рисунке Д.16 (Увх. - входной сигнал напряжения; Увых. - выходной сигнал напряжения; U_0 – напряжение компенсации начального сопротивления ТС; $U_{оп.}$ - опорное напряжение).

В момент t_1 СчК1 формирует адрес очередного канала преобразователя, по которому Квх. подключает ТС к МУ, УК и через КОТ к источнику питания положительной полярности. На выходе МУ устанавливается напряжение, пропорциональное разности действительной и начальной величины сопротивления ТС.

Через время $t_1 - t_2$ (20 ms), необходимое для установления сигнала на выходе МУ, к выходу последнего КН подключает вход ИУ, который осуществляет интегрирование сигнала ТС в течение времени $t_2 - t_4$, равного периоду промышленной частоты 50 Hz (20 ms).

В момент начала интегрирования t_2 СУ формирует сигнал записи результата преобразования по предыдущему каналу, по которому УУ в течение времени интегрирования сигнала по текущему каналу переписывает содержимое СчИ в ОЗУ, устанавливает СчИ в исходное состояние и увеличивает содержимое СчК2 на 1 (t_3).

В момент окончания интегрирования t_4 в СУ запоминается знак выходного сигнала ИУ, который определяет знак опорного напряжения, подключаемого в данный момент к ИУ. Одновременно СУ начинает заполнение СчИ импульсами ГТИ и изменяет с помощью КОТ направление тока питания ТС.

Под воздействием опорного напряжения выходной сигнал ИУ линейно изменяется в сторону уменьшения до момента достижения нулевого уровня t_5 , фиксируемого КНУ, изменение выходного сигнала которого запрещает поступление импульсов ГТИ на вход СчИ.

Так как выходной сигнал интегратора пропорционален среднему значению сигнала ТС за период интегрирования, который постоянен, то и время $t_4 - t_5$, а, следовательно, и количество импульсов в СчИ также пропорционально среднему значению сигнала ТС.

В момент t_5 КН под управлением СУ подключает вход ИУ к выходу КНУ, благодаря чему обеспечивается стабилизация нулевого уровня выходного сигнала ИУ до начала следующего этапа интегрирования.

С целью увеличения подавления помех и компенсации напряжения смещения МУ и УК производится повторное преобразование $t_4 - t_8$ сигнала ТС с изменением направления тока питания последнего и в СчИ производится суммирование результатов обоих преобразований.

Результат преобразования по данному каналу записывается в ОЗУ в момент t_9 .

Аналогичным образом происходит преобразование сигналов ТС по всем каналам преобразователя.

После завершения цикла преобразований сигналов ТС по всем шестнадцати каналам КОТ под управлением СУ отключает питание ТС и по всем каналам осуществляется преобразование напряжения компенсации начального сопротивления ТС, передаваемого с выхода УК на вход МУ через линию связи преобразователя с ТС. Таким образом, осуществляется контроль и преобразователя, и линии связи.

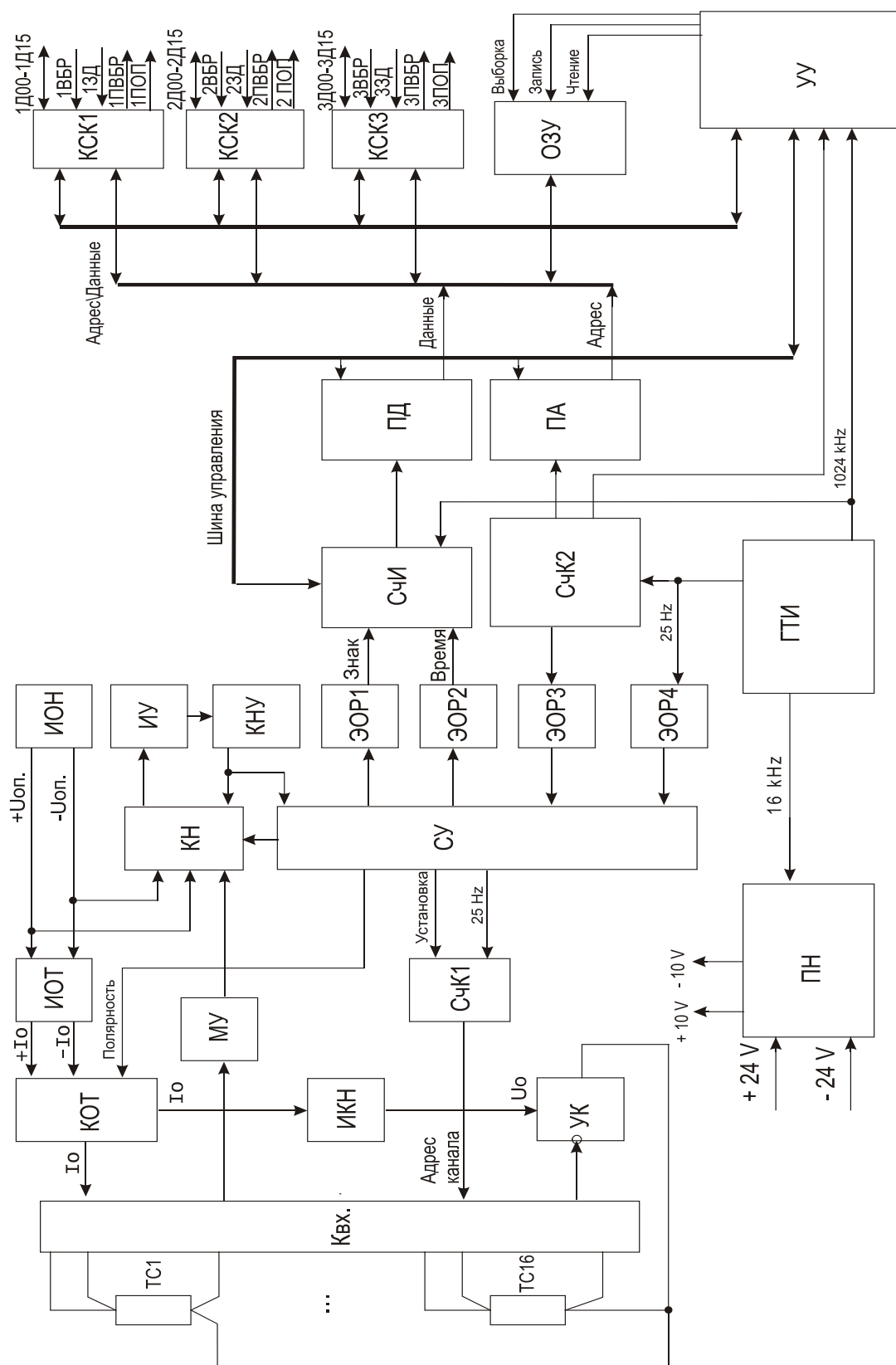


Рис. Д.15. Структурная электрическая схема преобразователя ПТС(В).

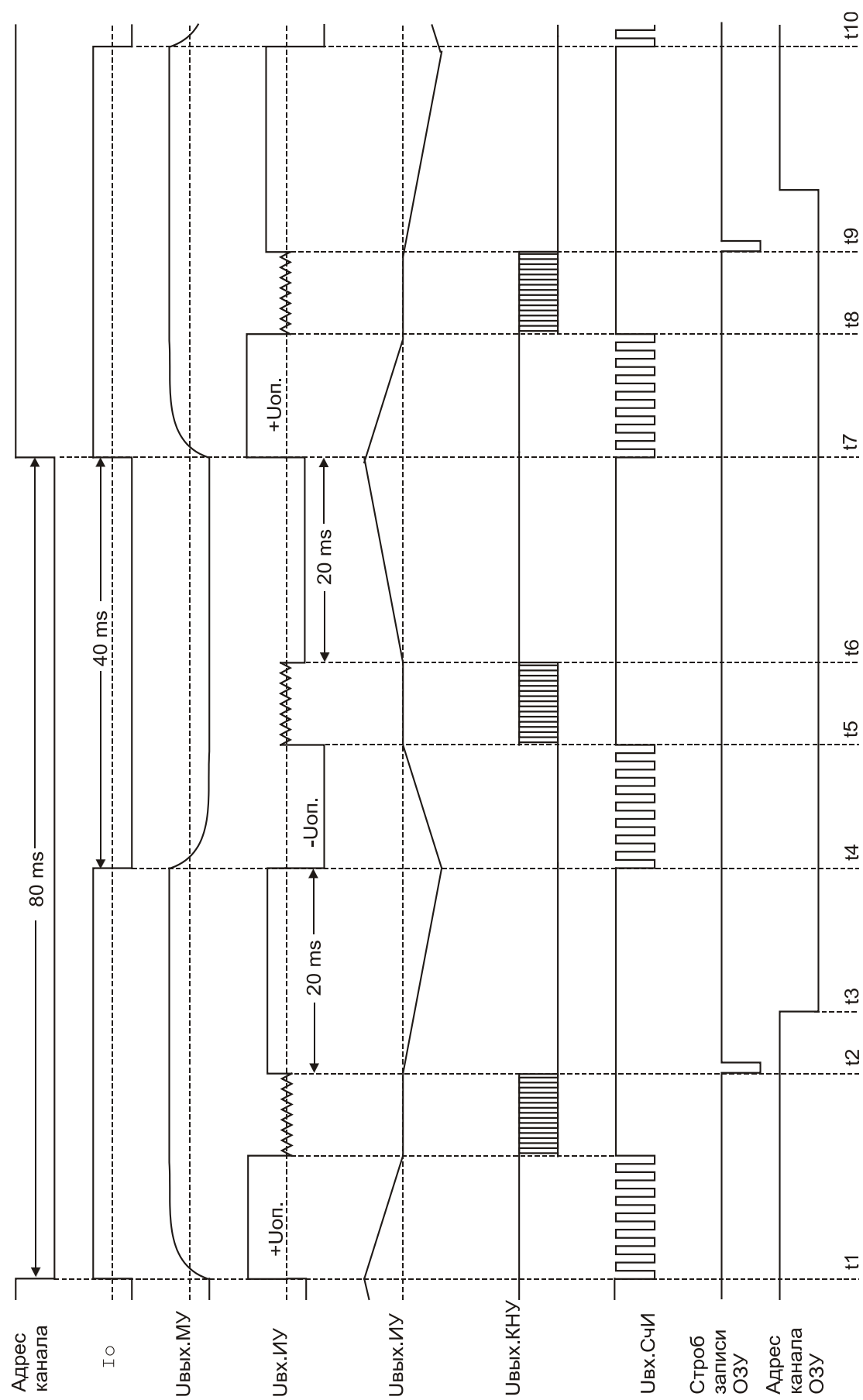


Рис. Д.16 - Временная диаграмма сигналов преобразователя

Преобразователь с входными искробезопасными электрическими цепями уровня "ia" имеет маркировку взрывозащиты "Ex ia". Вид взрывозащиты "Ex ia" основан на принципе ограничения предельной энергии, накапливаемой или выделяемой электрической цепью в аварийном режиме, или рассеивания мощности до уровня значительно ниже минимальной энергии или температуры воспламенения. Допустимые уровни энергии в искробезопасной электрической цепи находятся в пределах от 20 до 180 μJ (напряжение разомкнутой электрической цепи не более 30 V, значение тока короткого замыкания не более 100 mA, максимально допустимая мощность 0.45 W). Вид взрывозащиты "Ex ia" предполагает, что электрическая цепь может эксплуатироваться в помещениях, в которых выделяются горючие газы или пары легковоспламеняющихся жидкостей, которые могут образовывать с воздухом взрывоопасные смеси при нормальных режимах работы (например, на территории газзавода, на нефтеперерабатывающем заводе, при хранении или переливании легковоспламеняющихся жидкостей и т.п.). Искробезопасность входных цепей преобразователя достигается за счет ограничения тока в его входных цепях до искробезопасного значения, гальваническим разделением входных цепей от цепи первичного питания и конструктивными мерами. Преобразователь устанавливается в каркас МСКУ, располагаемого вне взрывоопасных зон помещений и наружных установок. К преобразователю могут подключаться серийно выпускаемые приборы общего назначения, устанавливаемые во взрывоопасных зонах помещений и наружных установок.

Д.3.4 Блоки ввода сигналов термоэлектрических преобразователей

Преобразователь сигналов термоэлектрических преобразователей ПТП(В)-16/3. Преобразователь предназначен для приема, преобразования в цифровой код сигналов термоэлектрических преобразователей. Характеристики каналов ввода с обеспечением взрывозащиты вида "искробезопасная электрическая цепь" при использовании ПТП(В)-16/3.

Преобразователь является 16-канальным аналого-цифровым интегрирующим преобразователем, осуществляющим преобразование сигналов напряжения термоэлектрических преобразователей в цифровой код, а также хранение полученных результатов и передачу в КМп по трем каналам связи интерфейса ИР.

Принцип работы. Преобразователь по интерфейсу ИР осуществляет: прием адреса входного канала; вывод на шины данных результата преобразования по адресованному каналу; вывод на шины данных кода по контрольному каналу. Преобразователь содержит (рис. Д.17):

- коммутатор входной Квх., подключающий выходные сигналы ТП к входу масштабирующего усилителя;
- масштабирующий усилитель МУ, осуществляющий усиление напряжения, снимаемого с ТП;
- источник опорных напряжений ИОН, формирующий опорные напряжения положительной и отрицательной полярности;
- коммутатор напряжения КН, подключающий под управлением тактового сигнала и СУ к входу ИУ преобразуемый сигнал, или опорное напряжение, или выход КНУ;
- интегрирующий усилитель ИУ, выполняющий функцию интегрирования усиленного МУ сигнала ТП и подавления помех нормального вида;
- компаратор нулевого уровня КНУ, фиксирующий момент перехода выходного сигнала ИУ через нулевой уровень;
- генератор тактовых импульсов ГТИ, формирующий непрерывную последовательность высокочастотных тактовых импульсов;
- счетчики каналов СчК1, СчК2, формирующие синхронные последовательности адресов каналов преобразователя;
- счетчик импульсов СчИ, преобразующий число-импульсные сигналы в цифровой код;
- схему управления СУ, реализующую алгоритм аналого-цифрового преобразования;
- ОЗУ, предназначенное для хранения результатов преобразования;
- передатчик данных ПД для передачи данных из СчИ в ОЗУ;

- передатчик адреса ПА для передачи адреса канала из СчК2 в ОЗУ;
- каналы связи с контроллером КСК1-КСК3, образующие три канала связи с контроллерами;
- элементы оптоэлектронной развязки ЭРО1-ЭРО4, служащие для гальванической развязки цепей управления и выходного сигнала преобразователя;
- узел управления УУ, осуществляющий обмен преобразователя с ИР и управление ОЗУ;
- преобразователь напряжения ПН, осуществляющий гальваническое разделение цепей питания аналого-цифрового преобразователя.

Таблица Д8

**Характеристики каналов ввода с обеспечением взрывозащиты вида
«искробезопасная электрическая цепь при использовании ПТП(В)-16/3.**

Наименование показателя и единица измерения	Значение параметра						
Диапазоны изменения входных сигналов, mV	± 10	± 20	± 30	± 40	± 50	± 80	± 100
Число подключаемых входных сигналов, шт.	16						
Период преобразования входных сигналов, s ,	$1,7 \pm 0,0017$						
Пределы допускаемой приведенной погрешности преобразования, %: -в рабочих условиях эксплуатации -в предельных условиях эксплуатации	$\pm 0,25$ от U_D $\pm 0,5$ от U_D U_D – верхнее значение диапазона, mV						
Номинальная функция преобразования	$N_{вых} = \frac{U_{вх}}{U_D} \cdot 32000$, где $N_{вых}$ – десятичное значение выходного кода $U_{вх}$ – значение входного сигнала, mV						
Формат выходного кода: -знак -информативные разряды	15 0-14						
Коэффициент подавления помех частотой $(50 \pm 1) Hz$, dB , не менее: -нормального вида -общего вида	40 100						
Вид гальванической развязки	Групповая						
Испытательное напряжение гальванической развязки, не менее, V	1500						
Время опроса одного канала по ИР не более, μs	10						
Вид взрывозащищенности по ГОСТ 22782	Ex ia 11C						

Принцип преобразования входных сигналов в цифровой код поясняется временной диаграммой, приведенной на рисунке Д.18 ($U_{вх}$ - входной сигнал напряжения; $U_{вых}$ - выходной сигнал напряжения; $U_{оп}$ - опорное напряжение). В момент t_1 СчК1 формирует адрес очередного канала преобразователя, по которому $K_{вх}$ подключает ТП к МУ. Через время $t_1 - t_2$ (30 ms), необходимое для установления сигнала на выходе МУ, выходной сигнал МУ подключается к входу ИУ через КН. МУ осуществляет интегрирование сигнала ТП в течение времени $t_2 - t_4$ равного периоду промышленной частоты 50 Hz (20 ms).

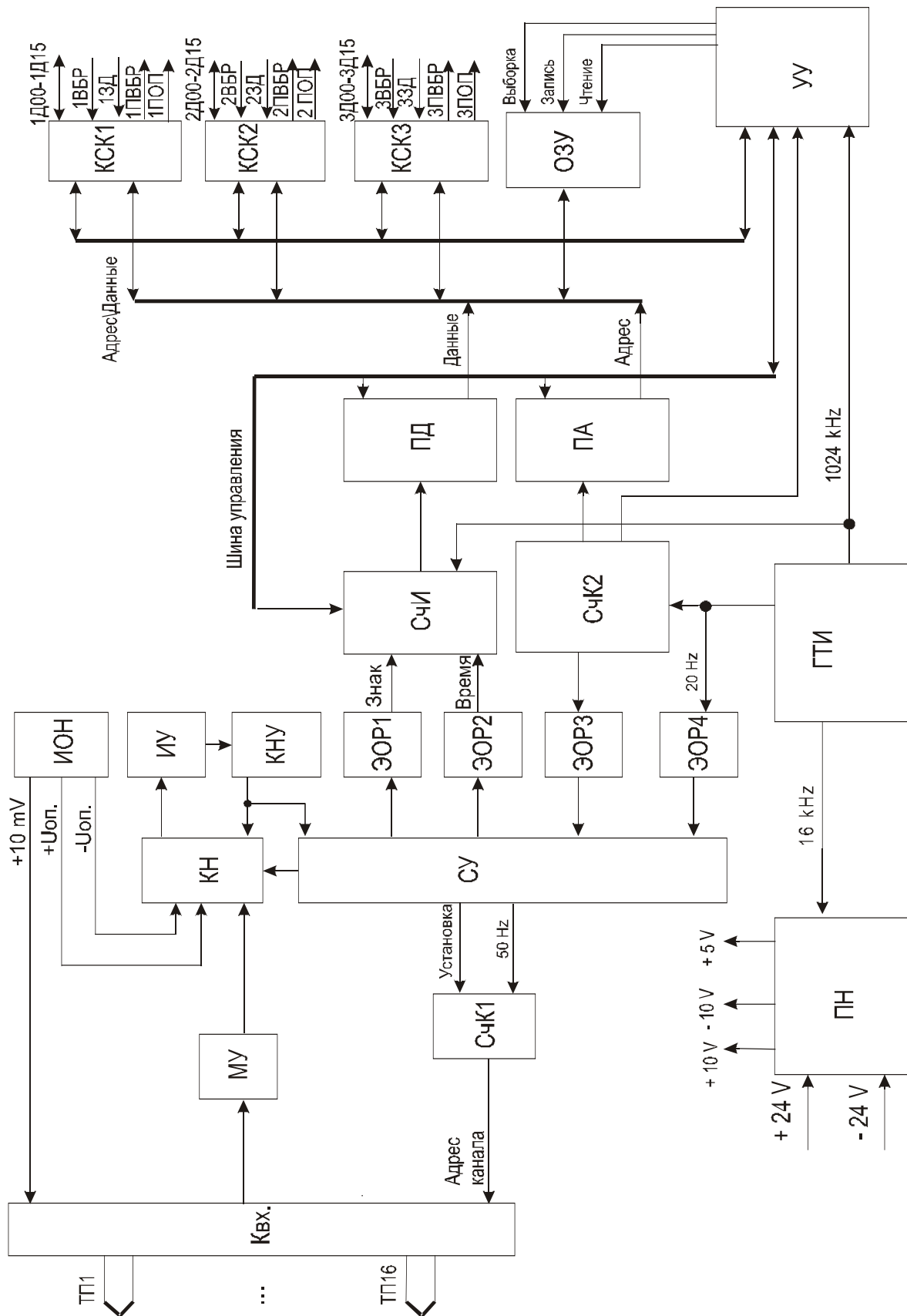


Рис. Д.17. Схема электрическая структурная преобразова-

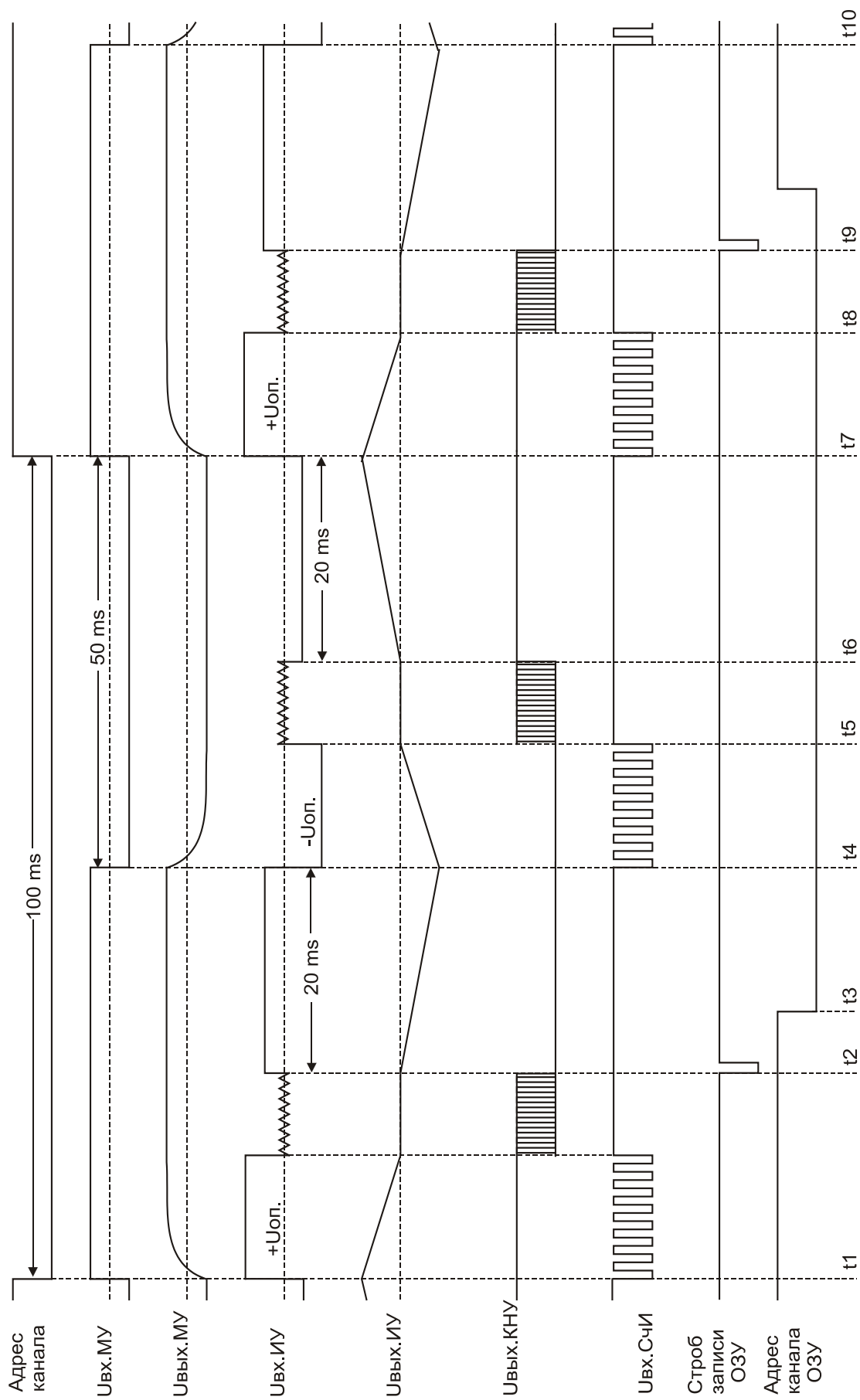


Рис. Д.18. Временная диаграмма сигналов преобразователя

В момент начала интегрирования t_2 СУ формирует сигнал записи результата преобразования по предыдущему каналу. По данному сигналу УУ (в течение времени интегрирования сигнала по текущему каналу) переписывает содержимое СчИ в ОЗУ, устанавливает СчИ в исходное состояние и увеличивает содержимое СчК2 на 1 (t_3). В момент окончания интегрирования t_4 в СУ запоминается знак выходного сигнала ИУ, который определяет знак опорного напряжения, подключаемого в данный момент к ИУ. Одновременно СУ начинает заполнение СчИ импульсами ГТИ.

Под воздействием опорного напряжения выходной сигнал ИУ линейно изменяется в сторону уменьшения до момента достижения нулевого уровня t_5 , фиксируемого КНУ, изменение выходного сигнала которого запрещает поступление импульсов ГТИ на вход СчИ. Так как выходной сигнал интегратора пропорционален среднему значению сигнала ТП за период интегрирования, который постоянен, то и время $t_4 - t_5$ (и количество импульсов в СчИ) также пропорционально среднему значению сигнала ТП. В момент t_5 КН подключает вход ИУ к выходу КНУ, за счет чего обеспечивается стабилизация нулевого уровня выходного сигнала ИУ до начала следующего этапа интегрирования.

С целью увеличения подавления помех и компенсации напряжения смещения МУ в момент t_4 K_{BX} изменяет полярность подключения МУ к ТП и производится повторное преобразование $t_4 - t_8$. В СчИ производится суммирование результатов обоих преобразований. Результат преобразования по данному каналу записывается в ОЗУ в момент t_9 . Аналогичным образом происходит преобразование сигналов ТП по всем другим каналам преобразователя.

После завершения преобразования сигнала по контрольному каналу в момент, аналогичный t_3 , СчК2 вырабатывает импульс, устанавливающий СчК1 в исходное состояние. Таким образом обеспечивается синхронизация счетчиков.

Д.3.5 Блоки ввода частотных сигналов

Преобразователь частотных сигналов ПЧ1(РТ)-8/3 предназначен для преобразования частоты входных токовых сигналов в цифровой код и ввода результата преобразования в КМП по трем направлениям интерфейса ИР. Характеристики каналов ввода при использовании ПЧ1(РТ)-8/3 приведены в табл.Д.9.

Таблица Д9

Характеристики каналов ввода при использовании ПЧ1(РТ)-8/3

Наименование показателя и единица измерения	Значение параметра
Диапазоны входных частот, kHz	8-31875; 4-15937; 2-7968; 1-3984; 1-1992; 1-996; 1-498; 1-249
Время преобразования, t_{np} , s	$t_{np} \leq 3 / F_{ex}$
Параметры входных сигналов: уровень "0"/"1"	(0-1,2) V/(4,2-7,8) V; (0-2,4) V/(9,6-14,4) V; (0-4,8) V/(19,2-28,8) V; (0-9,6) V/(38,4-57,6) V; (0-1) mA/(4-20) mA; (12-2000) mV/(60-10000) mV.
Число подключаемых входных сигналов, шт.	8
Минимальная длительность импульса и паузы, μs	10
Предел допускаемой относительной погрешности преобразования, %	$\pm (0,05 + \frac{N_{6bx}}{10480} + \frac{100}{N_{6bx}})$
Входной ток при номинальном напряжении, mA	7-13
Номинальная: -статическая характеристика преобразования -цена единицы наименьшего разряда выходного кода (μ_{sf})	$N_{6bx} = 4095 \cdot F_{ex} / F$, $\mu_{sf} = F / 4095$ где N_{6bx} – десятичное значение выходного кода; F_{ex} – значение входной частоты, Hz ; F – верхнее значение диапазона входной частоты, Hz ;

	Продолжение табл. Д.9
Формат выходного двоичного кода: -информативные разряды -неинформативные разряды	0-11 12-15
Вид гальванической развязки	Поканальная
Испытательное напряжение гальванической раз- вязки не менее, V	1500-1575
Время опроса одного канала по ИР не более, μs	10

Структурная схема преобразователя приведена на рис. Д.19. Преобразователь содержит:

•узлы нормализации U1-U8, осуществляющие преобразование уровней входных импульсных

•триггеры Шмитта TH1-TH8;
•узлы измерения периодов входных сигналов UT1-UT8 для каналов 1-8;
•коммутатор К;
•узел DIV, выполняющий операцию деления;
•двухпортовую оперативную память RAMDP;
•регистр результата RG;
•шинный формирователь BUSF, служащий для приема адреса канала и выдачи результата преобразования;

•шинные формирователи BUSF1.1, BUSF1.2; BUSF2.1, BUSF2.2; BUSF3.1, BUSF3.2, обеспечивающие прием и выдачу управляющих сигналов, а также выдачу результата преобразования по заданному адресу для каждого из трех направлений ИР;

•генератор импульсов для синхронизации работы преобразователя;
•делитель частоты, формирующий сигналы эталонной частоты и сигналы для тестирования преобразователя;

•схему управления CONTROL, обеспечивающую необходимую последовательность операций при преобразовании сигналов.

Принцип работы. Принцип преобразования входного частотного сигнала в цифровой код заключается в измерении периода входного сигнала путем подсчета количества импульсов эталонной частоты, укладываемого между двумя смежными импульсами входного сигнала, и последующего деления постоянного коэффициента на величину измеренного периода. Коэффициент равен 2^{20} . Эталонная частота импульсов равна $256 \cdot F$, где F – максимальная частота входного сигнала заданного диапазона преобразования. Диапазон преобразования частоты выбирается с помощью джампера. Диапазон - один для всех восьми каналов согласно табл. Д.10.

Таблица Д.10

Диапазон преобразования частоты

Диапазон	“32”	“16”	“8”	“4”	“2”	“1”	“0.5”	“0.25”
Максимальная частота входного сигнала(F), kHz	32	16	8	4	2	1	0,5	0,25

Для каждого из восьми диапазонов измерения частоты допускается зона нечувствительности, обусловленная методом преобразования частоты, при этом в ячейки результата каналов заносится код “0”, и зона “зашкаливания” вследствие превышения частотой входного сигнала максимального значения частоты преобразования для выбранного диапазона, при этом в ячейки результата каналов заносится код 4095. Параметры таких зон приведены в табл. Д.11.

Зоны нечувствительности и зашкаливания

Диапазон	Зона нечувствительности, Hz (код "0")	Диапазон измерения частоты, Hz (коды "1" – "4080")	Зона "зашкаливания", Hz (код "4095")
"32"	1 - 7	8 - 31875	≥ 31876
"16"	1 - 3	4 - 15937	≥ 15938
"8"	1	2 - 7968	≥ 7969
"4"	—	1 - 3984	≥ 3985
"2"	—	1 - 1992	≥ 1993
"1"	—	1 - 996	≥ 997
"0.5"	—	1 - 498	≥ 499
"0.25"	—	1 - 249	≥ 250

Токовые сигналы от датчиков поступают на вход узлов нормализации U1-U8 (рисунок Д.19). U1-U8 служат для нормирования уровней входных сигналов и гальванической развязки входных цепей от цепей управления. С выхода узлов нормализации входные сигналы преобразователя поступают в узлы измерения периода входных сигналов UT1-UT8 через триггеры Шмитта ТН1-ТН8, являющиеся пороговыми элементами, увеличивающими помехоустойчивость преобразователя.

Схемы узлов измерения периода входных сигналов UT1-UT8, коммутатора К, узла DIV, оперативной памяти RAMDP, регистра результата RG, шинного формирователя BUSF, делителя частоты DIVE, а также узла управления CONTROL помещены в программируемую логическую интегральную схему.

Узлы UT1-UT8 содержат счетчики для измерения периода входных сигналов путем подсчета количества импульсов эталонной частоты, укладываемого между двумя смежными импульсами входных сигналов. Коды счетчиков через коммутатор К поступают на вход узла DIV, выполняющего операцию деления для вычисления частоты входного сигнала.

Результат преобразования для каждого из восьми каналов заносится в оперативную память RAMDP. RAMDP представляет собой двухпортовую оперативную память, позволяющую проводить операции записи и чтения независимо друг от друга. По запросу от ИП выполняется чтение информации из RAMDP по адресу канала с записью ее в регистр результата RG. Адрес канала передается через шинные формирователи направлений ИП, через шинный формирователь BUSF и поступает на вход адреса чтения RAMDP. Код преобразования частоты из регистра RG через шинный формирователь BUSF выдается на шинные формирователи направлений ИП. Шинные формирователи BUSF1.1 - BUSF3.2 служат для передачи управляющих и информационных сигналов по каждому из трех направлений ИП.

Схема управления CONTROL представляет собой схемный автомат, вырабатывающий последовательность синхросигналов, координирующих работу отдельных узлов преобразователя, а также служащий для отработки временной диаграммы сигналов интерфейса ИП. Генератор G вырабатывает сигналы с частотой $(16384 \pm 8,2) \text{ kHz}$, которые поступают на делитель частоты DIV. Сигналы с выхода делителя частоты используются как импульсы эталонной частоты в узлах измерения периодов входных сигналов, а также для тестирования преобразователя.

Частота эталонных импульсов соответствует выбранному диапазону измерения. Эта частота - одна из ряда (8192, 4096, 2048, 1024, 512, 256, 128, 64) kHz.

Д.3.6 Блоки ввода время-импульсных сигналов

Преобразователь время-импульсных сигналов ПВИ(РТ)-8/3. Преобразователь обеспечивает преобразование время-импульсных токовых сигналов в 16-разрядный двоичный код и выдачу результата в КМп по трем направлениям интерфейса ИП. Входные каналы преобразователя гальванически развязаны от заземленных цепей питания и друг от друга. Характеристики каналов ввода при использовании ПВИ(РТ)-8/3 приведены в табл. Д.12.

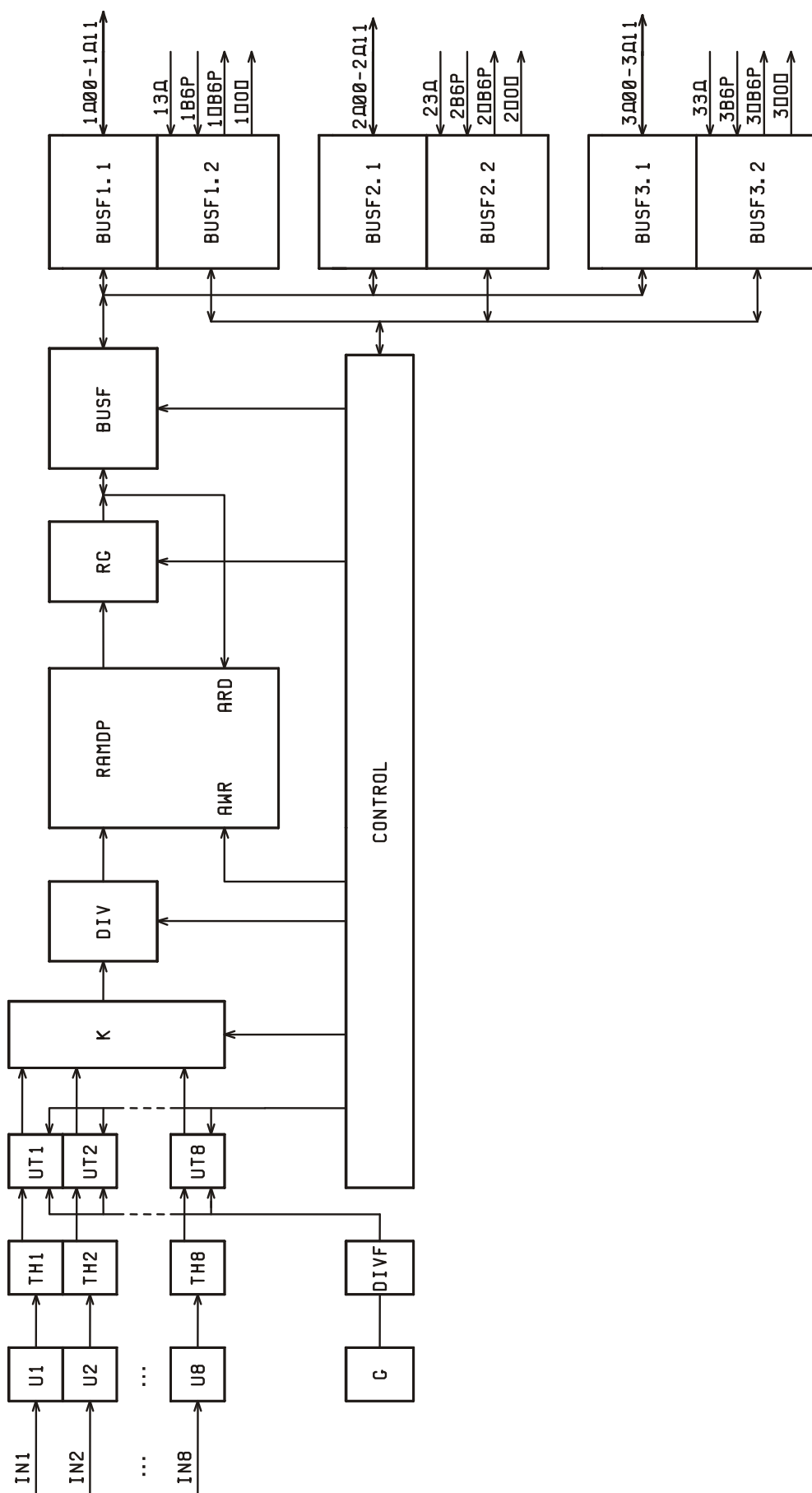


Рис. Д.19. Схема электрическая структурная преобразователя

Характеристики каналов ввода при использовании ПВИ(РТ)-8/3

Наименование показателя и единица измерения	Значение параметра
Измеряемый параметр	Период, длительность
Диапазоны изменения входных сигналов, с: -минимальный -максимальный	0-2 0-128
Параметры входных сигналов: уровень "0"/"1"	(0-1,2) В/(4,2-7,8) В; (0-2,4) В/(9,6-14,4) В; (0-4,8) В/(19,2-28,8) В; (0-9,6) В/(38,4-57,6) В; (0-1) мА/(4-20) мА.
Входной ток при номинальном напряжении ("1"), мА	7-13
Число подключаемых входных сигналов, шт.	8
Пределы допускаемой относительной погрешности преобразования, %	$\pm (0,01 + \frac{200}{N_{\text{вых}}})$, где $N_{\text{вых}}$ - десятичное значение выходного кода
Номинальная функция преобразования	$N_{\text{вых}} = \frac{T_{\text{вх}}}{T} \cdot 65535$, где $N_{\text{вых}}$ - десятичное значение выходного сигнала; $T_{\text{вх}}$ - значение входного импульса, с; T - верхнее значение диапазона входного импульса, с
Формат выходного кода: -информативные разряды	0-15
Вид гальванической развязки	Покальная
Испытательное напряжение гальванической развязки, не менее, В	500
Время опроса одного канала по ИР, мс, не более	10

Принцип работы. Преобразователь содержит (рис. Д.20):

- нормализаторы U1-U8, осуществляющие преобразование уровней входных сигналов и гальваническую развязку;
- коммутатор МХ, переключающий каналы;
- формирователь временных интервалов ФВИ, формирующий временные интервалы;
- генератор тактовых импульсов ГТИ, формирующий последовательность тактирующих импульсов;
- схему синхронизации СС, обеспечивающую синхронизацию приема адреса и считывания информации с работой преобразователя;
- кратковременное запоминающее устройство КЗУ, кратковременно хранящее результат опроса каналов;
- счетчик импульсов СЧ, суммирующий импульсы заполнения;
- делитель частоты ДЧ, формирующий адрес канала, импульсы для заполнения;
- формирователь импульсов заполнения ФИЗ, формирующий импульсы заполнения;
- узел совпадения УС, пропускающий импульсы совпадения;
- оперативное запоминающее устройство ОЗУ, хранящее результат предыдущего опроса каналов;
- выходные усилители ВУ1-ВУ3, обеспечивающие запоминание результата преобразования из КЗУ и выдачу на интерфейс ИР по трем направлениям;
- схема проверки СП, обеспечивающая выдачу проверочных сигналов.

Работа преобразователя основана на счетно-импульсном принципе, заключающемся в том, что СЧ считает количество поступающих на его вход импульсов в течение определенного интервала времени. Измеряемые сигналы с входа вилки ХЗ через нормализаторы и коммутатор поступают на схему формирования временных интервалов. При измерении СЧ считает количе-

H. HU/PT6) 22/2

Характеристики каналов ввода при использовании НД(РТФ)-32/3

Наименование показателя и единица измерения	Значение параметра
Параметры входных сигналов: уровень “0”/“1”	(0-1,2) V/(4,2-7,8) V; (0-2,4) V/(9,6-14,4)V; (0-4,8) V/(19,2-28,8) V; (0-9,6) V/(38,4-57,6) V; (0-1) mA/(4-20) mA.
Входной ток при номинальном напряжении (“1”), mA	7-13
Число подключаемых входных сигналов, шт.	32 (2 группы по 16 каналов)
Время опроса по ИР группы из 16 сигналов не более, μ s	10
Вид гальванической развязки	Покаанальная
Испытательное напряжение гальванической раз- вязки, не менее, V	500
Постоянная времени фильтра, ms (для НД(РТФ)- 32/3)	2 $\pm$ 0,5

Принцип работы. Нормализатор состоит из следующих функциональных узлов (рис. Д.21):

- узлов нормализации U1-U32 (индивидуально для каждого канала), обеспечивающих гальваническую развязку входных каналов, а также преобразование входных сигналов (совместно с компаратором в схеме фильтров Z1 - Z8) до уровней логических сигналов;
- коммутаторов K1 - K3 (индивидуально по каждому направлению), служащих для поочередного подключения групп входных каналов к шинным формирователям;
- фильтров Z1-Z8 (по одному на каждые четыре канала), обеспечивающих повышение помехоустойчивости, а также защиту от дребезга входного сигнала;
- шинных формирователей ШФ1-ШФ3 (индивидуально по каждому направлению ИР), обеспечивающих выдачу информации на шины данных ИР;
- узлов управления УУ1-УУ3 (индивидуально по каждому направлению), формирующих необходимые сигналы управления;
- узла диагностики УД, обеспечивающего контроль целостности электрических связей нормализатора и ПСД.

Сигналы от датчиков тока поступают в узлы нормализаторов U1 – U32. Схема электрическая принципиальная узла нормализатора приведена на рис Д.22. Узлы нормализации совместно с компаратором в схеме фильтров Z1 - Z8 преобразуют входные сигналы в логические сигналы. Гальваническая развязка входных цепей реализуется посредством оптопары, а фильтры Z1 - Z8 повышают помехоустойчивость входных каналов, обеспечивают защиту от импульсных помех и от дребезга входного сигнала в каждом канале.

Схема фильтра (рис. Д.23) реализована на компараторе D25 (для четырех входных каналов), на инвертирующие входы которого подано опорное напряжение 2,5 V. На конвертирующие входы подается напряжение с выходов узлов нормализаторов с задержкой. Задержка для каждого из четырех каналов задается постоянной времени цепи R39, C17; R40, C18; R41, C19; R42, C20. Если входной сигнал изменился с низкого на высокий уровень или наоборот, то сигнал на выходе компаратора появится после того, как напряжение на неинвертирующем входе достигнет величины 2,5 V (через 0,7 RC). Делитель R38, R44 служит для формирования гистерезиса с шириной петли около 40 mV. Цепь R44, C21 служит для предотвращения помех на выходах компаратора D25 при возникновении пульсаций в цепи питания, C22 - для предотвращения помехи на инвертирующих входах (опорный сигнал).

Нормализаторы и фильтры являются общей частью для всех направлений.

Далее информация через коммутаторы K1 - K3 поступает на ШФ1 - ШФ3 по трем направлениям ИР.

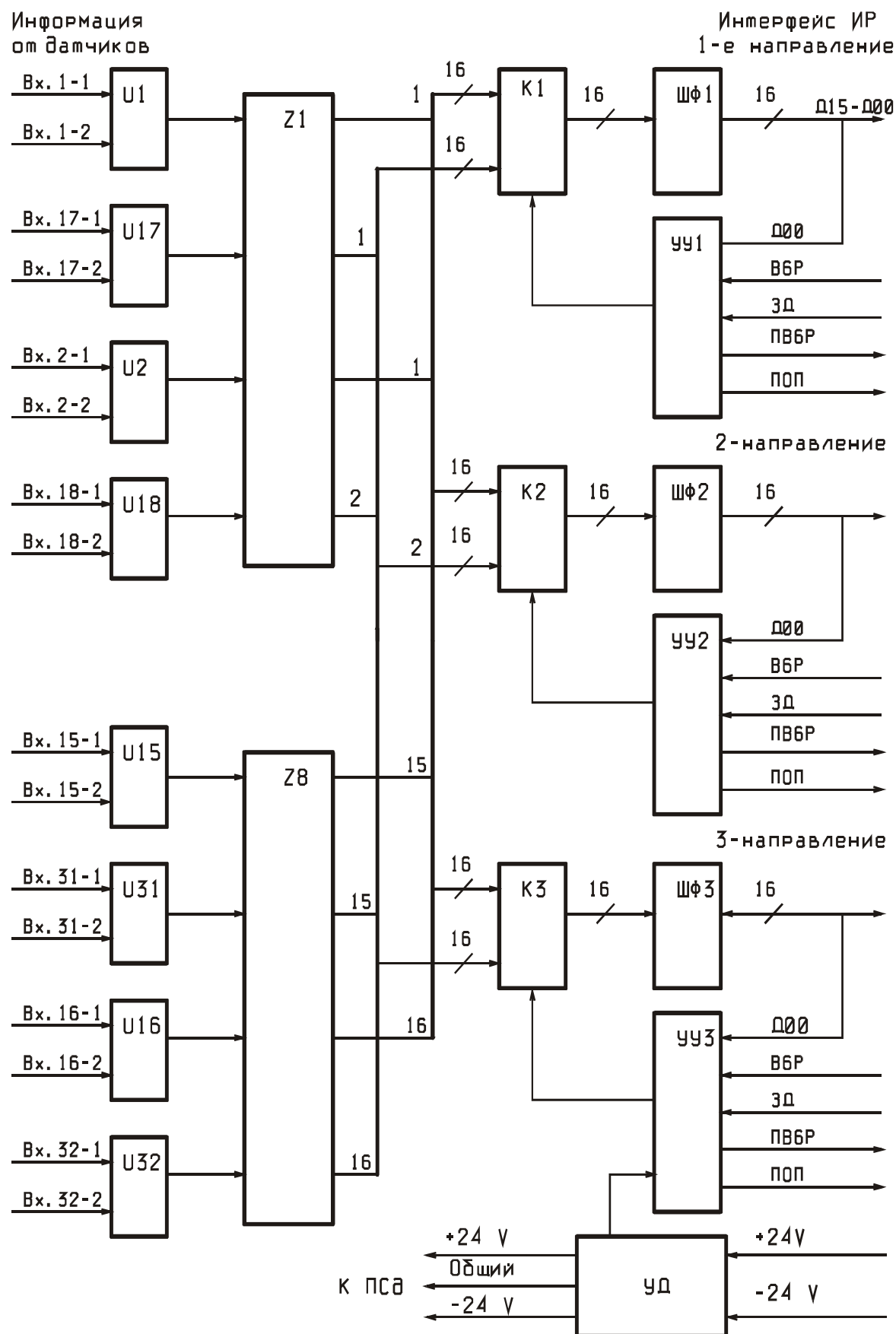


Рис. Д.21. Схема структурная нормализатора.

ШФ1 - ШФ3 выходят на магистральные шины адрес/данные интерфейса ИР. Формирование управляющих сигналов для К1 - К3 и ШФ1 - ШФ3 производится узлом управления.

Предусмотрена возможность приема сигналов от датчиков типа «Сухой контакт» и потенциальных датчиков +24 V через панели соединительные.

Питание датчиков типа «Сухой контакт» производится от системного источника питания МСКУ.

Для контроля работоспособности ПСд 103 и целостности электрической связи с нормализатором имеется узел диагностики УД.

Д.3.8 Блоки ввода сигналов от датчиков типа «сухой контакт»

Нормализатор дискретных сигналов НД(РКФ)-16/3 состоит из следующих функциональных узлов (рисунок Д.24, Д25):

- узлов нормализации U1-U16 (индивидуально для каждого канала), обеспечивающих гальваническую развязку входных каналов, а также преобразование входных сигналов (совместно с компаратором в схеме фильтров Z1-Z4) до уровней логических сигналов;
- трансформаторов Т1-Т8 (один для двух каналов), обеспечивающих подключение источника питания к входной цепи датчика;
- модуляторов UB1-UB4 (один для четырех каналов), обеспечивающих организацию питания входных цепей датчика;
- фильтров Z1-Z4 (один для четырех каналов), обеспечивающих повышение помехоустойчивости входных каналов, а также защиту от дребезга входного сигнала;
- шинных формирователей ШФ1-ШФ3 (индивидуально по каждому направлению ИР), обеспечивающих связь с КМп по ИР;
- узлов управления У1-У3 (индивидуально по каждому направлению), обеспечивающих формирование управляющих сигналов для функциональных узлов нормализатора.

Характеристики каналов ввода при использовании НД(РКФ)-16/3 приведены в табл. Д14.

Информация об изменении величины сопротивления датчиков (контакты реле, транзистор с "открытым" коллектором) поступает в U1-U16, где совместно с компаратором (в схеме фильтров Z1-Z4) производится преобразование входного сигнала в логический сигнал. Гальваническая развязка входных цепей реализуется посредством оптопары, а фильтры Z1 – Z4 повышают помехоустойчивость входных каналов, обеспечивают защиту от импульсных помех и от дребезга входного сигнала в каждом канале. Далее информация поступает на ШФ1-ШФ3, имеющие выход на ИР.

Передача входной информации через U1-U16 и схемы фильтров Z1-Z4 на шины данных ИР осуществляется при наличии сигнала управления на ШФ1-ШФ3, который формируется узлами управления У1-У3.

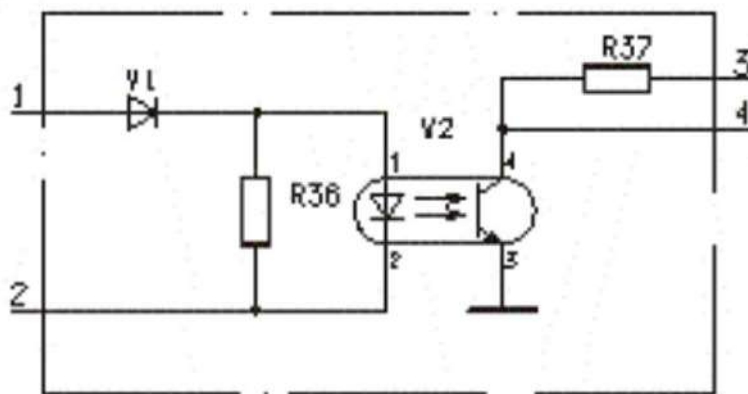


Рис. Д.22. Схема электрическая принципиальная узла нормализации.

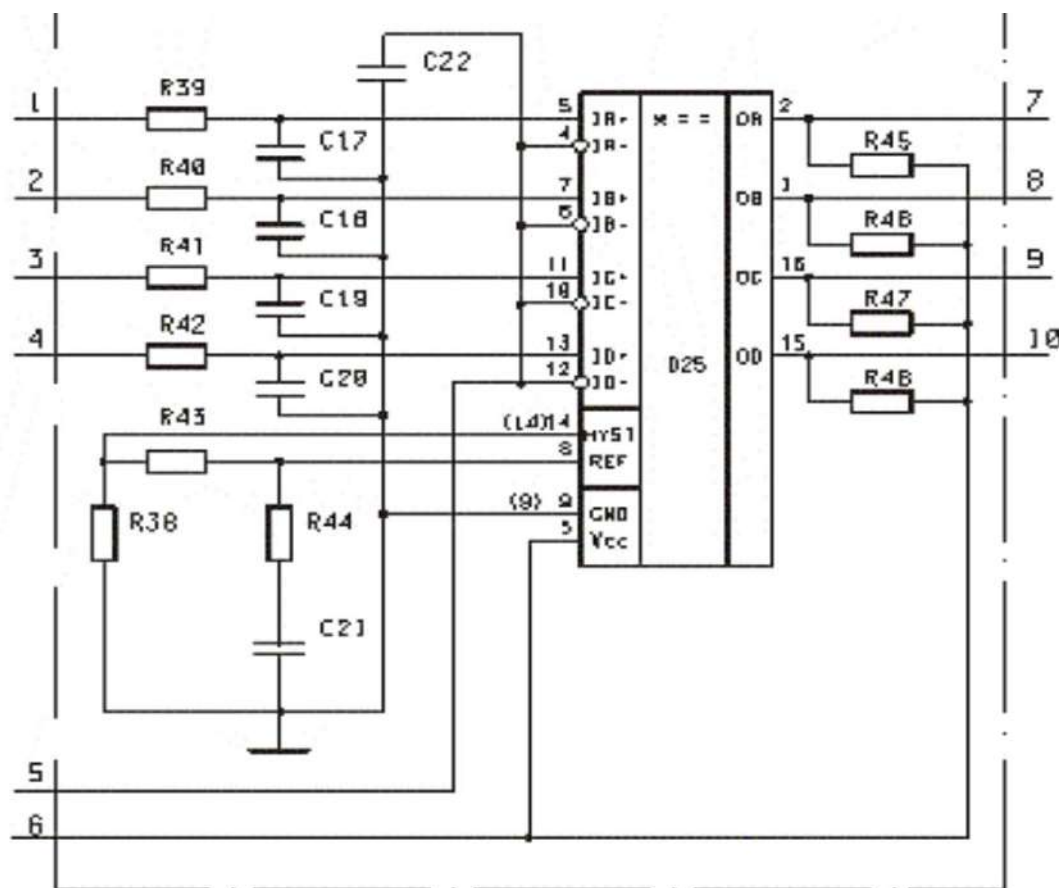


Рис. Д.23. Схема электрическая принципиальная фильтра.

В нормализаторе каждая входная цепь датчика запитывается от изолированного источника питания, состоящего из модуляторов (UB1-UB4), трансформаторов, а также выпрямителя (V8, V9) и фильтра (C22, C23), расположенных в U1 – U16 (рис. Д.26). Изменение сопротивления датчика приводит к изменению рабочего тока в цепи датчика и диода оптопары V10, включенного в эту цепь.

При поступлении низкого уровня (R^0) на входы узлов нормализаторов U1 – U16 токи через диоды оптопар V10 не протекают, транзисторы оптопар V10 закрыты и на выходах U1 – U16 присутствуют сигналы логической "1".

Информация
от датчиков

Интерфейс ИР

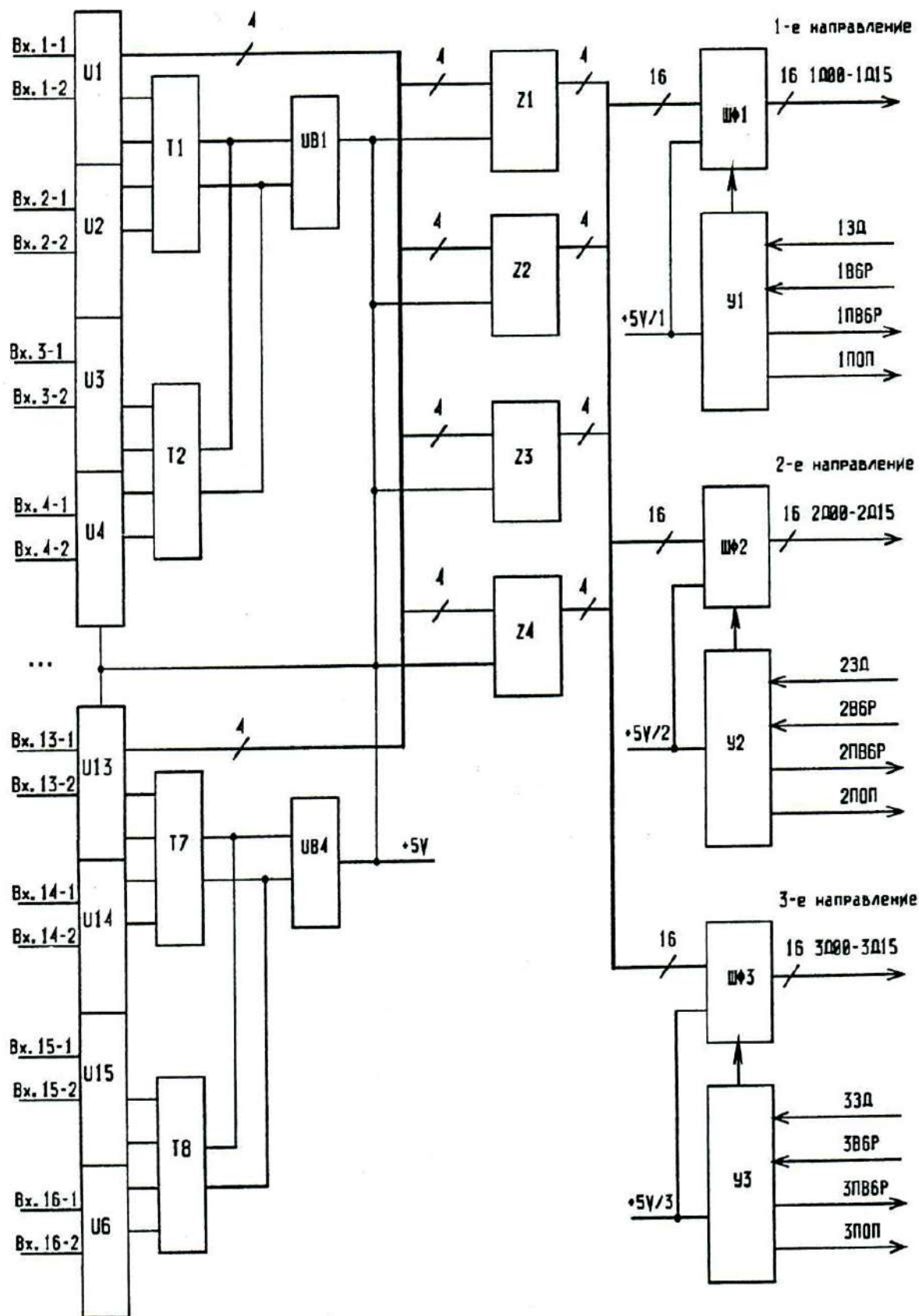


Рис. Д.24. Схема электрическая структурная нормализатора.

Схема модуляторов UB1...UB4

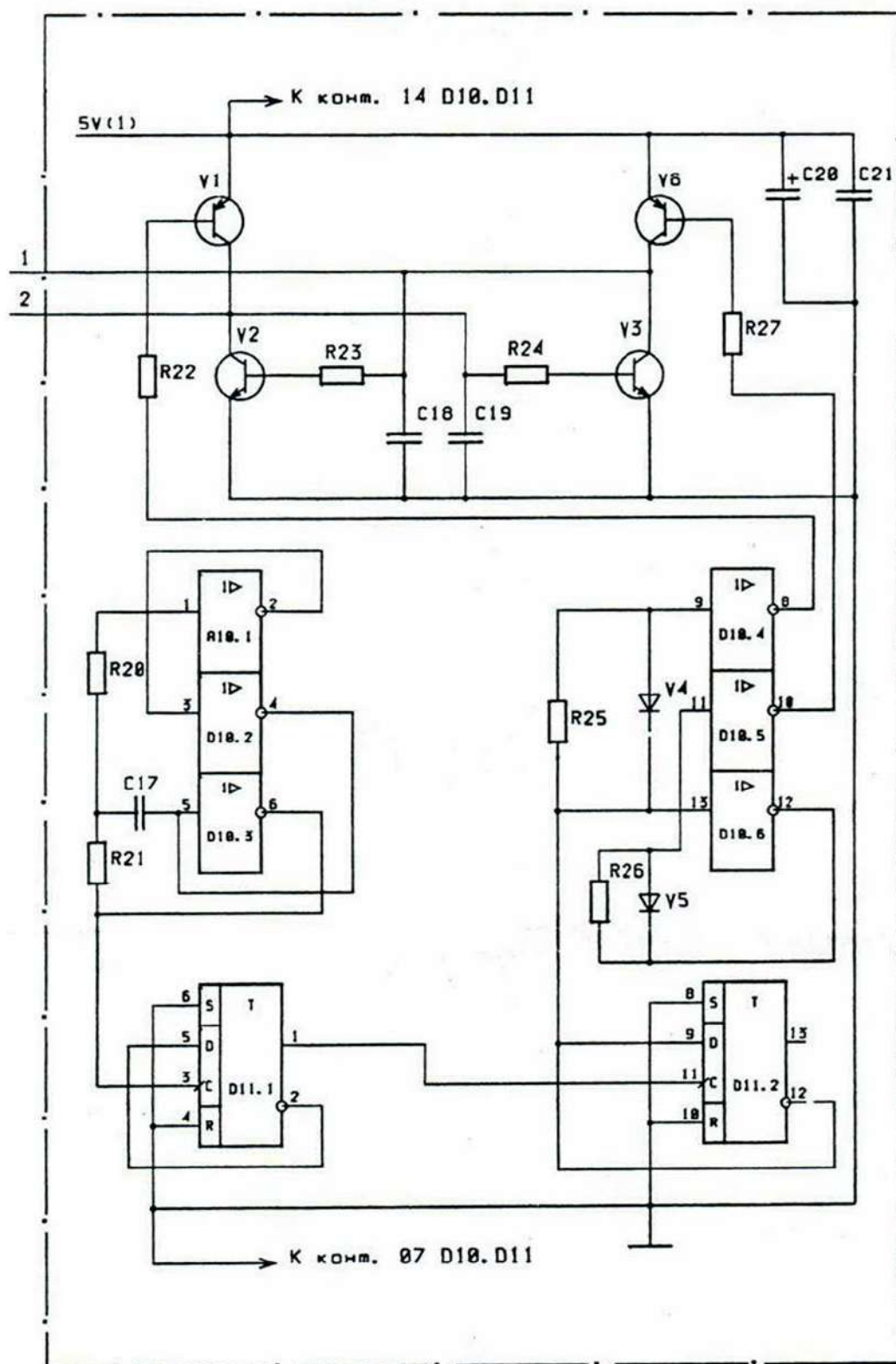


Рис. Д.25 –Схемы модуляторов

Схема нормализаторов U1...U16

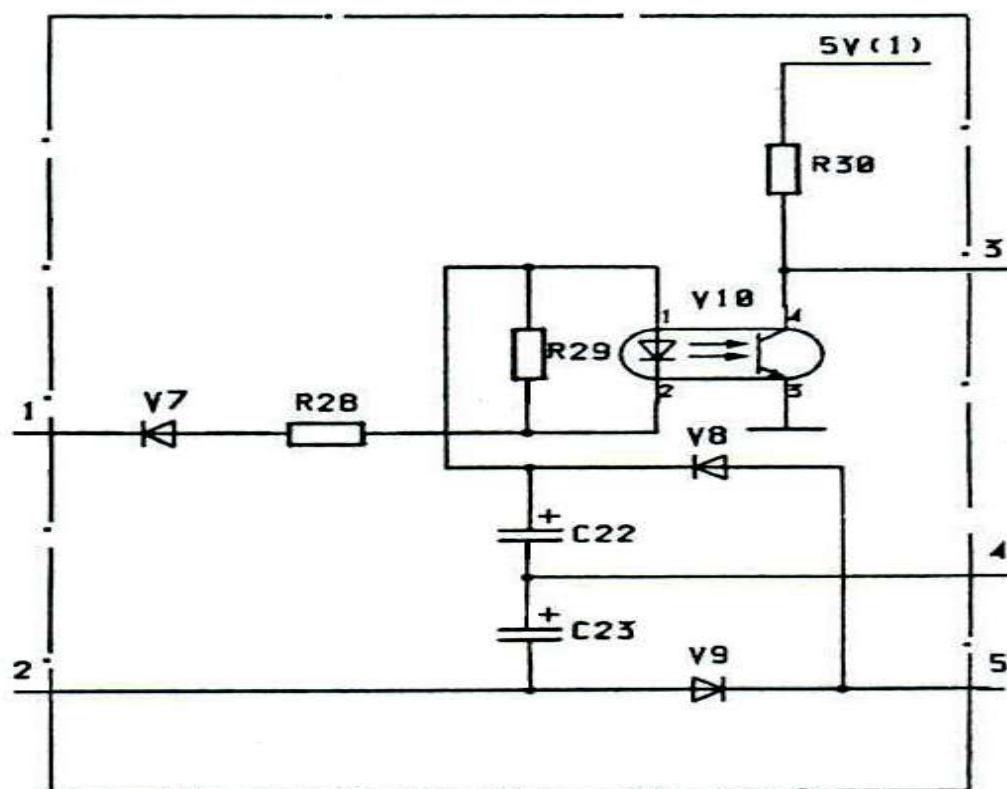


Схема фильтров Z1...Z4

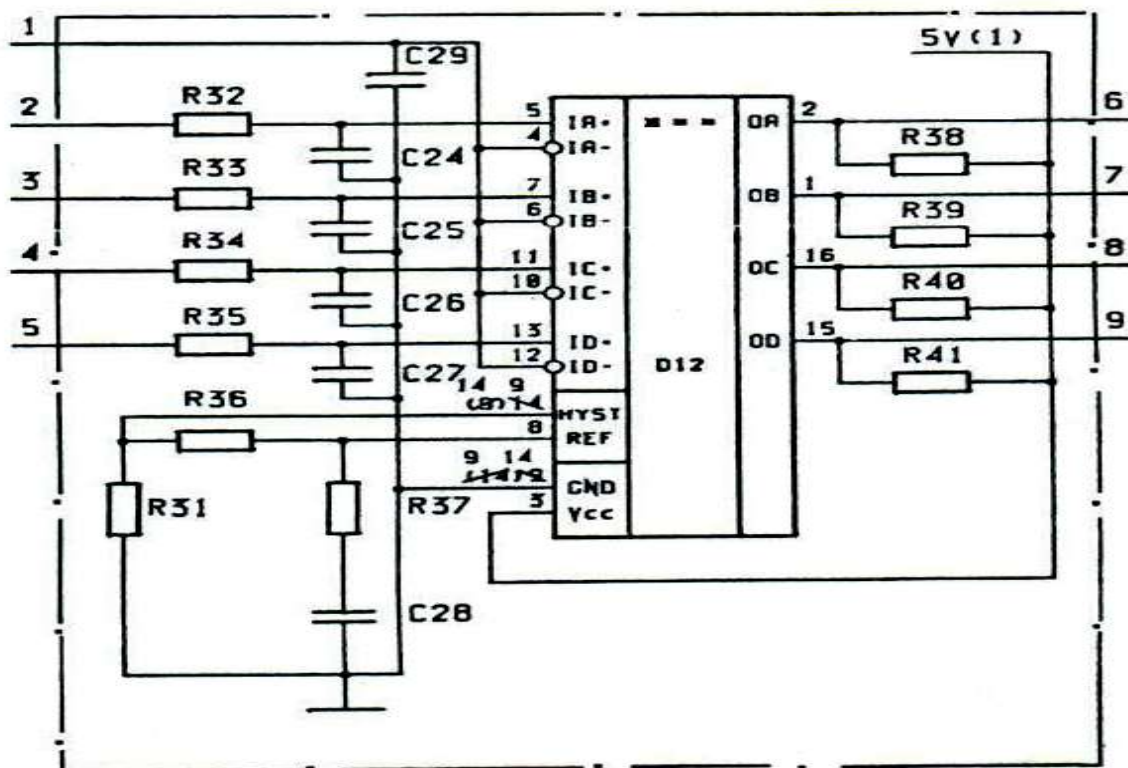


Рис. Д.26. Схемы нормализаторов и фильтров.

При поступлении высокого уровня (R^1) на входы $U1-U16$ транзисторы оптопар $V10$ открываются за счет тока, протекающего через диоды оптопар $V10$, и на выходах узлов нормализаторов $U1-U16$ присутствуют сигналы логического "0". Резистор $R29$ в узлах $U1-U16$ определяет уровень срабатывания оптопары $V10$. Резистор $R28$ во входных цепях ограничивает ток через диоды оптопар $V10$. Диод $V7$ защищает диод оптопары $V10$ от обратного напряжения.

Информация об изменении величины сопротивления датчиков (контакты реле, транзистор с "открытым" коллектором) поступает в $U1-U16$, где совместно с компаратором (в схеме фильтров $Z1-Z4$) производится преобразование входного сигнала в логический сигнал. Гальваническая развязка входных цепей реализуется посредством оптопары, а фильтры $Z1-Z4$ повышают помехоустойчивость входных каналов, обеспечивают защиту от импульсных помех и от дребезга входного сигнала в каждом канале. Далее информация поступает на ШФ1-ШФ3, имеющие выход на ИР.

Передача входной информации через $U1-U16$ и схемы фильтров $Z1-Z4$ на шины данных ИР осуществляется при наличии сигнала управления на ШФ1-ШФ3, который формируется узлами управления $У1-У3$.

Таблица Д14

Характеристики каналов ввода при использовании НД(РКФ)-16/3

Наименование показателя и единица измерения	Значение параметра
Параметры входных импульсов: -замкнутый контакт, Ω -разомкнутый контакт, $k\Omega$	менее 200 более 50
Число подключаемых входных сигналов, шт.	16
Питание подключаемых датчиков (внутренний источник питания): -напряжение, V -ток, через замкнутый контакт датчика не более, mA	7 ± 2 7
Вид гальванической развязки	Поканальная
Испытательное напряжение гальванической развязки, не менее, V	500
Время опроса по ИР, μs , не более	10
Постоянная времени фильтра, ms (для НД(РКФ)-16/3)	$2 \pm 0,5$

Трансформаторы $T1-T8$ осуществляют передачу переменного напряжения от модуляторов $UB1-UB4$ к узлам нормализаторов $U1-U16$. Дополнительно трансформаторы $T1-T8$ выполняют функцию гальванической развязки каналов нормализатора и шин питания $+5 V$.

Каждый из модуляторов $UB1-UB4$ (рис. Д.25) является преобразователем напряжения постоянного тока, работающим по принципу мостового модулятора, управляемого от ГТИ через делитель частоты $D11.1, D11.2$ и фазосдвигающие звенья $V4, V5, R25, R26$, инверторы $D10.4-D10.6$, которые устраняют перекрытие транзисторных ключей $V1-V3, V6$. Конденсаторы $C18, C19$ снижают скорость изменения напряжения на концах обмоток трансформаторов $T1-T8$ с целью уменьшения высокочастотного тока утечки между первичной и вторичными обмотками трансформаторов $T1-T8$. ГТИ для каждого из модуляторов $UB1-UB4$ реализован на инверторах микросхем $D10.1-D10.3$, образующих инвертирующий усилитель, возбуждаемый положительной обратной связью через конденсатор $C17$. Частота колебаний ГТИ равна $(60 \pm 5) kHz$.

Д.3.9 Блоки ввода сигналов от датчиков типа «сухой контакт» с контролем линий связи

Нормализатор дискретных сигналов НД(РК)-28/3 предназначен для приема, преобразования и ввода в КМп информации о состоянии датчиков типа "сухой контакт", установленных на объекте. Нормализатор подключается к датчикам типа "сухой контакт" через две панели ПКр-24 или две панели ПСд-101 или одну панель ПСд 107. Обеспечивает ввод состояний 28 сигналов датчиков типа "сухой контакт".

Нормализатор имеет три канала связи с КМп по интерфейсу ИР, двадцать восемь рабочих каналов связи с датчиками и четыре контрольных канала, обеспечивающих проверку работоспособности нормализатора.

Если датчик шунтирован резистором С2-23-0,5-1,5 $k\Omega$, сохраняющим проводимость цепи при разомкнутом контакте, нормализатор обеспечивает контроль обрыва линии связи с датчиком.

В нормализаторе предусмотрен изолированный источник питания для контроля состояния контактов датчиков и линии связи. Напряжение - от 18 до 22 V. Допускается использование внешнего источника питания, напряжением от 21 до 30 V.

Нормализатор обеспечивает непрерывный циклический ввод состояний контактов и линий связи с запоминанием информации в буферной памяти. Время цикла опроса не более 4 ms. В нормализаторе предусмотрена возможность работы с аналогичным резервным блоком на одну группу датчиков.

Формат записи информации в буферной памяти следующий: тридцать два датчика разбиты на восемь групп с адресами от 0 до 7, по четыре датчика в группе. Адрес 7 присвоен группе контрольных каналов и информация по датчикам в группе представлена младшим байтом слова данных. Младшие четыре разряда содержат побитную информацию о состоянии контактов. Логическая "1" - контакт замкнут, "0" - контакт разомкнут.

Таблица Д 15

Характеристики каналов ввода дискретных сигналов от датчиков типа «сухой контакт» с контролем линий связи при использовании НД(РК)-28/3.

Наименование показателя и единица измерения	Значение параметра
Параметры входных сигналов: -замкнутый контакт, Ω -разомкнутый контакт, $k\Omega$ -разрыв линии связи с датчиком $k\Omega$	менее 130 от 1,4 до 1,8 более 20
Число подключаемых входных сигналов, шт.	28 (2 группы по 14 каналов)
Питание подключаемых датчиков: напряжение, V: от внутреннего источника от внешнего источника; ток через замкнутый контакт, mA, не более	От 17 до 22 От 22 до 30 40
Вид гальванической развязки	Групповая
Испытательное напряжение гальванической развязки, не менее, V	1000
Длительность цикла опроса, ms, не более	4
Время опроса по ИР состояния входных сигналов одной группы, μs , не более	10

Принцип работы. Нормализатор содержит следующие функциональные узлы (рис. Д.27): преобразователь напряжения ПН; коммутатор К; узел преобразования ток-напряжение УПТН; узел компараторов УК; узел регистров сдвига УРС; узел буферной памяти УБП; каналы связи с

КМп КСК; ключ Кл; узел синхронизации УС; клапан В; тактовый генератор ТГ; счетчик адреса СчА1; счетчик адреса СчА2; дешифратор Дш.

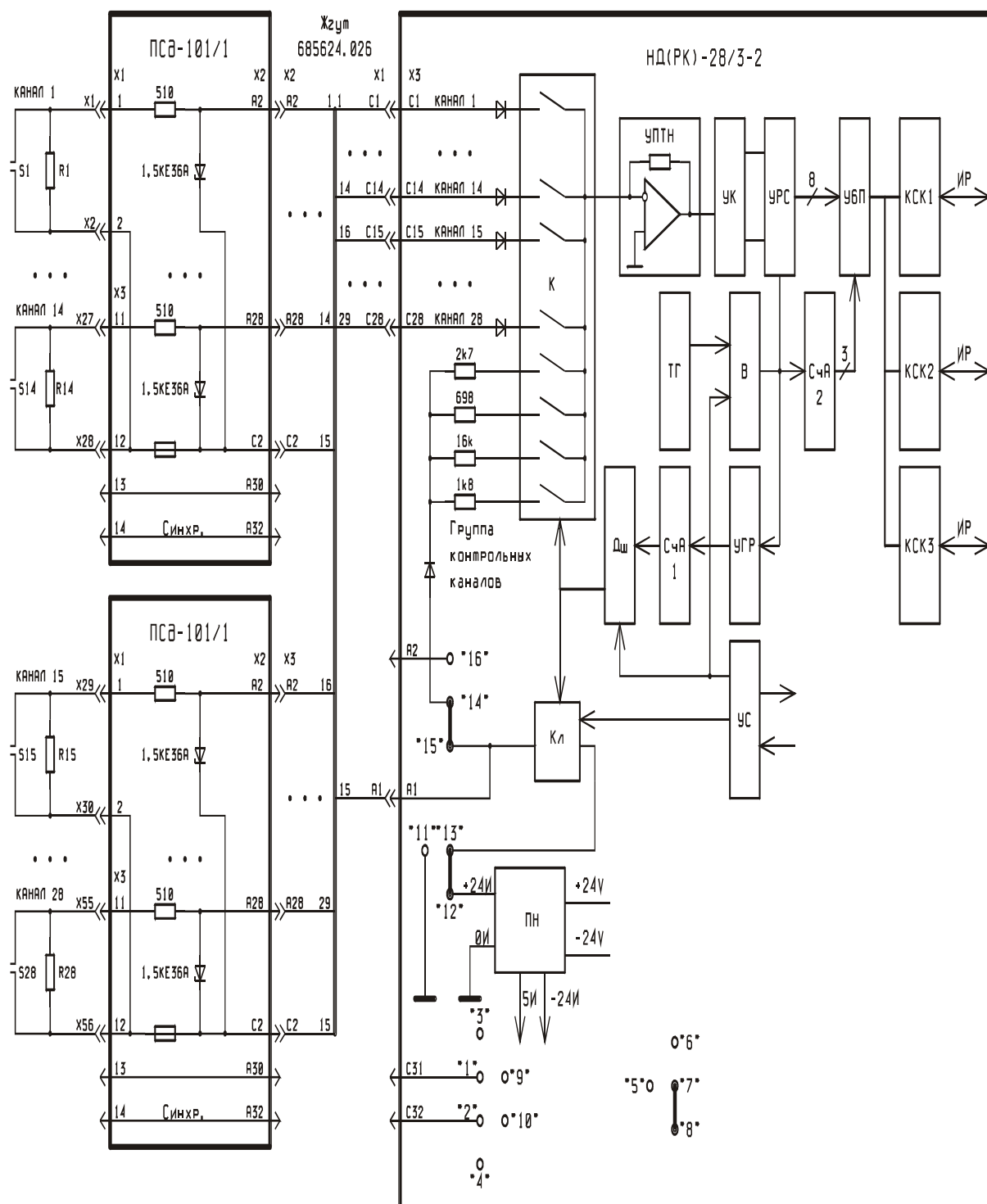


Рис. Д.27. Структурная схема НД(РК)-28/3-2 без резервирования с внутренним источником питания датчиков.

Преобразователь напряжения ПН формирует изолированные напряжения для питания датчиков и изолированных узлов нормализатора. Напряжение питания через ключ Кл и токозадающие резисторы, расположенные в ПСд, поступает на датчики и возвращается на входы блока.

Коммутатор *K* под управлением счетчика адреса СчА1 и дешифратора Дш поочередно подключает цепи датчиков ко входу узла преобразования ток-напряжение (УПТН). На входы 29-32 коммутатора через токозадающие резисторы подаются сигналы, соответствующие следующим состояниям рабочих каналов: разомкнутый контакт, замкнутый контакт, обрыв линии связи; УПТН выполняет преобразование тока в напряжение.

Узел компараторов УК преобразует сигнал напряжения в два бита дискретной информации, соответствующие состоянию контакта датчика и линии связи и поступающие на вход узла регистров сдвига,

Узел регистров сдвига УРС содержит два четырехразрядных сдвиговых регистра, преобразующих последовательный код от четырех датчиков в параллельный восьмиразрядный код. Восьмиразрядный код, содержащий информацию по четырем каналам группы датчиков, переписывается в узел буферной памяти. Адресация ячеек УБП осуществляется счетчиком адреса СчА2.

Выход УБП связан с КСК1 – КСК3, обеспечивающими передачу информации на ИР.

Ключ Кл отключает источник питания под управлением дешифратора Дш (для контроля) или под управлением узла синхронизации УС (при работе нормализатора в резервированной системе).

R1...R28 – резистор С2-23-0,5-1,5 $\kappa\Omega$ $\pm 5\%$ S1...S28 – датчик типа «сухой контакт»

X1...X56 – контакт-клипса 7.732.255

Жгут 685624.026 входит в комплект поставки МСКУ ПС1001.90. Вместо двух ПСд-101 возможно подключение через две ПКр-24.

Узел синхронизации УС, обеспечивает блокировку работы данного нормализатора (на время работы резервного), при этом УС отключает питание датчиков, останавливая цикл опроса и закрывая вентиль В, и отключает дешифратор Дш, при этом выключаются все каналы коммутатора.

Вентиль В передает тактовые импульсы на счетчики адреса СчА1 и СчА2 в рабочем режиме.

Тактовый генератор ТГ генерирует тактовые импульсы для синхронизации работы узлов нормализатора.

Подключение датчиков к нормализатору осуществляется через две панели ПСд-101. Каждая панель имеет клеммы для подключения четырнадцати датчиков и клеммы для подключения сигнала синхронизации при резервировании нормализаторов.

В ПСд установлены ограничители напряжения, обеспечивающие защиту входных узлов нормализатора от перенапряжений, возникающих на линиях связи, кроме того, установлен предохранитель (0,5 А; 250 В), предназначенный для защиты входных цепей нормализатора от токов, превышающих номинальный (40 мА).

Д.3.10 Блоки вывода аналоговых сигналов тока

Формирователь тока ФТ-16. Формирователь выполняет функции 16-канального цифро-аналогового преобразователя, осуществляющего прием от трех КМп цифровых кодов, мажорирование и преобразование в сигнал постоянного тока в соответствии с результатами мажорирования. Мажорирование производится с учетом состояния сигналов РАБ интерфейса ИР. При выключении всех сигналов РАБ формирователь сохраняет на выходе установленное ранее значение сигнала.

Принцип работы. Формирователь содержит (рисунок Д.28):

- АЦП, служащий для калибровки выходных каналов формирователя;
- ЦАП, осуществляющий преобразование двоичного кода в сигнал напряжения;
- схему управления СУ, формирующую тактовый сигнал для работы ПН, а также осуществляющую связь ОЗУ с МК и БУ1-БУ3;
- микроконтроллер МК, осуществляющий мажорирование и масштабирование цифровых данных, поступающих по ИР, и регенерацию выходных сигналов формирователя;

- ОЗУ, служащее для хранения преобразуемых кодов;
- буферные усилители БУ1-БУ3 для согласования формирователя с ИР;
- дешифратор Дш для управления ключами аналоговой памяти $U1-U8$ в соответствии с адресом, полученным от МК;
- коммутаторы К1-К4 для подключения ЦАП к входам усилителей $U1-U8$;
- выходные усилители $U1-U8$, преобразующие сигналы ЦАП в сигналы постоянного тока и осуществляющие гальваническое разделение выходных цепей формирователя;
- преобразователь напряжения ПН, формирующий постоянные напряжения +15 и минус 15 V и переменные напряжения 24 V для питания выходных усилителей $U1-U8$.

Таблица Д16

Характеристики каналов вывода при использовании ФТ-16

Наименование показателя и единица измерения	Значение параметра	
Диапазоны изменения выходных сигналов, mA	0-5	0-20
Сопротивление нагрузки, Ω , не более	2000	500
Число подключаемых выходных каналов, шт.	16 (в любом сочетании)	
Время установления выходного сигнала при изменении от 0 до максимального значения, ms , не более	2	
Пределы допускаемой приведенной погрешности, %: -в рабочих условиях эксплуатации (18-35) $^{\circ}C$ в предельных условиях эксплуатации (5-50) $^{\circ}C$	$\pm 0,4$ от I_d $\pm 0,8$ от I_d I_d – верхнее значение диапазона выходного сигнала, mA	
Номинальная функция преобразования	$N_{\text{вых}} = \frac{I_{\text{вх}}}{I_d} \cdot 4095$, где $N_{\text{вых}}$ – десятичное значение выходного кода; $I_{\text{вх}}$ – значение входного сигнала, mA	
Формат выходного кода: знак информативные разряды неинформативные разряды	15 3-14 0-2	
Вид гальванической развязки	Поканальная	
Испытательное напряжение гальванической развязки, не менее, V	1000	
Время вывода информации по ИР, μs , не более	10	

Каждый из $U1-U8$ содержит два независимых идентичных канала. Каждый из каналов включает в себя аналоговую память и преобразователь напряжение – ток. ЦАП включает: преобразователь двоичного кода в ток, усилитель и источник опорного напряжения.

Запись цифрового кода двумя байтами с выходов 0-7 порта РА микроконтроллера в ЦАП производится по сигналу WR, поступающему от МК. Подключение выхода ЦАП к входам $U1-U8$ осуществляется через коммутаторы К1-К4, управляемые Дш с задержкой 10 μs , необходимой для установления сигнала на выходе ЦАП.

Управление чтением и записью данных в ОЗУ осуществляется с помощью СУ (на основе ПЛИС), устанавливающей очередность обслуживания запросов доступа к ОЗУ от БУ1-БУ3 и МК.

Каждый из трех БУ1-БУ3 осуществляет подключение формирователя к шинам данных Д00-Д15 интерфейса ИР. СУ управляет формирователями и передатчиками БУ1-БУ3.

При записи данных МК устанавливает на шинах адрес-данные первый байт данных, сопровождая его низким уровнем сигнала, поступающего с выхода МК на контакт WR микросхемы СУ. Первый байт данных фиксируется в младших разрядах регистра данных СУ. Аналогично МК выдает второй байт данных, который фиксируется в старших разрядах регистра данных СУ. После записи второго байта СУ выдает данные на шины данных ОЗУ и формирует сигнал записи данных в ОЗУ WR.

При чтении данных из ОЗУ МК стробирует чтение байта данных низким уровнем сигнала RD. Первым читается младший байт данных. По завершении операции записи-чтения МК снимет сигнал VBR0.

МК осуществляет мажорирование и коррекцию данных, поступающих от трех КМп, а также калибровку выходных каналов формирователя.

Режим калибровки инициируется наличием логической единицы в разряде Д06 адресного слова, поступающего от любого КМп. При этом СУ формирует сигнал прерывания для МК. По сигналу прерывания МК входит в программу калибровки канала, адрес которого указан при адресации формирователя.

При калибровке МК осуществляет:

- формирование на выходе адресованного канала сигнала, близкого к предельному значению;
- ввод действительного значения выходного сигнала калибруемого канала через канал аналого-цифрового преобразования, подключенного к выходу калибруемого канала;
- расчет и занесение в FLASH по адресу калибруемого канала масштабного коэффициента.

Первым калибруется канал аналого-цифрового преобразования, на вход которого должно быть подключено эталонное напряжение +2 V.

АЦП служит для автоматизации процесса калибровки входных каналов формирователя. АЦП осуществляет преобразование сигналов напряжения постоянного тока в диапазоне (0-2,5) V в 16-разрядный двоичный дополнительный код.

АЦП работает в автоматическом режиме с циклом преобразования около 160 ms.

МК вводит данные по последовательному порту SCK по сигналу готовности SD.

ПН осуществляет преобразование суммарного напряжения источников питания +24 и минус 24 V в постоянные напряжения +15 и минус 15 V, а также формирует переменное напряжение питания 24 V для питания выходных усилителей U1-U8. Модуляция напряжений источников производится с частотой 16 kHz.

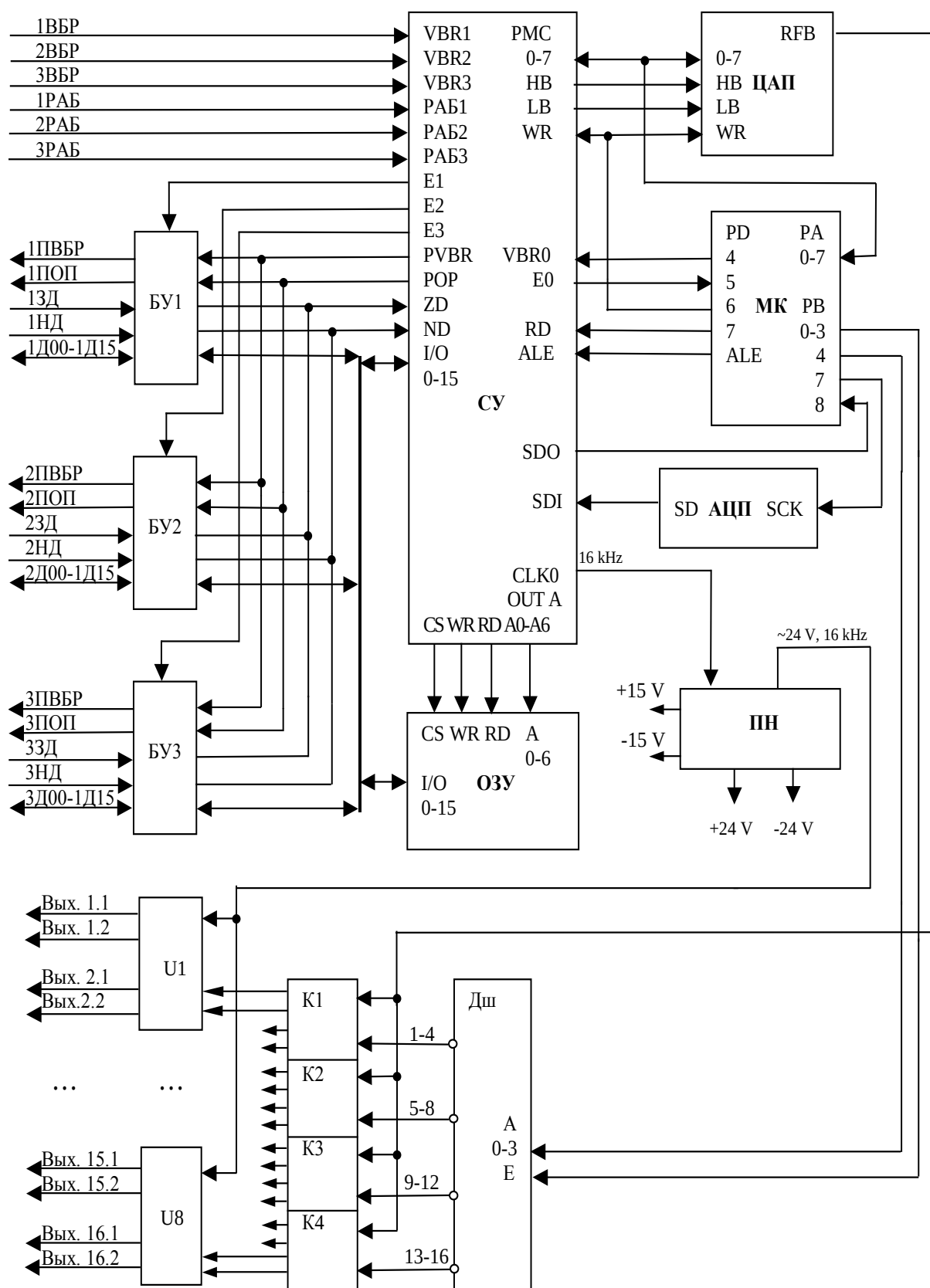


Рисунок Д.28 – Схема электрическая структурная ФТ-16

Д.3.11 Блоки вывода дискретных сигналов тока

Таблица Д17

Характеристики каналов вывода дискретных сигналов
при использовании ФД(Р)-16/3, ФК-16/3

Наименование показателя и единица измерения	Значения параметров	
	ФД(Р)-16/3	ФК-16/3
Число выходных каналов, шт.	16	
Тип выходного элемента в каждом канале	Бесконтактный ключ	Переключающий контакт электромагнитного реле
Параметры выходных каналов: -коммутируемое напряжение нагрузки не более, V	48	50
-коммутируемый ток нагрузки не более, A	0,2	1,0
Время переключения при активной нагрузке не более, ms	0,05	5
Время вывода информации по ИР не более μs	10	
Вид гальванической развязки	Поканальная	
Испытательное напряжение гальванической развязки не менее, V	1000	

Формирователь дискретных сигналов ФД(Р)-16. Предназначен для приема, запоминания 16-разрядного параллельного двухпозиционного кода, поступающего от трех КМп по интерфейсу ИР, и выдачи управляющих воздействий на исполнительные устройства постоянного тока. Формирователь содержит (рис. Д.29):

- регистры Rг1, Rг2, Rг3 для хранения принятой информации по трем независимым направлениям;
- узлы управления приемом и выдачей информации УУ1, УУ, УУ3;
- узлы выходных клапанов К1, К2, К3 для передачи в КМп по трем независимым направлениям информации о состоянии выходных каналов;
- узел мажорирования УМ для передачи достоверной информации на выходные ключи формирователя;
- узлы гальванической развязки УГр.1 и УГр.2 для гальванического разделения выходных каналов между собой и относительно цепей управления интерфейса ИР;
- узел выходных бесконтактных ключей Кл для управления внешней нагрузкой.

Формирователь обеспечивает защиту выходных ключей от перенапряжений при подключении к выходам активно-индуктивной нагрузки. Формирователь обеспечивает предпочтительное состояние выходных ключей после включения электропитания. Предпочтительное состояние (замкнутое или разомкнутое состояние всех выходных ключей) определяется пользователем и устанавливается с помощью переключателей.

Технические характеристики: количество каналов – 16, коммутируемый ток - не более 0,2 А, коммутируемое напряжение – от 5 до 48 V, время переключения – не более 50 μs , ток утечки (закрытого канала) – не более 0,5 mA. Остаточное напряжение (открытого канала) – не более 2 V. Испытательное напряжение гальванической развязки – 1500 V.

Формирователь контактный ФК-16/3. Предназначен для приема, запоминания 16-разрядного двухпозиционных кодов, поступающих от трех КМп по интерфейсу ИР, и выдачи управляющих воздействий по мажоритарному принципу на исполнительные устройства постоянного и переменного тока. Формирователь содержит (рис. Д.30):

- регистры Rг1, Rг2, Rг3 для хранения принятой информации от ВК по трем независимым направлениям;
- узлы управления приемом и выдачей информации УУ1, УУ2, УУ3;
- узлы выходных клапанов К1, К2, К3 для передачи в КМп по трем независимым направлениям информации о состоянии выходных каналов;
- узел мажорирования УМ для передачи достоверной информации на выходные реле формирователя;
- узел выходных реле УР для коммутации электрических цепей постоянного и переменного тока.

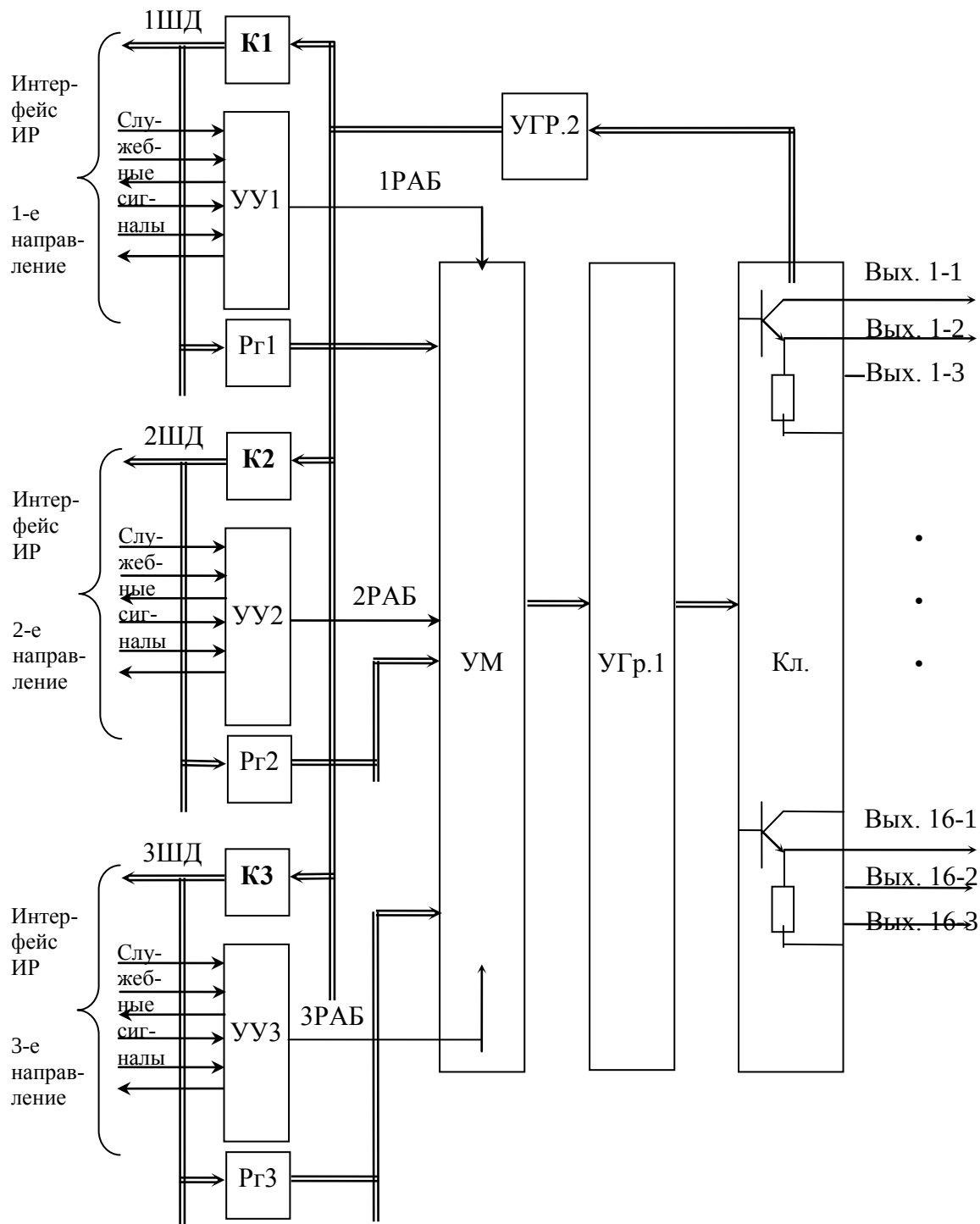


Рис. Д.29. Структурная схема формирователя дискретных сигналов ФД(Р)-16/3.

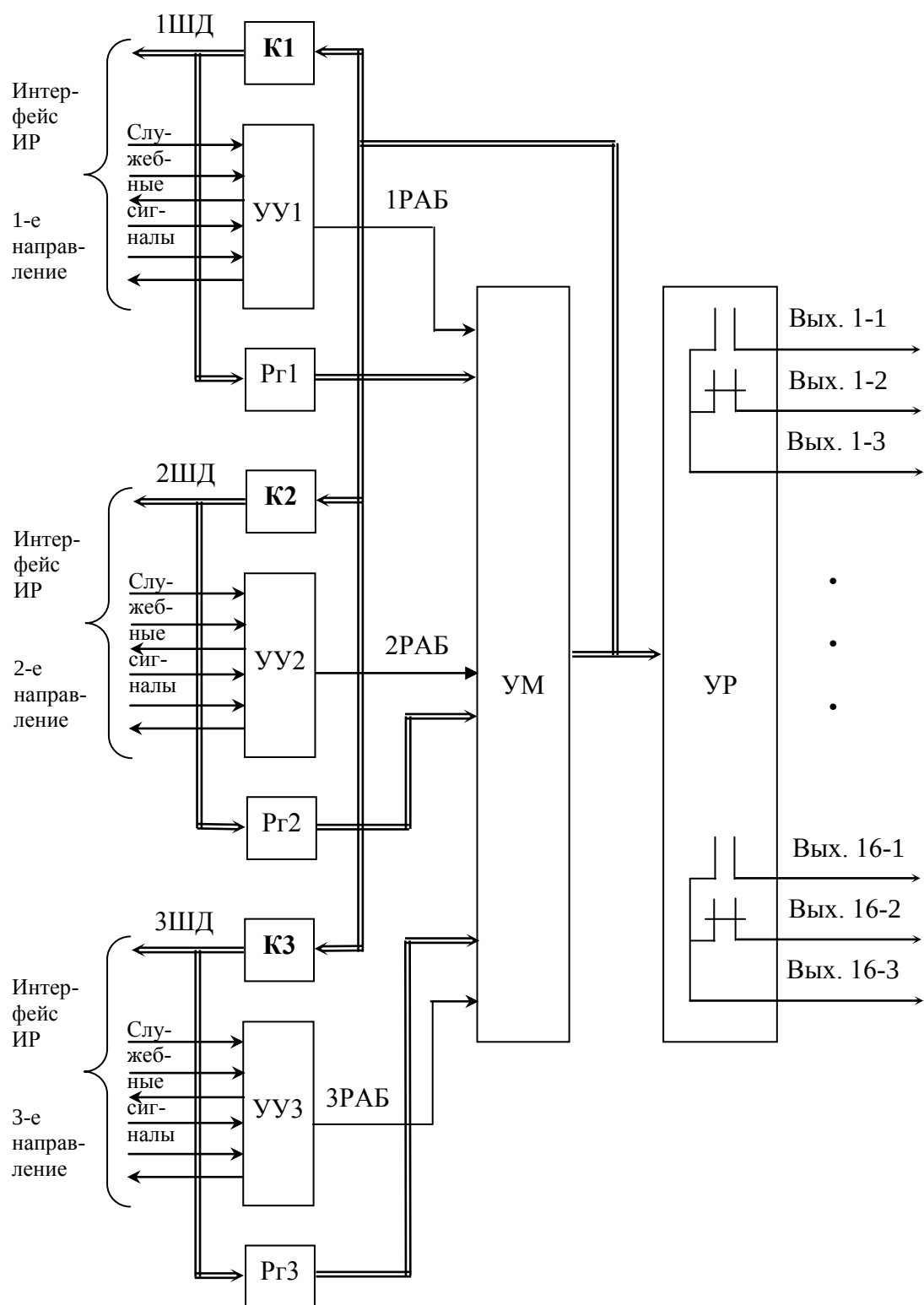


Рис. Д.30. Структурная схема формирователя контактного ФК-16/3.

Выход формирователя – переключающий контакт электромагнитного реле РЭС55А.

Исходное состояние всех реле после включения электропитания (включенное или выключенное) устанавливается после включения электропитания и определяется пользованием с помощью перемычек в блоке. Количество каналов – 16.

Технические характеристики. Максимальный коммутируемый ток - 1 А; максимальное коммутируемое напряжение постоянного и переменного тока - 50 В; максимальная коммутируемая мощность - 50 Вт; максимальная частота коммутации реле - 0,05 Hz.

Д.3.12. Адресация каналов в функциональных блоках связи с объектом

Каждый функциональный блок имеет выход на три магистрали интерфейса ИР и равнодоступен всем контроллерам МСКУ. Допустимое количество функциональных блоков, устанавливаемых в МСКУ, определяется исполнением МСКУ. Блоки идентифицируются свободными адресами и принимают значения от 0 до 31. Каналы в блоке идентифицируются своими физическими адресами, структура адреса в памяти МСКУ имеет вид, показанный на рис. Д.31.

15	12	11	7	6	1	0
Код функции	Адрес блока на интерфейсе ИР			Адрес канала в блоке или группы каналов для дискретных блоков		0

Рис. Д.31. Структура адреса в памяти МСКУ.

Для дискретных каналов ввода-вывода группа из 16 каналов имеет общий адрес в блоке, а каждый канал имеет адрес в группе, который определяет номер бита в информационном слове. Группа дискретных каналов неделима с точки зрения организации процессов ввода-вывода и обработки. Другие функциональные блоки имеют индивидуальный физический адрес и операции над ними выполняются независимо друг от друга. Адреса групп каналов дискретных блоков могут принимать от 0 до (n-1), где n - количество групп дискретных каналов в блоке.

Управление релейными блоками БР32-8, не имеющими выхода на ИР, осуществляется через блок ФД(Р)-16. Максимальное число БР32-8, подключаемых к ФД(Р)-16, восемь, т.е. адрес группы из 8 дискретных выходов одного БР32-8 может изменяться от 0 до 7.

Преобразователи ПНК(И)-30 имеют на каждую из 16 входов один контрольный вход, которому также присваивается адрес. Таким образом, рабочие каналы в преобразователе имеют адреса 0-14, 16-30, а контрольные - 15 и 31. Преобразователи ПТП(П)-8, ПТС(П)-8 имеют по 8 каналов: для каждого канала задействовано два адреса - по четному адресу читается сумма входного и контрольного сигнала, а по нечетному - разность входного и контрольного сигнала. В качестве адреса канала условно принимается четный адрес, т.е. каналы имеют адреса 0, 2, ...14. В качестве текущего значения на канале передается сумма значений, прочитанных по четному и нечетному адресу. Преобразователь ПКИ-2 обеспечивает преобразование сигналов от 7 датчиков прямой зарядки. Для каждого датчика имеется четыре измерительных канала: 1 - канал токового сигнала от 0 до 5 мА, адреса каналов - 32-38; 2 - канал фоновой составляющей токового сигнала от 0 до 5 мА, адреса каналов - 24-30; 3 - канал измерения проводимости изоляции по токовому сигналу, адреса каналов - 16-22; 4 - канал измерения проводимости изоляции по каналу фоновой составляющей сигнала, адреса каналов - 8-14.

Преобразователь дополнительно формирует коэффициенты передачи по каналам измерения, рассматриваемым как логические каналы. Адреса каналов - 0-6. Преобразование в ПКИ-2 имеет циклический характер, запуск очередного цикла преобразования осуществляется программным способом. В каждом цикле осуществляется преобразование основных сигналов (0-5) мА. Все остальные параметры являются вспомогательными и преобразовываются один раз в 40 циклов. Преобразователь ПКИ-3 обеспечивает преобразование в цифровой код сигналов от температурных датчиков:

- от 6 термоэлектрических преобразователей ± 15 мВ (адреса каналов - 0, 2, 4, 8, 10, 12), ± 30 мВ (адреса каналов - 16, 18, 20, 24, 26, 28);
- от 2 термопреобразователей сопротивления (30-50-70) Ω (адреса каналов - 6, 14), (10-50-90) Ω (адреса каналов - 22, 30).

Выбор диапазона осуществляется чтением каналов с адресами от 0 до 15 или от 16 до 31. Одновременная работа по двум диапазонам не допускается.

Физические адреса блоков и каналов задаются в программе генерации управляющей системы, включаются в состав базы данных МСКУ и доступны по запросам от программ текущего МСКУ и внешних абонентов МСКУ.

Д.3.13 Организация каналов связи с объектом

Каждый канал связи с объектом komponуется в следующем составе (со стороны подключения объектовых кабелей): панель соединительная ПСд; системный жгут; блок (преобразователь) связи с объектом; контроллер микропроцессорный.

Блок связи с объектом является основным элементом преобразования информации МСКУ. Связь блоков с кроссовым оборудованием (ПСд) осуществляется с помощью системных жгутов. Жгуты могут соединять от одного до четырех блоков с одной панелью (ПСд). Панели ПСд обеспечивают переход от объектовых кабелей к системным жгутам. Панели выполнены на печатных платах с установленными на них кроссовыми соединителями для подключения жил объектовых кабелей и разъемов для подсоединения системных жгутов. Одна панель может принять до шестнадцати двухпроводных линий связи.

Точки подключения к каждой панели промаркированы на лицевой панели сквозной нумерацией от 1 до 32. Подключение кабелей от объекта осуществляется последовательно снизу вверх для каждого датчика и исполнительного механизма соответственно двумя, тремя или четырьмя проводами. Распределения каналов в панели и системных жгутах кратно числам 4, 8, 16.

Д.4 Выносные блоки и приборы связи с объектом

Выносное оборудование, подключаемое к МСКУ через блоки ввода-вывода аналоговых и дискретных сигналов, расширяет функциональные возможности МСКУ, значительно сокращает количество линий связи, обеспечивает уменьшение затрат на монтажные и пуско-наладочные работы. В состав данной группы оборудования входят: выносные блоки ключей, обеспечивающие управление исполнительными механизмами повышенной мощности, различные коммутаторы входных и выходных дискретных сигналов, выносные измерительные преобразователи аналоговых сигналов и др. Выносное оборудование, как правило, выполнено в защищенных конструктивах для навесного и щитового монтажа.

Д.4.1 Блоки ключей

Блоки ключей предназначены для коммутации цепей постоянного и переменного тока и используются для управления исполнительными механизмами и установками повышенной мощности. В состав данной группы входят несколько модификаций ключей, которые выполнены в виде выносных приборов для навесного монтажа. Управление блоками ключей осуществляется через каналы вывода дискретных сигналов с использованием блока связи с объектом ФД(Р)-16, управление блоками усиления мощности БУМщ - через каналы вывода аналоговых сигналов с использованием блока связи с объектом ФН(Р)-4. Управление блоками ключей производится от блока управления БУ-403.

Выходные каналы блоков ключей имеют схемы защиты от перегрузки и схемы подавления электрического шума, возникающего при коммутации индуктивной нагрузки постоянного и переменного тока. Блок БКл-82 содержит схемные узлы, обеспечивающие автоматическое восстановление исходного состояния после снятия перегрузки, формирует и передает в МСКУ информацию о своем состоянии.

Д.4.2 Блок усилителей мощности

Для управления аналоговыми исполнительными механизмами в состав группы ключей входит блок усилителей мощности БУМщ-2. Он предназначен для управления и преобразования входного аналогового сигнала напряжения в сигнал напряжения с повышенной нагрузочной способностью. Количество каналов - 2. Диапазон изменения входного сигнала - 0-10 V. Максимальный входной ток - 5 mA. Диапазон изменения выходного сигнала - 0-20 V. Максимальный выходной ток в одном канале - 3 A.

Внешний вид выносных блоков показан на рис. Д.32.

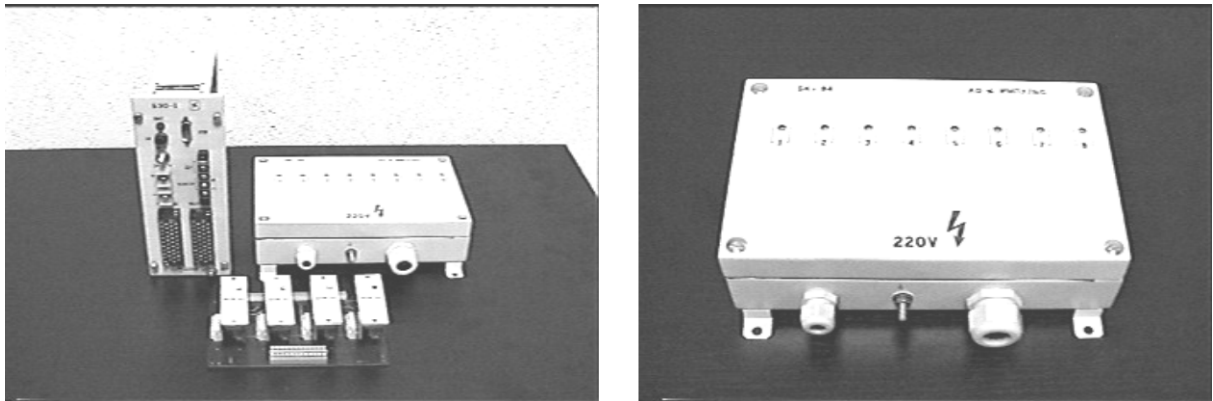


Рис. Д.29. Внешний вид выносных блоков

Напряжение питания блока - до 27 V. Управление производится от блока ФН(Р)-4/3. Содержит схему контроля значения выходного сигнала. Выходной контрольный сигнал - (0-10 V).

Д.5 Модуль контроля оборудования МКО-1

Модуль предназначен для выполнения функций контроля внутришкафного оборудования МСКУ, формирования синхросигнала интерфейса ИР, ведения времени, измерения температуры окружающего воздуха, формирования световых сигналов индикации, отражающих состояние оборудования шкафа. Основные параметры модуля соответствуют таблице Д.18. Модуль принимает по каждому направлению ИР:

- а) периодическую команду перезапуска, интервал – не более 1 s;
- б) команду управления состоянием программного флага «Нарушение».

Модуль обеспечивает:

- 1) при подключении к сети единого времени:
 - 1.1) прием и преобразование последовательного двоично-десятичного кода времени и даты в параллельный код с дискретностью 1 s;
 - 1.2) формирование параллельного двоичного кода времени в диапазоне от 0 до 1 s с дискретностью 1 μ s от внутреннего генератора. Частота внутреннего генератора (10000 $\pm$ 0,05) kHz;
 - 1.3) обнаружение наличия приема кодовой последовательности;
- 2) в автономном режиме:
 - 2.1) текущий счет времени с погрешностью не более ± 1 s в сутки;
 - 2.2) обнаружение отсутствия приема кодовой последовательности

Модуль формирует периодический логический сигнал на шине «Синхросигнал С1» ИР, временные характеристики которого находятся в пределах: период (1677,7 $\pm$ 0,5) ms; длительность логической «1» (104,86 $\pm$ 0,1) ms. Модуль обеспечивает смену цвета свечения на передней панели с зеленого на красный при выходе контролируемых сигналов за допустимые

пределы их изменения. Модуль состоит из следующих функциональных узлов (рис. Д.32): тактового генератора ТГ, узла синхронизации УС, узла приема кодовой последовательности УПКП, узла индикации УИ, реле твердотельного РТ, узла преобразования аналоговых сигналов УПАС, узла управления УУ, сторожевого таймера СТ, энергонезависимых часов RTC, узла гальванического разделения УГР, узла загрузки УЗ, шинных формирователей ШФ1 – ШФ3.

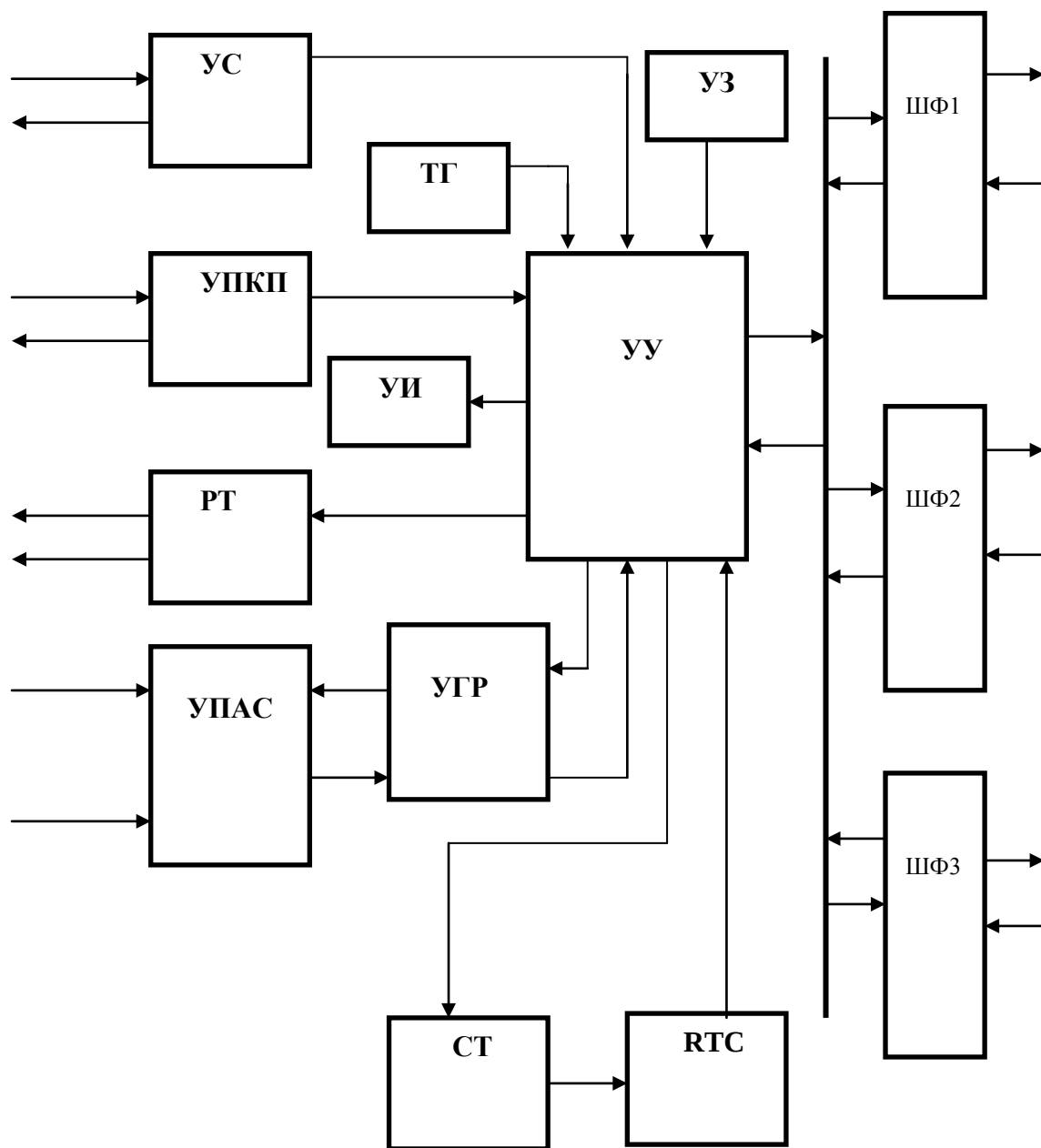


Рис. Д.32. Схема электрическая структурная МКО-1.

Тактовый генератор ТГ обеспечивает синхронную работу всех узлов модуля, а также формирование кода времени с дискретностью $1 \mu s$.

Узел синхронизации УС обеспечивает формирование периодического сигнала «Синхросигнал С1» ИР для организации работы двух шкафов МСКУ в дублированном режиме.

Узел приема кодовой последовательности УПКП обеспечивает синхронизацию шкалы времени модуля сигналами от синхронметра. УПКП содержит:

- схему для приема кодовой последовательности от синхронметра;

- ПЛИС, в которой реализована функция преобразования кодовой последовательности в параллельный двоичный код;
- фильтр низких для подавления помех и выделения полезного сигнала;
- схему формирования сигнала прерывания ПР интерфейса ИР.

Таблица Д.18

Основные параметры МКО-1

Наименование параметра	Значение параметра
1 Интерфейс связи	ИР
2 Количество направлений	3
3 Количество входных аналоговых каналов преобразования:	
- напряжения постоянного тока;	10
- температуры окружающей среды;	1
- контрольных сигналов	4
4 Выходной код	16-разрядный двоичный

Доступ к информации о текущей дате и времени возможен с любого из трех направлений ИР.

Узел индикации УИ обеспечивает двухцветную индикацию (красный – нарушение, зеленый – нормальная работа) состояния оборудования шкафа.

Реле твердотельное РТ обеспечивает формирование выходного сигнала «Работа шкафа».

Узел преобразования аналоговых сигналов УПАС обеспечивает прием входных контрольных сигналов и преобразования их с помощью АЦП в 16 - разрядный дополнительный двоичный код. Узел осуществляет контроль работоспособности источников питания шкафа МСКУ, источников питания вентиляторов, измеряет температуру в шкафу и определяет выход ее за 60 °С, контролирует работоспособность вентиляторов и состояние двери шкафа (открыта/закрыта), а также осуществляет контроль работоспособности самого модуля. Узел также осуществляет сравнение полученного кода с допустимыми пределами изменения контролируемого сигнала и в случае отклонения отображает его в словах состояния СС1 и СС2 (Таблицы Д.19, Д.20).

Узел управления УУ обеспечивает управление функционированием всех узлов модуля. УУ содержит микропроцессор и ПЛИС. В ПЛИС размещено ОЗУ на 32 шестнадцатиразрядных слова, арбитр – для поочередного доступа трех направлений ИР и микропроцессора в ОЗУ. Реализованы функции: приема и преобразования кодовой последовательности от синхронизатора; формирования периодического сигнала на шине ИР «Синхросигнал С1»; управления работой УИ.

Сторожевой таймер СТ обеспечивает организацию питания RTC и формирование выходного сигнала «Работа шкафа». К входу СТ подключен элемент питания для организации бесперебойного электропитания RTC при пропадании основного электропитания модуля +5 V. СТ отслеживает снижение напряжения за уровень ниже 2,1 V, формируя при этом соответствующий сигнал, который поступает в УУ. УУ устанавливает бит в СС1 «напряжение ниже 2,1 V – необходима замена элемента питания». При своевременном приеме периодической команды перезапуска (с интервалом перезапуска не более 1 s) формируется соответствующий сигнал, который совместно со схемой, расположенной в ПЛИС, формирует выходной сигнал «Работа шкафа» в РТ. Выходной сигнал становится пассивным (состояние «разомкнуто» на РТ) при отсутствии всех трех сигналов РАБ1, РАБ2 и РАБ3 интерфейса ИР независимо от приема периодической команды перезапуска. Выходной сигнал также становится пассивным, если ни одно направление ИР не выдает периодическую команду перезапуска независимо от состояния сигналов РАБ1, РАБ2 и РАБ3.

Энергонезависимые часы RTC обеспечивают счет времени и даты. RTC реализованы на интегральной микросхеме, реализующей функции часов. Точность хода RTC от минус 10 до 2 с в сутки. Значения регистров RTC доступны со стороны ИР. Есть возможность корректировки времени или даты.

Узел гальванического разделения УГР обеспечивает гальваническое отделение цепей УПАС от цепей ИР.

Узел загрузки УЗ обеспечивает загрузку содержимого ПЛИС после включения напряжения питания модуля и загрузку микропрограмм в микроконтроллер УУ.

Шинные формирователи ШФ1 – ШФ3 обеспечивают выдачу и прием информации по трем направлениям ИР.

РТ обеспечивает гальваническую развязку и формирование сигнала «Работа шкафа» под управлением ПЛИС.

Примечание - ГТ - готовность блока, логическая «1» в этом разряде означает, что блок приступил к контролю параметров, ему можно выдавать необходимые команды, считывать результаты контроля;

Таблица Д.19

Структура СС1

Шина ИР	Назначение
0	БПН1, -24 V
1	БПН1, +24 V
2	БПН1, +5 V
3	БПН1 - ВН
4	БПН2 - ВН
5	Ошибка ОЗУ
6	Резерв
7	Установлен программный флаг “Нарушение”
8	БПН2, -24 V
9	БПН2, +24 V
10	БПН2, +5 V
11	Напряжение электропитания RTC ниже 2,1 V (необходима замена элемента питания GB1 на новый)
12	Резерв
13	БВН
14	ДВ
15	Температура в шкафу МСКУ больше 60 °C

Таблица Д.20

Структура СС2

Шина ИР										
15	14	13 - 10	9	8	7	6	5	4	3	2 - 0
ГТ	Ош. t	Резерв	К	Т	GND-II	Уоп-II	АЦП	GND-I	Уоп-I	РР

К - поправочный коэффициент настройки конца шкалы АЦП в УПАС, логическая «1» в этом разряде означает, что коэффициент не настроен; Ош.t - неработоспособность датчика температуры модуля (логическая «1»); Т - результат завершения тестового режима. Логическая «1» в этом разряде соответствует нормальному завершению теста проверки; GND-II, Уоп.-II – неработоспособность второй группы аналоговых входов (каналы 8 - 15) или АЦП (логическая «1»); АЦП - неработоспособность АЦП модуля (логическая «1»); GND-I, Уоп.-I –

неработоспособность первой группы аналоговых входов (каналы 0 - 7) или АЦП (логическая «1»); РР – код режима работы, который может принимать следующие значения:

- 000 - исходное состояние, ГТ в состоянии логической «1»;
- 001 - режим установки времени для RTC;
- 010 - режим настройки АЦП (для работы на пульте);
- 011 - режим тестирования модуля по контрольной задаче. В этом режиме модуль выполняет тестовую задачу до тех пор, пока контрольная задача не восстановит его исходное состояние. Результатом выполнения процедуры тестирования является сообщение в разряде 8 ячейки СС2 (Т);
- 100 - режим наладки модуля на пульте;
- 101 - режим приведения модуля в исходное состояние. Модуль устанавливает ГТ - в состояние логического «0», после чего тестирует ОЗУ, проводит инициализацию схемных и программных средств. Наличие ошибок при выполнении тестов памяти фиксируется в разряде 5 СС1;
- 110 - режим проверки ОЗУ со стороны КМп.

Литература

1. Стефани Е.П. Основы построения АСУ ТП: Учебное пособие для вузов. - М.: Энергоиздат, 1982.-352.
2. Вальков В.М., Вершинин В.Е. Автоматизированные системы управления технологическими процессами. – Л.: Политехника, 1991.-269.
3. Анисимов Д.Н. Использование нечеткой логики в системах автоматического управления. М.: Приборы и системы. Управление. Контроль. Диагностика. 2001.№8.
4. Артюх С.Ф., Дуэль М.А., Шелепов И.Г. Автоматизированные системы управления энергогенерирующими установками электростанций. Харьков. «Знание», 2000
5. Дуэль М.А. Автоматическое управление блочными энергоустановками с применением средств вычислительной техники. М.: Энергоатомиздат. 1983.
6. Елисеев В.В., Пивоваров Г.Ю. и др. Микропроцессорная система контроля и управления МСКУ М. Приборы и системы управления, Москва, 1994, №9.
7. Елисеев В.В., Пивоваров Г.Ю. и др. Перспективы развития микропроцессорной системы контроля и управления МСКУ М. ПСУ, Москва, 1996, №10.
8. Колотов Ю.Н., Минчев Т.Ц., Баршунин В.В. Опыт внедрения ИВС технологического контроля на энергоблоке №3 с реактором ВВЭР-440 Кольской АЭС. Приборы и системы управления, 1994, №6.
9. Белоусова Р.И., Жамойдин А.А. и др. Информационно-вычислительная система для открытых распределительных устройств. Приборы и системы управления, 1997, №6.
10. Горелик А.Х., Хаит Я.Г. и др. АСУ энергоблоками тепловых и атомных электростанций на базе средств МСКУ М. Приборы и системы управления, 1996, №10.
11. Щербашин Ю.Д., Лукоянов Б.Е. и др. Автоматизированная система управления нагревательными печами нефтеперегонной установки на Кременчугском НПЗ. Приборы и системы управления, 1997, №6.
12. Грыцив П.Е. АСУ ТП сахарных заводов на базе комплексов МСКУ М. Приборы и системы управления, 1997, №7.
13. Дуэль М.А. и др. АСУ тепловой электростанции, Киев, «Техніка», 1977.
14. Клаассен К. Б. Основы измерений. Электронные методы и приборы в измерительной технике. – М.: Изд-во «Постмаркет», 2000.
15. Земельман М. А. Метрологические основы технических измерений. – М.: Изд-во стандартов, 1991. –228 с.
16. Маликов М. Ф. Основы метрологии.- М.: Комитет по делам мер и измерительных приборов при Совете Министров СССР, 1949 г.-477с.
17. Земельман М. А., Тронова И. М. Методический материал по применению ГОСТ 8.009-84. ГСИ. Нормируемые метрологические характеристики средств измерений. – М.: Изд-во стандартов.
18. МИ 2002-89. Системы информационно-измерительные. Организация и порядок проведения метрологической аттестации. М.: Изд-во стандартов, 1991.
19. www.schneider-electric.com.
20. Каталог СА01. Автоматизация и приводы. М: Сименс.2001.
21. Новости от фирмы Advantec. М: Промышленные АСУ и контроллеры. №9, 2001.
22. www.elpribor.euro.ru
23. Каталог фирмы PROSOFT, 1998.
24. Сорокин С. IBM PC в промышленности. - СТА, 1996, №1.
25. Гибко программируемые контроллеры или решения на базе ПК? Сименс. - СТА, 1998, №2.
26. Кривоносов А., Харитонов А., Гуличев В. Система управления газоперекачивающими агрегатами. - СТА, 1997. №2.
27. Годына В., Свищенко В. И др. АСУ ТП рудотермической электропечи для выплавки сплавов на основе кремния. - СТА, 1998, №1.

28. Сизов Н.Н., Гольдштейн А.М. и др. Новые возможности контроллера Ломиконт. Промышленные АСУ и контроллеры. №9, 2001
29. ADAM-5510 как зеркало современных тенденций автоматизации. М: СТА. №2, 1998
30. Чвилов Ю.А., Блинов В.В., Офштейн Б.М.. Микропроцессорные субкомплексы контроля и управления для АСУ промышленными объектами. - Приборы и системы управления. 1995, №2.
31. Комплексы управляющие вычислительные МСКУ 2. Техническое описание. СНПО "Импульс". Северодонецк. 2001г.
32. Архитектура сети МАПС. Техническое описание. АО "Импульс". Северодонецк. 1995.
33. КСв-31. Техническое описание. СНПО «Импульс». Северодонецк. 2000
34. Рабочие станции ПС5110. Техническое описание. НПО "Импульс". Северодонецк. 2001.
35. Елисеев В.В., Набатов А.С., Пивоваров Г.Ю. Рабочие станции ПС 5101. Москва. Приборы и системы управления, 1998, №2.
36. Сорокин С. Как много ОС РВ хороших. - СТА, 1997, №2.
37. IEEE Standart P1003.4
38. L. Sha, J. B. Goodenough. Real-Time Scheduling Theory and Ada. IEEE Computer, Vol. 23. No.4, April 1990.
39. Дейтел Г. Введение в операционные системы. Т. 1. - М : Мир, 1987. - 359.
40. QNX Operating System. System Architecture. - QNX Software Systems. Kanata. Ontario, 1996. - 142 с.
41. Прожогин С. Neutrino: быстрее, выше и меньше. – СТА, №1, 2000.
42. OS-9. System Architecture. - Microware System Corporation. USA. 1995.
43. Операционная система ОС5000. Техническое описание. – Северодонецк: АО «Импульс». 1996.
44. Richter J. Advanced Windows NT, Microsoft Press. Redmond, Washington, 1994.
45. Петерсон Р. LINUX. Справочник по операционной системе: - К.: Издательская группа BHV, 1997 - 688с.
46. Системы диспетчерского управления и сбора данных (SCADA- системы). МКА №3, 1999.
47. Шакиров С., Биусов Б. И др. ULTRALOGIC – система подготовки программ для промышленных контроллеров. – СТА, №3, 1997.
48. Спецификация COM. Версия 0.9, Microsoft Corporation. Октябрь 24, 1995.
49. The OPC Overview, OPC Fondation 1998.
50. The OPC Common Definitions and Interfaces, OPC Fondation 1998.
51. The OPC Data Acces Custom Specefications 2.3, OPC Fondation 1999.
52. Айзенберг А.Б., Макарова В. И. и др. Программное обеспечение системы МСКУ М. – Приборы и системы управления. 1995, №4.
53. Программное обеспечение микропроцессорных субкомплексов контроля и управления. Управляющая система. Руководство системного программиста. АО «Импульс». 2000.
54. Пакет программ для операторских станций КОРУНД. Техническое описание. АО «Импульс». 1995.
55. OPC сервер МСКУ М. Руководство по использованию. АО «Импульс». 2000.
56. Пакет программ доступа к оперативной базе данных комплекса МСКУ М. Руководство по использованию. АО «Импульс». 2000.
57. Козлов Б., Ушаков И. Справочник по расчету надежности аппаратуры радиоэлектроники и автоматики. М, «Советское радио». 1975.
58. Иыуду К.А. Надежность, контроль и диагностика вычислительных машин и систем. М. Высшая школа. 1989.
59. Коваленко А.Е., Гула В.В. Отказоустойчивые микропроцессорные системы. Киев. Техника.1986.

60. Додонов А.Г., Кузнецова М.Г., Горбачик Е.С. Введение в теорию живучести вычислительных систем. Киев: Наук. Думка, 1990
61. Гуляев В. А. Техническая диагностика управляющих систем. К.: Наук. Думка, 1983.
62. Набор технологических программ для МСКУ. Руководство по эксплуатации. Северодонецк. АО «Импульс». 1997.
63. Управляющая вычислительная система энергоблока ТЭС. Техничко-коммерческие предложения. ХИКА. Харьков. 1997.
64. Балашов Е.П., Пузанков Д.В. Проектирование информационно-управляющих систем. - М.: Радио и связь, 1987. - 256с.
65. Калянов Г.Н. CASE. Структурный системный анализ (автоматизация и применение). М., "Лори", 1996.
66. Марка Д.А., МакГоуэн К. Методология структурного анализа и проектирования. М., "МетаТехнология", 1993.
67. Шлеер С., Меллор С. Объектно-ориентированный анализ: моделирование мира в состояниях. Киев, "Диалектика", 1993.
68. Международные стандарты, поддерживающие жизненный цикл программных средств. М., МП "Экономика", 1996.
69. Юдицкий С. А., Барон Ю. А. И др. Схемно-ориентированная технология логического моделирования. М. ПСУ, №8, 1996.
70. Питерсон Дж. Теория сетей Петри и моделирование систем. М.: Мир. 1984.
71. Котов В. Е. Сети Петри. М.: Наука. 1984.
72. Раскин Л.Г. Анализ сложных систем и элементы теории оптимального управления. М.. «Сов. Радио», 1976.
73. Лифшиц А. Л., Мальц Э. А. Статистическое моделирование систем массового обслуживания. М.. «Сов. Радио», 1978.
74. Широчин В. П., Москальков А. М., Обейдат А-С. Оценка производительности распределенных систем на основе стохастических сетей Петри. Киев. Электронное моделирование. №4, 1994.
75. Елисеев В.В., Ларгин В.А., Макарова В.И., Пивоваров Г.Ю. Системы контроля и управления на базе МСКУ М для объектов тепловой и атомной энергетики. Промышленные АСУ и контроллеры. №6, 1999.
76. Елисеев В.В., Ларгин В.А., Пивоваров Г.Ю., Ященко В. И. МСКУ М на объектах газового комплекса. Промышленные АСУ и контроллеры. №9, 1999.
77. Лэм Г.. Аналоговые и цифровые фильтры. Расчет и реализация. Москва, "Мир", 1982.
78. Хемминг Р.В.. Цифровые фильтры. Москва, "Недра", 1987.
79. Фролов А.Г., Фролов В.Г. Защищенный режим процессоров Intel. – Москва. Диалог-МИФИ. 1993.

Содержание

Перечень сокращений	4
Введение	5
Глава 1 ОБЩАЯ ХАРАКТЕРИСТИКА ПТК АСУ ТП	11
1.1. Основные понятия и определения	11
1.2. Функции ПТК	13
1.2.1. Информационные функции	13
1.2.1.1 Сбор и первичная обработка аналоговой измерительной информации	14
1.2.1.2. Сбор и обработка дискретных сигналов	15
1.2.1.3 Сбор и обработка кодированных сигналов	15
1.2.1.4 Архивация	17
1.2.1.5 Расчет и анализ ТЭП	19
1.2.1.6 Реализация информационных функций	20
1.2.2 Управляющие функции ПТК	25
1.3 Виды обеспечения ПТК	28
1.3.1 Лингвистическое обеспечение	29
1.3.2 Информационное обеспечение	29
1.3.2.1 Архитектура информационной базы	30
1.3.3 Метрологическое обеспечение	32
1.4 Создание ПТК	42
Глава 2 ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПТК	45
2.1 Контроллеры	45
2.1.1 Программируемые логические контроллеры	45
2.2. Микропроцессорные управляющие вычислительные субкомплексы МСКУ	57
2.2.1 Структура МСКУ	58
2.2.2 Функциональные возможности МСКУ	62
2.2.3 Типы и основные характеристики МСКУ	62
2.2.4 Микропроцессорные контроллеры	63
2.2.5 Интерфейс ИП	64
2.3. Сети	64
2.3.1 PROFIBUS	65
2.3.2 Локальные сети МСКУ М	66
2.4 Технические средства для компоновки верхнего уровня ПТК	75
2.4.1 Рабочие станции ПС 5110	77
2.5 Посты управления и обслуживания	82
2.6 Организация электропитания и заземления ПТК	82
2.6.1 Организация электропитания	82
2.6.2. Защитное заземление	83
2.6.3 Рекомендации по повышению помехозащищенности измерительных цепей	83
Глава 3 БАЗОВОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПТК	93
3.1 Операционные системы	93
3.1.1 Стандартизация ОС РВ	94
3.1.1.1 Стандарт POSIX.4	94

3.1.1.2 <i>Threads-расширение</i>	96
3.1.1.3 <i>Дополнительные расширения стандарта POSIX.4b для приложений реального времени</i>	97
3.1.1.4. <i>Профили прикладных контекстов реального времени</i>	98
3.1.2 Основные свойства ОС RV	99
3.1.2.1. <i>Основные особенности задач систем реального времени</i>	99
3.1.2.2 <i>Планирование задач</i>	100
3.1.2.3 <i>Синхронизация задач</i>	102
3.1.3 Операционная система QNX	104
3.1.3.1. <i>Архитектура</i>	105
3.1.3.2. <i>Диспетчеризация процессов</i>	110
3.1.3.3. <i>Менеджер процессов</i>	111
3.1.3.4. <i>Менеджер файловой системы</i>	112
3.1.3.5. <i>Менеджер устройств</i>	114
3.1.3.6. <i>Менеджер сети</i>	116
3.1.4 Операционная система OS-9	118
3.1.5. Операционная система OC5000	123
3.1.5.1. <i>Управление задачами</i>	123
3.1.5.2. <i>Программы-администраторы ОС</i>	126
3.1.5.3. <i>Супервизор</i>	129
3.1.6 Операционная система WINDOWS NT	131
3.1.6.1. <i>Система управление памятью</i>	131
3.1.6.2. <i>Файловая система</i>	133
3.1.6.3. <i>Применение Windows NT в АСУ ТП</i>	134
3.1.7 Операционная система LINUX	134
3.1.7.1. <i>Файловая структура</i>	135
3.1.7.2. <i>Принципы работы с ОС LINUX</i>	136
3.2 SCADA – системы	138
3.3 Системы технологического программирования	143
3.3.1 Система ULTRALOGIC	145
3.4 Современные технологии для разработки базового ПО	145
3.4.1 OPC-стандарт	147
3.5 Программное обеспечение МСКУ М	152
3.5.1. Системное ПО МСКУ 2М	153
3.5.2 Средства программирования МСКУ 2	162
3.5.3 Системное ПО операторских и рабочих станций	163
Глава 4 ПРИНЦИПЫ ОРГАНИЗАЦИИ ОТКАЗОУСТОЙЧИВЫХ ПТК	174
4.1 Системные способы обеспечения отказоустойчивости	174
4.2 Методы организации отказоустойчивого программного обеспечения	176
4.3 Способы построения отказоустойчивых систем	177
4.4 Организация отказоустойчивых ПТК на базе МСКУ М	182
4.5 Контроль и диагностирование ПТК	195
4.5.1. Организация процессов диагностирования в отказоустойчивых ПТК	195
4.5.2. Организация диагностирования ПТК на базе МСКУ М	196
4.5.3. Контроль и диагностирование МСКУ	197

Глава 5 ОСНОВНЫЕ МЕТОДЫ ПРОЕКТИРОВАНИЯ И ИССЛЕДОВАНИЯ ПТК	204
5.1 Основные методы проектирования ПТК	204
5.1.1 Методологии и технологии проектирования ПТК	205
5.1.2 Структурный подход к проектированию ПТК	208
5.1.2.1. <i>Пример синтеза структуры ПТК учета электрической энергии</i>	209
5.1.2.2. <i>Синтез функциональной структуры технического обеспечения</i>	210
5.1.2.3. <i>Синтез функциональной структуры программного обеспечения</i>	212
5.2 Основные методы исследования ПТК	213
 Глава 6 ПРОГРАММНО-ТЕХНИЧЕСКИЕ КОМПЛЕКСЫ НА БАЗЕ МСКУ М	215
6.1 ПТК на объектах энергетики	215
6.1.1 ПТК АСУ ТП энергоблока 800 MW Запорожской ГРЭС	215
6.1.2 Информационно-вычислительная система технологического контроля энергоблока Кольской АЭС с реактором ВВЭР-440	216
6.1.3 Информационно-вычислительная система АСУ ТП ОРУ Курской АЭС	217
6.1.4 ПТК модернизированной системы внутриреакторного контроля реактора ВВЭР-1000 (ПТК СВРК-М)	218
6.1.5 ПТК информационно-вычислительной системы энергоблока с реактором ВВЭР-1000	219
6.2 ПТК на объектах газового комплекса	222
6.2.1 ПТК автоматизированных систем управления компрессорными станциями	222
6.2.2 ПТК системы автоматического управления ГПА-Ц6,3В	225
6.3 ПТК на объектах нефтехимии	228
6.3.1 АСУ ТП узла К-105, К-106/І, ІІ установки концентрирования пропилена на АО "УФАОРГСИНТЕЗ"	228
6.3.2 АСУ ТП нагревательных печей нефтеперегонной установки на АО «Укртатнефть»	229
6.4 ПТК на объектах пищевой промышленности	231
6.4.1 АСУ ТП основного производства сахарного завода	231
6.4.2 АСУ ТП ТЭЦ сахарного завода	232
6.5 Системы энергоучета	233
6.5.1 Автоматизированная система учёта и контроля электроэнергии на подстанции "Подземгаз 110/35" Лисичанского ПЭС	233
Приложение А Интерфейс резервированный (ИР)	237
Приложение Б Архитектура сети МАПС	247
Приложение В Некоторые аспекты первичной обработки измерительной информации	258
Приложение Г Архитектурные особенности IBM PC-контроллеров	266
Приложение Д Характеристики блоков МСКУ	272
Литература	332