



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)

Факультет	<u>Энергетика и нефтегазопромышленность</u>
Кафедра	<u>АММ НГК</u>
Направление	<u>15.03.04 Автоматизация технологических процессов и производств (бакалавриат)</u>

КОНСПЕКТ ЛЕКЦИЙ ПО ДИСЦИПЛИНЕ
«Программируемые контроллеры систем управления»

Ростов-на-Дону

2017

Лекция 1-2. Общая характеристика программируемого контроллера

Объектный программируемый контроллер воспринимается пользователем или потребителем как нечто цельное, имеющее внешние системотехнические атрибуты, заложенные в его архитектуре. То есть архитектура ОПК должна четко определять то, что при автоматизации конкретного технологического объекта может быть реализовано с помощью конкретного ОПК[12].

Таким образом, архитектура ОПК - это принципиальные возможности аппаратно - программных средств, которыми располагает ОПК для решения конкретных задач автоматизации.

Итак, ресурсы ОПК против перечня задач. В соответствии с опубликованной в 1990 году Национальной ассоциацией производителей электротехнического оборудования США (NEMA) "Концепцией классификации объектных контроллеров промышленного назначения и руководством по их выбору" под объектными понимаются ПК, обеспечивающие контроль хода технологического процесса, накопление данных об автоматизируемом объекте, поддержку систем программного управления, протоколирование и связь с оператором или другими системами автоматизации [3]. Для выполнения функций управления конкретным объектом ПК наделены необходимыми для этого средствами объектной ориентации. В общем случае область задач автоматизации технологического объекта перекрывается ресурсами ОПК, выделяемыми на объектную ориентацию (рис.1.1).

В процессе решения задач автоматизации технологического объекта пользователь, оценивая архитектуру ОПК, решает также оптимизационную задачу объектной ориентации ресурсов ОПК, минимизируя избыточные ресурсы.

Архитектура ОПК отражает следующие особенности его роли и места в системе автоматизации:

ОПК в соответствии с перечнем решаемых задач автоматизации находится в точке пересечения физических сигналов, идущих к или от технологического объекта, и команд (или информационных реакций), ими вызываемых или их вызывающих. Поэтому архитектура ОПК должна представлять возможности согласования физических "технологических" сигналов с их командной или информационной интерпретацией (в том числе по количеству, номенклатуре, типу, диапазону и т.п.).

ОПК решает задачи

управления

технологическим объектом

в режиме реального

времени, в то же время

взаимодействует с

оператором (или обмен

информацией с другими

системами) ведется вне

требований этого режима.

Архитектура ОПК должна

давать ясное представление

о возможностях

согласования этих

режимов, или о ресурсах ОПК,

которые могут быть выделены

на решение задач такого

согласования.

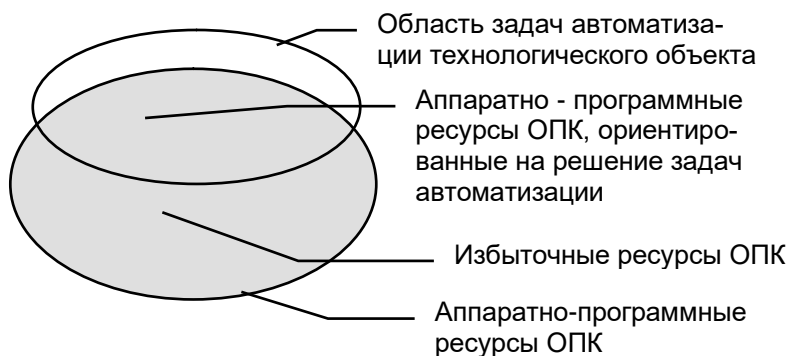


Рис.1.1 Графическая иллюстрация сочетания ресурсов ОПК и области задач автоматизации

ОПК - аппаратно - программное устройство, и поэтому эффективность объектной ориентации ОПК во многом определяется стратегией использования ресурсов аппаратных, программных или аппаратно - программных средств.

ОПК - цифровое вычислительное устройство, характеризующееся конкретными вычислительными возможностями при реализации тех или иных задач. Архитектура ОПК в этой части характеристик совпадает с понятием архитектуры ЭВМ.

ОПК - программируемое устройство, обладающее средствами программирования, как встроенными (резидентными), так и подключаемыми (внешними). Архитектура ОПК определяет принцип решения задач программирования (может быть различным для разных стадий жизненного цикла).

ОПК в силу своей объектной ориентации обладает изменяемой частью своих аппаратно - программных ресурсов и, соответственно, частью ресурсов (ядро) не изменяемой. Понятие ядра в ОПК чисто условное, и определяется не только архитектурными возможностями, но и принятой стратегией объектной ориентации.

ОПК в зависимости от архитектуры (а в итоге от достигнутой эффективности объектной ориентации) обладает той или иной степенью избыточности средств ориентации (аппаратных, программных, информационных и т.п.).

Неизменяемая часть (Ядро)				
Аппаратная			Программная	
Микроконтроллер: процессор, память, порты, таймер, системная магистраль			В IOS	Монитор реального времени
Интерфейсные адаптеры	Средства расширения		Драйверы	Средства программирования
Средства адаптации к физическим сигналам объекта			Исполняемый файл прикладной программы	
Изменяемая часть				
Средства адаптации к физическим сигналам объекта		Вычислительные и сетевые средства		Средства программного
Типы модулей		Адаптер	Интерфейс	Прикладные
Характеристики	Параметры	Стандарт		Спецификация

Рис.1.2. Обобщенная архитектура ОПК

На рис.1.2 представлена обобщенная архитектура ОПК [13], отражающая перечисленные выше особенности оценки его возможностей в процессе автоматизации технологических объектов. Представленная на рисунке архитектура позволяет оценивать возможности объектной ориентации контроллера исходя из предварительной декомпозиции задачи автоматизации. Неизменяемое ядро ОПК включает аппаратную и программную части. Основу аппаратной части составляют процессор, память, порты, таймер, системная магистраль - то есть микроЭВМ с присущими ей архитектурными особенностями. Дальнейшее наращивание возможностей аппаратных средств связано с использованием средств расширения и интерфейсных адаптеров, позволяющих дополнять архитектуру микроконтроллера новыми структурными элементами. Программная поддержка работы перечисленных аппаратных средств осуществляется с помощью резидентного загрузочного модуля (BIOS) и программы - монитора реального времени. Наличие программ драйверов и средств программирования в программном ядре ОПК позволяет реализовать программный доступ к аппаратным ресурсам с целью решения задач автоматизации.

В зависимости от решаемой задачи автоматизации ОПК может быть выполнен в виде неизменяемого ядра, встраиваемого в технологический объект. Встраивание ОПК наряду с оптимизацией режимов работы автоматизируемого технологического объекта преследует цель оптимизации стоимости ОПК. Технологический объект в течение всего жизненного цикла не изменяется, поэтому средства адаптации к физическим сигналам объекта и исполняемый файл прикладной программы управления объектом также остаются неизменными. На рисунке средства адаптации к физическим сигналам объекта и

исполняемый файл прикладной программы для такого случая внесены в состав ядра. Однако в общем случае архитектура ОПК имеет средства адаптации к физическим сигналам объекта и прикладные программы среди средств изменяемой части, наряду с вычислительными, сетевыми и средствами диалогового интерфейса. Средства адаптации к физическим сигналам объекта представлены в виде модулей соответствующих типов, организуемых в структуры с помощью средств развития и работающих в стандартах определенных интерфейсов. Вычислительные возможности ОПК в соответствии с функциональными требованиями наращиваются с помощью модулей памяти, сопроцессоров и т.п., сетевые средства для организации связи в среде системы автоматизации могут быть представлены соответствующими адаптерами. Средствами диалогового интерфейса организуется среда оператора на соответствующем уровне доступа, то есть в предметной среде соответствующего иерархического уровня с предоставлением возможности использования эффективных языка и инструмента. И, наконец, программные средства включают собственно прикладные программы, реализующие алгоритмы управления, в том числе, инструментальные средства, обеспечивающие программирование ОПК с помощью универсальных или специальных языков или средств программирования. Непосредственно количественные характеристики объектной ориентации ОПК оптимизируются на уровне характеристик, спецификаций и параметров соответствующих модулей, адаптеров и программ.

Итак, предметом дальнейшего рассмотрения является реализация (обзор, анализ, описание и оценка) электронных компонентов, аппаратуры, технологий программирования и, наконец, практических аспектов проектирования объектных программируемых контроллеров.

Лекция 3. Обзор и классификация микропроцессоров и микроконтроллеров

Микропроцессор (МП) или MPU (microprocessor unit) характеризуется очень большим числом параметров и качеств, поскольку он является, с одной стороны, сложным программно - управляемым цифровым процессором (устройством ЭВМ), а с другой - интегральной микросхемой (электронным прибором). Поэтому для микропроцессора важны такие качества и параметры, как тип корпуса, количество источников питания, требования к синхронизации, мощность рассеяния, температурный диапазон, возможность расширения разрядности, уровни сигналов на выходе, надежность, долговечность - то есть параметры технологические. К числу параметров, определяющих архитектуру МП, можно отнести: разрядность АЛУ; наличие и параметры аппаратных расширителей; емкость и организация адресуемой памяти; число уровней прерывания; тип и объем стековой памяти; способы адресации и система команд; тип внутренней и внешней магистрали; число портов ввода-вывода и их организацию.

По числу БИС в микропроцессорном наборе различают одно - и многокристальные микропроцессоры.

Однокристальные микропроцессоры получают при реализации всех аппаратурных средств процессора в виде одной БИС или СБИС. По мере увеличения степени интеграции элементов в кристалле и числа выводов корпуса параметры однокристалльных микропроцессоров улучшаются. Однако возможности однокристалльных микропроцессоров могут быть ограничены аппаратурными ресурсами кристалла и корпуса. Поэтому получили распространение многокристальные микропроцессоры, а также многокристальные микропроцессоры секционные.

Для получения *многокристального микропроцессора* необходимо провести разбиение его логической структуры на функционально законченные части и реализовать их в виде БИС или СБИС. Функциональная законченность БИС многокристального микропроцессора означает, что его части выполняют заранее определенные функции и могут работать

автономно, а для построения развитого процессора не требуется организации большого количества новых связей и каких-либо других электронных БИС.

Многокристалльные секционные микропроцессоры получаются в том случае, когда в виде БИС реализуются части (секции) логической структуры процессора при функциональном разбиении ее вертикальными плоскостями. Для построения многоразрядных микропроцессоров при параллельном включении секций БИС МП в них добавляются средства стыковки. Таким образом, микропроцессорная секция - это БИС, предназначенная для обработки нескольких разрядов данных или выполнения определенных управляющих операций. Секционность БИС МП определяет возможность наращивания разрядности обрабатываемых данных или усложнения устройств управления микропроцессором при параллельном включении большого числа БИС. За годы с момента создания однокристалльные микропроцессоры развились от простых специализированных 2-разрядных до 32-разрядных процессоров, сравнимых по параметрам с процессорами ЭВМ 80-х годов. Многокристалльные секционные микропроцессоры позволяют создавать высокопроизводительные процессоры большой разрядности. Как предельный случай секционирования можно рассматривать однородные вычислительные среды, например, Райта.

По назначению различают универсальные и специализированные микропроцессоры.

Универсальные микропроцессоры - это такие, в системе команд которых заложена алгоритмическая универсальность. Последнее означает, что выполняемый машиной состав команд позволяет получить преобразование информации в соответствии с любым заданным алгоритмом. К универсальным МП относятся и секционные, поскольку для них система команд может быть оптимизирована в каждом конкретном случае. Универсальные микропроцессоры могут быть применены для решения широкого круга задач. При этом их эффективная производительность слабо зависит от проблемной специфики задач.

Специализация МП, т.е. его ориентация на ускоренное выполнения определенных функций, позволяет увеличить эффективную производительность при решении только определенных задач, а иногда только для решения одной конкретной задачи. Их существенными особенностями являются простота управления, компактность аппаратных средств, низкая стоимость и малая мощность потребления. Среди специализированных МП можно выделить различные микроконтроллеры, ориентированные на выполнение сложных последовательностей операций: математические МП, предназначенные для повышения производительности при выполнении арифметических операций за счет, например, матричных методов их выполнения; МП для обработки данных в различных областях применения и т.д. С помощью специализированных МП можно эффективно решать новые сложные задачи параллельной обработки данных.

По характеру временной организации работы МП делят на синхронные и асинхронные. *Синхронные* МП - микропроцессоры, в которых начало и конец выполнения операций задаются устройством управления (время выполнения операций в этом случае не зависит от вида выполняемых команд и величин операндов). *Асинхронные* МП позволяют начало выполнения каждой последующей операции определять по сигналу фактического окончания выполнения предыдущей операции. Для более эффективного использования микропроцессорной системы, в состав ее асинхронно работающих устройств вводят электронные цепи. Закончив работу над какой-либо операцией, устройство вырабатывает сигнал запроса, означающий его готовность к выполнению следующей операции.

По языковым (командным) возможностям различают МП с полным набором команд, с ограниченным набором команд (RISC - архитектура) и с непосредственной реализацией языков программирования высокого уровня. *МП с полным набором команд* это обычно универсальные МП или микроконтроллеры. Набор команд однозначно определяется архитектурой, и в процессе функционирования команды имеют равные приоритеты исполнения. *RISC (reduced instruction set computer)* - компьютеры имеют архитектуру с

универсальным процессором, работающим с высокой скоростью. Благодаря сокращению набора команд упрощаются аппаратные схемы, а значит, обеспечивается оптимизация выполнения часто используемых команд. За счет применения большого числа регистров уменьшается частота обращений к памяти, что также позволяет повысить скорость выполнения команд. Благодаря применению микропрограмм облегчается описание команд, которые требуют выполнения сложных действий. Используя микропрограммы, можно так организовать управление МП, чтобы непосредственно на нем интерпретировать текст и выполнять программу, написанную на языке высокого уровня, например, Форт. Такие МП принято называть по типу реализуемого языка, то есть, например, *Форт - процессор*.

Краткий обзор МП

В современных МП в качестве основного элемента используются "полевые" транзисторы - транзисторы, изготовленные по МОП-технологии (металл - окисел - полупроводник). Полевые транзисторы могут быть р - канальными (р-МОП) или n - канальными (n-МОП) в зависимости от того, какой тип канала является проводящим. Схемы некоторых БИС построены из так называемых дополняющих пар канальных МОП-транзисторов р- и n- типа. Они носят название дополняющих или комплиментарных МОП-структур (КМОП-структур).

В схемах МП используются все три типа структур. Схемы на основе р-МОП наименее быстродействующие, но проще в изготовлении. Схемы на основе n-МОП и КМОП более распространены. Первые потребляют большую мощность, чем КМОП при близких характеристиках по быстродействию.

Обозначения серий и отдельных микросхем состоят из нескольких элементов. В России первый элемент - буква К - указывает на использование микросхем в устройствах широкого применения. Второй элемент также может быть буква, обозначающая тип корпуса. В микропроцессорной технике наиболее употребительны корпуса типа ДИП, обозначаемые буквами Р - пластмассовый и М - металлокерамический. Функциональные свойства микросхем определяются полным номером серии (три или четыре цифры) и следующим за ним буквенным обозначением типа микросхемы. Все серии микросхем, выпускавшиеся до 1981 г., обозначались тремя цифрами. Микропроцессорные БИС имели обозначение ИК - комбинированные схемы цифровых устройств, например, К581ИК1, К587ИК2 (цифры, стоящие после обозначения типа микросхемы, указывают на условный номер разработки данной серии по функциональному признаку). Серии микропроцессорных БИС, выпускаемые с 1981 г., имеют четырехзначный номер (1810, 1816 и т.п.). Введена также подгруппа схем вычислительных средств (первая буква В) для обозначения микросхем, входящих в состав МПК. Микропроцессор в этой подгруппе имеет обозначение ВМ, микропроцессорная секция - ВС, функциональный расширитель - ВР. Для некоторых серий в литературе встречаются как старые, так и новые обозначения типов микросхем, например: МП серии 580 обозначают К580ИК80 или КР580ВМ80, последовательный интерфейс - К580ИК51 или КР580ВВ51. В конце условного обозначения микросхем может добавляться буква, указывающая на различия электрических параметров, обусловленные технологическими особенностями. Так, БИС КР580ВМ80А является полным схемотехническим аналогом КР580ВМ80, но может работать с более высокой тактовой частотой.

Для БИС иностранных фирм - производителей принято указывать фирму, далее номер микросхемы, например, Intel 8031. Иногда для БИС используются и просто имена, например, Pentium, Alpha. В спецификациях, однако, применяются обозначения и классификация конкретной фирмы. Например, для микросхем фирмы Intel применяется система обозначения из четырех полей: X XX XXXXXXXXXXXXXXX XXXXXX. Первый однобуквенный префикс отражает температурный диапазон, далее одно- или двухбуквенный префикс указывает на вариант исполнения корпуса, далее до 15 буквенных и цифровых символов указывают на тип кристалла, и, наконец, до 6 буквенных и цифровых символов отражают различные особенности и варианты исполнения микросхемы.

В связи с широким спектром возможных применений эволюция архитектуры МП пошла по нескольким различным направлениям. Выпускаемые промышленностью микропроцессорные средства можно сгруппировать в универсальные микропроцессоры, однокристальные микроконтроллеры и специализированные микроЭВМ [1].

Среди **универсальных микропроцессоров** следует выделить комплекты однокристальных МП универсального назначения и секционированные. Эти микропроцессоры имеют архитектуру **фон Неймана**.

Микропроцессорный комплект БИС серии 580 (семейство Intel-архитектуры - Intel 8080) предназначен для создания широкого класса средств вычислительной техники и обработки информации. Этот однокристальный параллельный 8-разрядный микропроцессор с фиксированной системой команд, выполнен по n-МОП технологии и содержит 5000 транзисторов. МикроЭВМ на основе комплекта K580 строятся по модульному принципу, который обеспечивает подключение к единой магистрали модулей МП, ЗУ и УВВ. Для работы МП используется три источника питания, синхронизация от внешнего генератора, среднее быстродействие - 500 тыс. оп/сек.

Для создания средств автоматики используется 8-разрядный МП K1821BM85 (Intel 8085), который является усовершенствованием K580. МП совместим по системе команд с МП K580, но имеет более высокое быстродействие, одно напряжение питания, возможность последовательного ввода-вывода данных, встроенные тактовый генератор, системный контроллер и шинный формирователь, а также малое потребление, т.к. выполнен по КМОП-технологии.

8- и 16-разрядный комплект серии 1810 выполнен по n-МОП технологии, содержит около 29 000 элементов и предназначен для реализации не только контроллеров, но и высокопроизводительных вычислительных систем. В состав серии входят: K1810BM86 (Intel 8086) - однокристальный 16-разрядный МП с быстродействием до 2,5 млн. оп/с, емкостью адресуемой памяти 1Мбайт и системой команд, совместимой с K580 на ассемблере; K1810BM87 (Intel 8087) - однокристальный 16-разрядный сопроцессор арифметики с плавающей запятой; K1810BM88 (Intel 8088) - однокристальный 8-разрядный МП с быстродействием в 2 раза большим, чем у K1821BM85A, емкостью адресуемой памяти 1Мбайт и системой команд, совместимой с K1810BM86; K1810BM89 (Intel 8089) - однокристальный 16-разрядный сопроцессор ввода-вывода.

К наиболее важным достоинствам BM86 относятся: развитая регистровая структура, существенно уменьшающая число обращений к памяти; конвейерный принцип выполнения команд с предварительной выборкой; мультиплексированная шина адреса - данных; многофункциональное использование выводов, позволяющее адаптировать МП к уровню сложности разрабатываемой системы; способность координировать взаимодействие нескольких процессоров, упрощающее построение мультиплексных систем. Дальнейшим развитием указанной линии являются процессоры Intel 80286, 80287, 80386 и 80387 различных модификаций (табл.2.1).

Основные характеристики МП

Таблица 2.1

Марка МП	Сопроцессор	Адрес. память	Такт. частота (МГц)	Быстродействие
8086/88 (1979г.)	8087	1 Мбайт	4,77(8; 10)	0,33
80286 (1982г.)	80287	16 Мбайт	8(12; 16; 20)	1,2
80386SX (1985г.)	80387	4 Гбайт	16 (20-33)	2,5
80386DX	встроенный	4 Гбайт	16 (20-40)	6,0

Последний из них - микропроцессор 80386 совместим снизу вверх с предыдущими поколениями МП фирмы Intel. Так как набор команд МП 80386 и его модули обработки являются расширениями набора команд предшествующих моделей от 8086/88 до 80286, программное обеспечение последних совместимо снизу вверх с МП 80386.

Специфическими особенностями МП 80386 являются многозадачность, встроенное управление памятью, виртуальная память с разделением на страницы, защита программ и большое адресное пространство. Аппаратная совместимость с предыдущими моделями

сохранена посредством динамического изменения разрядности магистрали. МП 80386 предусматривает переключение программ, выполняемых под управлением различных операционных систем, такие как MS-DOS и UNIX. Это свойство позволяет разработчикам программ включать стандартное прикладное программное обеспечение для 16-разрядных МП непосредственно в 32-разрядную систему. Процессор определяет адресное пространство как один или несколько сегментов памяти любого размера в диапазоне от 1 байт до 4 Гбайт. Эти сегменты могут быть индивидуально защищены уровнями привилегий и таким образом могут избирательно разделяться различными задачами. Механизм защиты основан на иерархии привилегий или ранжированного ряда. Это означает, что разным задачам или программам могут быть присвоены определенные уровни, которые используются для данной задачи (Например, для ядра операционной системы - уровень 0, а для программ МП8086 - уровень 3). МП 80386 имеет два режима работы: режим реальных адресов, называемый реальным режимом, и защищенный режим. При подаче сигнала сброса или при включении питания устанавливается реальный режим, причем МП 80386 работает как очень быстрый МП 8086, но, по желанию программиста, с 32-разрядным расширением. В реальном режиме МП 80386 имеет такую же базовую архитектуру, что и МП 8086, но обеспечивает доступ к 32-разрядным регистрам. Полные возможности МП 80386 раскрываются в защищенном режиме. Программы могут исполнять переключение между процессами с целью входа в задачи, предназначенные для режима виртуального МП 8086. Каждая такая задача проявляет себя в семантике МП 8086 (т.е. в отношениях между символами и приписываемыми им значениями независимо от интерпретирующего их оборудования). Это позволяет выполнять на МП 80386 программное обеспечение для МП 8086 - прикладную программу или целую операционную систему. В то же время задачи для виртуального МП 8086 изолированы и защищены как друг от друга, так и от главной операционной системы МП 80386.

Завершим обзор данной линии перечнем современных типов 32-разрядных микропроцессоров Intel: 80486, 80586, Pentium, Pentium MMX, PentiumPro и т.п. Среди других фирм, производителей 32-разрядных микропроцессоров, следует упомянуть Motorola (MC68020), National Semiconductor (NS32032), Zilog (Z80000).

Комплекты однокристальных МП серий 1801, выполненных по n-МОП технологии, и 1811, выполненных по КМОП-технологии, ориентированны на микропрограммную эмуляцию команд "Электроники-60" (или DEC - стандарта). Развитой комплект, включающий более 30 БИС и позволяющий строить разнообразные управляющие и вычислительные системы, в течение длительного времени был базовым в России для микроЭВМ промышленного назначения. Дальнейшим развитием указанной линии можно считать МП серии K1843 - первый отечественный комплект, выполненный по КМОП-технологии и предназначенный для построения систем высокой производительности.

Комплекты быстродействующих МП серии 1802 и серии 1804 рассчитаны на реализацию структур машин с произвольной системой команд за счет секционной архитектуры. Ориентация на выбранную систему команд реализуется эмуляцией ее на микропрограммном уровне путем занесения в ПЗУ необходимых для этого прошивок. На базе серий МП могут быть построены специализированные средства автоматики, различные контроллеры и микроЭВМ. Трудоемкость таких разработок может быть преодолена с помощью САПР, созданных для этих целей.

Форт - МП K1865BМ1 -КМОП-БИС представляет собой однокристальный 16-разрядный МП, непосредственно реализующий язык программирования Форт. МП реализует RISC-системы команд, стек - ориентированную архитектуру с частотой синхронизации до 4 МГц, выполнением всех команд за один или два цикла. МП можно использовать для построения систем, работающих в реальном времени. Аппаратная поддержка интерактивного языка программирования обеспечивает возможность выполнения программ с экономией до 75% времени в сравнении с традиционными языками программирования.

Однокристалльные микроконтроллеры

В отличие от рассмотренных ранее МП с архитектурой **фон Неймана**, в однокристалльных микроконтроллерах (МК) используется, как правило, так называемая **Гарвардская** архитектура. Основное различие состоит в использовании памяти и внутренних шин. В архитектуре фон Неймана имеется общая память для программ, данных и общее адресное пространство. Такой подход обеспечивает универсальность, унификацию и возможность модификации программ. В Гарвардской архитектуре используются отдельные блоки ОЗУ и программной памяти. Так как обычно объем памяти, занимаемый программой, существенно больше памяти данных и обращение к данным вдвое чаще, чем к командам, то такое разделение дает большие преимущества, особенно при однокристалльном исполнении. Ввиду существенного сокращения адресного пространства данных уменьшаются также разрядность адресов и длина команды в целом. Наличие таких отдельных блоков памяти позволяет организовать их параллельную работу и сократить цикл выполнения команды. Следует отметить также развитые средства ввода-вывода, реализуемые в виде нескольких независимых портов, отдельных тестируемых линий, упрощенной системной магистрали, канала последовательного ввода-вывода, а в некоторых случаях наличие встроенных АЦП-ЦАП и таймеров-счетчиков, а также схемы для организации несложной системы прерываний.

Фирма Intel в течение 1976-1980 гг. создала семейство 8-разрядных однокристалльных микроконтроллеров **MCS-48**, включающее приборы 8048, 8035, 8049, 8039 и др. Архитектура этого семейства стала своеобразным стандартом в классе недорогих контроллеров. Отечественной промышленностью разработаны микроконтроллеры серии K1816BE48 аналогичной архитектуры. Все МК этой серии имеют одинаковую структуру. МК включает: 8-разрядный ЦПУ; блок ввода-вывода с тремя 8-разрядными портами P0, P1, P2; блок памяти программ; блок ОЗУ для хранения данных; таймер-счетчик T на 8 разрядов; блок прерываний; встроенный тактовый генератор. К средствам ввода относятся и три тестируемые командами независимые линии: T0, T1, INT. Следующим шагом является создание в 1981 г. семейства MCS-51 (K1816BE51), до настоящего времени доминирующего на рынке однокристалльных микроконтроллеров (характеристики будут рассмотрены более подробно в следующих главах) [14]. Далее в этом ряду были созданы MCS-8096 (далее 80C196) - 16-разрядные МК, содержащие 8К программной памяти, 232 байта памяти данных, а также п/с аналого-цифрового обмена, средства поддержки режима реального времени, включая скоростной ввод-вывод.

Кроме упомянутых 8-разрядных МК отечественная промышленность выпускает 4-разрядные МК для простых устройств управления промышленной автоматикой K1814, K1820; микроЭВМ для цифровой обработки сигналов KM 1813BE1, K1827; набор МК для бытовой техники K145, K1829 и др.

Специализированные микроЭВМ

За последние годы в микропроцессорной технике заметна тенденция создания высокопроизводительных приборов в ущерб универсальности. Этот факт можно связать с все более широким внедрением МП в системы реального времени, осуществляющие обработку сигналов и изображений и общим усложнением задач, решаемых на основе МП[1].

Наиболее мощным классом специализированных приборов являются МП для цифровой обработки сигналов (ЦОС). Небольшое число типов используемых команд, преобладание арифметических операций в алгоритмах ЦОС позволяют за счет специализации структуры и системы команд почти на два порядка поднять производительность и таким образом достигнуть нового качества - обработки сигналов в реальном масштабе времени. Такие МП, система команд и архитектура которых ориентирована на ЦОС, получили название цифровых процессоров обработки сигналов (ЦПОС). С точки зрения функциональной законченности весьма привлекательным является

размещение на кристалле вместе с процессором также многоканальных АЦП и ЦАП, тактового генератора, схем выборки и хранения.

Именно так и построена микроЭВМ КМ1813ВЕ1, представляющая собой микроЭВМ с встроенными ЦАП и АЦП, архитектура и система команд которой преимущественно ориентирована на решение задач цифровой фильтрации.

Следующий шаг был сделан в микроЭВМ ТМС32010, где были предусмотрены два режима использования - микропроцессорный и микроЭВМ. Завершают этот краткий перечень однокристалльные микроЭВМ КМ1827ВЕ3,4, предназначенные для цифровой обработки сигналов звукового диапазона частот в реальном масштабе времени.

Настоящий обзор в максимально-возможной степени привязан к отечественным сериям МП и МК. В связи с этим и из-за ограниченного объема были упущены и другие используемые для промышленной автоматики МП и МК, например, однокристалльные микроконтроллеры фирм Atmel, Philips, AMD и Matra, совместимые с iMCS-51 по системе команд; однокристалльные PIC - контроллеры фирмы Microchip с RISC-архитектурой, однокристалльные микроконтроллеры M16C/61 фирмы Mitsubishi electric с CISC-архитектурой, однокристалльные микроконтроллеры 68HC708 фирмы Motorola, C166, C167 - фирмы Siemens, Z80, Z8000 - фирмы Zilog и др.

Лекция 4. Организация запоминающих устройств

Запоминающее устройство (ЗУ) является неотъемлемой частью любой вычислительной системы. Сейчас мало кто помнит, что до середины 60-х годов термин “память” было запрещено применять в отношении вычислительных машин, так как считалось лженаучно отождествлять работу человеческого мозга с работой счётной машины. Вместо этого придумали массу терминов - оперативное запоминающее устройство (ОЗУ), постоянное запоминающее устройство (ПЗУ) и т.п. Многие из них применяются до сих пор.

Запоминающие устройства можно классифицировать по функциональному назначению на оперативные ЗУ (ОЗУ), которые служат для хранения быстроменяющихся данных, и постоянные ЗУ (ПЗУ), предназначенные для хранения постоянной или редко изменяемой информации.

Оперативные запоминающие устройства по способу обращения к массиву элементов памяти делятся на адресные и ассоциативные. В адресных ЗУ обращение к элементам памяти производится по их физическим координатам, задаваемым внешним двоичным кодом - адресом. Ассоциативные ЗУ поиск и выборку информации выполняют по содержанию ячеек памяти, независимо от их физических координат. По способу хранения информации ЗУ бывают статическими и динамическими. Элементы памяти статических ЗУ представляют собой бистабильные элементы, что определяет потенциальный характер управляющих сигналов и возможность сохранения информации с минимальными затратами. В динамических ЗУ для хранения информации используются инерционные свойства конденсаторов, что требует периодического восстановления состояния элементов памяти - регенерации. Несмотря на более простое управление, статическая память имеет ряд недостатков по сравнению с динамической памятью. Во-первых, она потребляет большую мощность, так как для реализации ячейки статической памяти требуется как минимум четыре транзистора, а для ячейки динамической памяти - один. Во-вторых, она дороже и при одинаковой степени интеграции с динамической памятью обладает меньшей информационной ёмкостью. Поэтому оперативная память современных микропроцессорных систем обычно строится на микросхемах памяти динамического типа. По технологии исполнения интегральные ЗУ имеют следующие разновидности: полупроводниковые ЗУ на основе структур, использующих схемотехнику ТТЛ; полупроводниковые ЗУ на основе МОП-технологии, использующие структуры p-МОП, n-МОП, КМОП. ЗУ на основе МОП-технологии совместимы по входам-выходам с ТТЛ логическими элементами.

Постоянные запоминающие устройства можно разделить на неизменяемые или масочно-программируемые, однократно электрически программируемые и многократно перепрограммируемые.

Неизменяемые или масочно – программируемые и однократно электрически программируемые устройства программируются только один раз и в процессе эксплуатации используются только для считывания хранимой в них информации. Информация заносится на заводе-изготовителе путем маскирования или пользователем-разработчиком с помощью специализированных программаторов.

Многократно перепрограммируемые устройства предполагают наличие возможности электрического занесения информации и ее стирание или изменение путем неэлектрического или электрического воздействия (соответственно различают устройства памяти с ультрафиолетовым стиранием и электрически перепрограммируемые). Особую роль в бортовых и промышленных ПК играет многократно перепрограммируемая флэш-память. В этих применениях она выполняет функции дискового накопителя. Дело в том, что ограниченные размеры и работа в условиях повышенных вибрационных и ударных нагрузок зачастую исключают возможность использования в микропроцессорной системе дисковой памяти. Микросхемы флэш-памяти отличаются высокой плотностью записи информации, небольшими размерами и высокой стойкостью к внешним воздействиям. Устройства постоянной памяти на схемах флэш-памяти и микросхемах ПЗУ других типов, служащие для замены дисковых накопителей, получили название “полупроводниковых дисков” [15]. Технология флэш-памяти анонсирована компанией Intel в 1989 г. Ежегодный прирост мирового производства флэш-памяти составляет 200%. Этот сектор рынка считается одним из самых динамичных и перспективных. Современная флэш-память выдерживает не менее 1 млн. циклов перезаписи. По стоимости флэш-память приближается к динамическому ОЗУ, сохраняя существенное преимущество - энергонезависимость. В настоящее время флэш-диски выпускаются с различными интерфейсами [16]. Предусмотрены их установка непосредственно в разъемы системной шины компьютера или контроллера (например, для шин ISA), подключение вместо жестких дисков (через интерфейсы IDE или SCSI) и, наконец, использование в качестве сменных дисков – карт (в стандартах PCMCIA или PC Card).

Подсистема памяти микропроцессорной системы имеет иерархическую структуру. В этой иерархии традиционно выделяются следующие уровни: регистровая память, которую часто называют местной памятью, регистрами общего назначения - РОН или сверхоперативным запоминающим устройством - СОЗУ; буферная память или кэш-память; основная оперативная память или ОЗУ; постоянная память (ПЗУ) и внешняя память (рис.2.1.).

Регистровую или местную память образуют регистры общего назначения процессора, большая часть которых доступна программисту и предназначена для хранения адресов, операндов и результатов выполнения операций. Число РОН в зависимости от типа микропроцессора лежит в пределах 16-64. При небольшой емкости этот уровень памяти имеет самое высокое быстродействие - время доступа к данным не

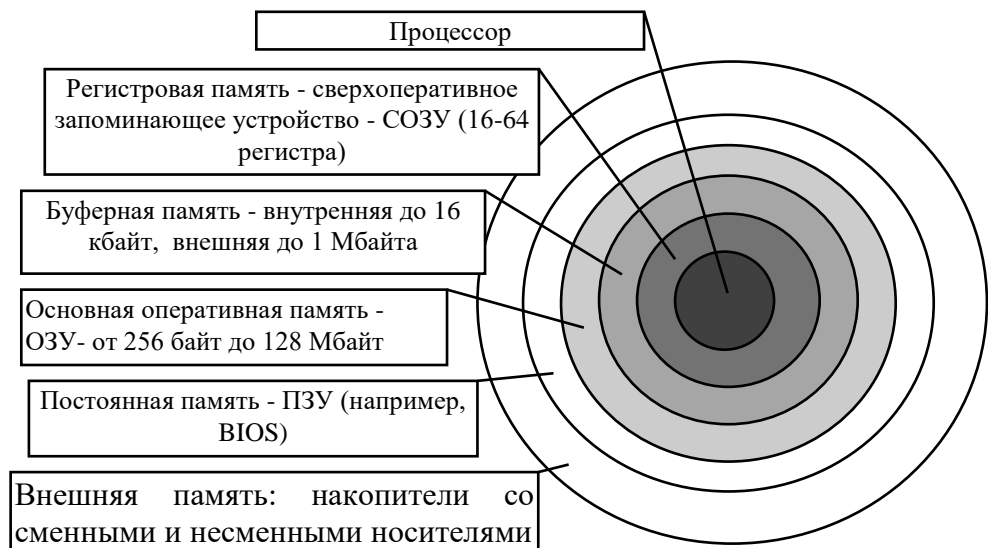


Рис.2.1. Структура памяти микропроцессорных устройств

превышает единиц или десятков наносекунд.

Промежуточное положение между регистровой и основной оперативной памятью занимает кэш-память. Большинство современных микропроцессоров имеют двухуровневый кэш. Первый уровень - внутренний кэш - располагается на кристалле процессора и работает на его тактовой частоте. Второй уровень - внешний кэш - устанавливается на плате процессора и работает на частоте шины. Кэш-память выполняет в микропроцессорной системе роль буфера между быстрым процессором и медленной динамической памятью. Она реализована на микросхемах памяти статического типа и может работать на тактовой частоте процессора, что позволяет исключить такты ожидания и существенно повысить пропускную способность канала процессор - основная память. Кэш используется также для обмена данными между оперативной памятью и внешними накопителями. Объём внутреннего Кэша у 32 разрядных процессоров достигает 256 кбайт, а внешнего - до 1 Мбайта [17].

Оперативная память - ОЗУ составляет от 256 байт у простейших 8-разрядных однокристальных микропроцессоров до 384 Мбайт у мощных 32-разрядных процессоров. По организации, методу доступа, постоянная память (ПЗУ) существенно не отличается от оперативной и служит для хранения управляющих программ, например, программ базовой системы ввода-вывода (BIOS). В ПЗУ также хранятся редко изменяемые данные микропроцессорной системы. Постоянная память может быть реализована и на уже упоминавшейся флэш-памяти, объём которой на момент написания пособия составлял до 3Гбайт.

Верхний уровень структуры занимает внешняя память. Она реализуется в виде различных накопителей со сменными (накопители на жестких и гибких дисках, накопители на оптических дисках, стримеры и т.д.) и несменными носителями (флэш-память). Это память самая медленная, но самая большая по объёму. В частности, ёмкость накопителей на оптических и жестких дисках достигает нескольких гигабайт. Как видно из названий, этот тип памяти может уже не являться чисто электронным, но содержать и механические элементы. Однако следует учитывать, что для промышленного применения созданы малогабаритные накопители в защищенном исполнении, допускающие установку даже непосредственно на платы контроллеров. Это позволяет рассматривать их как элемент памяти объектного контроллера.

Лекция 5. Организация интерфейса, магистрали, шины

Как уже было указано выше, объектный контроллер наряду с вычислительными элементами (процессорами, памятью и т.п.) содержит элементы, обеспечивающие сопряжение с внешними устройствами систем автоматизации (датчиками, преобразователями, компьютерами, контроллерами). Существует несколько связанных между собой понятий, описывающих совокупность схмотехнических средств и правил, обеспечивающих непосредственное взаимодействие составных элементов систем обработки данных и компьютеров. В их числе: “интерфейс”, “протокол”, “канал” и т.п.

Под интерфейсом понимают совокупность аппаратных, программных и конструктивных средств и правил, обеспечивающих взаимодействие компонентов вычислительной системы или сети и учитывающих информационную, электрическую и конструктивную совместимость их элементов. Протокол подразумевает правила, определяющие взаимодействие абонентов системы или сети и способ выполнения определённых функций, а канал - среду распространения сигналов, используемый совместно различными блоками системы [18].

При этом линиями интерфейса являются электрические цепи (провода или проводники на плате, провода припаянные к выводам разъемов, в которые вставляются печатные платы, либо плоский кабель), по которым распространяются те или иные сигналы. Совокупность линий интерфейса, сгруппированных по функциональному назначению, называют “шиной”. Шина данных, шина адресов, шина управления и состояния вместе образуют совокупность всех линий интерфейса, которую называют “магистралью”. Слова

шина и интерфейс часто употребляют как взаимозаменяемые, потому что далеко не всегда ясно, где кончаются функции аппаратных средств, какова роль программно-аппаратных средств и где начинаются функции системного и прикладного программного обеспечения.

Системные шины

Номенклатура шин постоянно увеличивается, что отражает, с одной стороны, неудовлетворённость имеющимися, а с другой - возникновением всё более сложных задач, решаемых компьютерным оборудованием, непрерывное усовершенствование архитектуры и элементной базы компьютеров. Разнообразие системных шин объясняется различием в типах микропроцессоров. Например, фирма DEC использует шину Q-bus для систем с микропроцессорами MicroVAX, фирма Motorola - шину VME-bus для микропроцессоров 68000, фирма Texas Instruments - шину Nubus для микропроцессоров TI 99000 и Motorola 68030. Наиболее распространёнными являются системные шины для компьютеров с Intel-подобными микропроцессорами. Системные шины предназначены для организации взаимодействия периферийного оборудования, подключаемого через соответствующие контроллеры, с ядром компьютера, в состав которого входят микропроцессор и память (ОЗУ и ПЗУ). Пропускная способность шины измеряется в килобайтах (мегабайтах) в секунду и зависит от тактовой частоты, на которой работает шина и её разрядности. Шины могут быть 8-, 16-, 32- и 64-разрядными в зависимости от типа микропроцессора.

Первой системной шиной, разработанной для компьютеров PC/XT, в основе которых лежали микропроцессоры Intel 8086 и 8088, была шина PC/XT-bus. Она была 8-разрядной, а её контроллер обеспечивал работу на частоте микропроцессора (4,77 МГц). С появлением 16-разрядных микропроцессоров 80286 и 80386 (версии SX) была разработана системная шина PC/AT-bus. На базе этих шин был разработан международный промышленный стандарт ISA (Industry Standard Architecture), широко использующийся в современных микропроцессорных системах и компьютерах. Помимо увеличения разрядности, числа линий запросов прерывания (IRQ) и числа каналов прямого доступа к памяти (DMA) сохранилась преемственность шин на уровне контактов разъёмов. Благодаря этому в новых системах можно использовать разработанные ранее контроллеры. С появлением 32-разрядных микропроцессоров 80386 (версия DX) и 80486 было создано несколько новых системных шин. Фирма IBM создала системную шину MCA (Micro Channel Architecture), отказавшись от совместимости с шиной ISA, а фирмы Compaq, NEC и ряд других продолжили совершенствовать шину ISA, приняв новый стандарт системной шины EISA (Extended ISA), который позволял одновременно использовать ISA-контроллеры и карты, разработанные для ISA-16 и ISA-8. Однако многочисленные исследования показали, что единая системная шина, объединяющая быстрые и медленные устройства, является “узким местом” компьютера и не позволяет добиться максимальной производительности вычислительной системы. Поэтому появились локальные шины.

Локальные шины

Локальной шиной обычно называется шина, выходящая непосредственно на контакты процессора. Она объединяет процессор, память, схемы буферизации для системной шины и её контроллер, а также некоторые другие вспомогательные схемы, расположенные на плате процессора, и является шиной процессора. Локальная шина позволяет ускорить операции обмена данными, т.к. работает на частоте, равной внешней частоте микропроцессора. Для того чтобы добиться совместимости плат расширения разных производителей компьютерной техники была создана ассоциация стандартов видео оборудования - Video Equipment Standards Association (VESA), которая в 1992 году выпустила спецификации на локальную шину VLB (VESA Local Bus). Достоинством шины VLB является высокая скорость обмена данными и отсутствие специального контроллера. Однако эта шина не была рассчитана на работу с микропроцессорами, пришедшими на смену процессору 80486: Pentium, Alpha, PowerPC и др. Поэтому был разработан альтернативный стандарт, названный Peripheral Component Interconnection (PCI) - взаимосвязь периферийных компонентов. Фактически оба стандарта продолжают дорабатываться до сих пор, но уже нет

сомнения, что стандарт PCI завоёвывает всё более прочные позиции, т.к. благодаря интегральному контроллеру/акселератору шина стала процессорно-независимой, что даёт возможность избежать разных версий одной платы расширения. Кроме того, стандарт PCI позволяет конфигурировать устройства компьютера программным способом без перезагрузки сразу после обнаружения нового устройства. Это свойство шины особенно полезно в промышленных компьютерах, контроллерах и файловых серверах, то есть в наиболее бурно развивающихся приложениях вычислительной техники [19]. Графическая иллюстрация иерархии шин в структуре компьютера представлена на рис. 2.2.

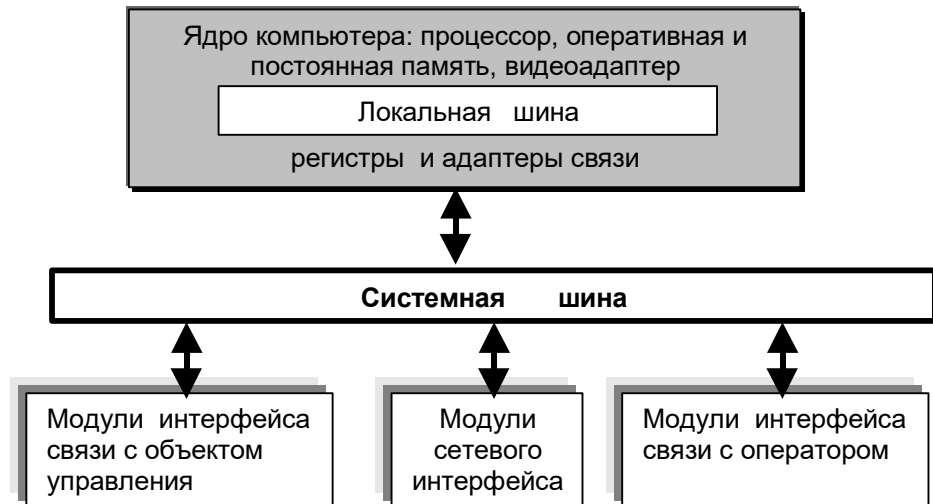


Рис. 2.2. Шинная архитектура компьютера

Шинная архитектура промышленных компьютеров

Под промышленными понимают компьютеры, используемые в составе систем управления промышленным оборудованием и производственными процессами. Первые промышленные компьютеры были попыткой адаптации ПК массового применения к нуждам промышленного производства.

Этот шаг сделала фирма IBM, стремясь расширить сферу применения архитектуры PC/XT/AT. Промышленные компьютеры представляли собой обычные ПК, в конструкцию которых был внесён ряд изменений, обеспечивающих возможность работы в более тяжелых условиях эксплуатации. Очень скоро стали ясны недостатки такого подхода - прежде всего ненадёжность процессорных (системных) плат. Одним из путей повышения надёжности промышленных компьютеров стало использование пассивной объединительной панели вместо системной платы. Активные компоненты стандартной системной платы (микропроцессор, ОЗУ, контроллеры и др.) в этом случае монтируются на отдельной плате, которая, как обычная расширительная плата, вставляется в гнездо панели. Пассивная электрическая плата по электрическим параметрам и требованиям к соединительным гнездам совместима со стандартами ISA и EISA и, следовательно, позволяет использовать все расширительные платы, созданные для обычных ПК на базе этих шин. Подобные платы, если они построены на микросхемах повышенной степени интеграции, дают возможность создать значительно более компактную машину, сохраняя при этом полную совместимость с архитектурой IBM PC/AT. К плюсам можно отнести более эффективное охлаждение за счет равномерной вентиляции, простоту ремонта, обслуживания и модернизации системы [20]

Однако стандарт шины ISA не позволяет значительно повысить надёжность вычислительной системы, т.к. он создавался без учёта требований промышленных применений. Это обстоятельство побудило американскую фирму Rapid System разработать промышленный вариант стандарта ISA под названием "Расширение шины ПК для промышленности" (PC bus extension for Industry, PCXI). Стандарт PCXI допускает использование соединительных гнезд ISA/EISA, но увеличивает расстояние между ними, что

позволяет использовать платы с дополнительными элементами, увеличивающими их толщину. Каждая плата заключается в экранирующий кожух, на лицевой панели которого расположены индикаторы и органы управления, а также элементы крепления к корпусу компьютера [21].

Постоянное удешевление аппаратных средств стимулировало работы по внедрению РС - совместимых компьютеров на нижний уровень управления. Это привело к появлению встроенных ПК, которые не имеют собственного корпуса, источника питания и терминала пользователя. Как правило, это одно - или двухплатный модуль, содержащий элементы стандартной системной платы и блок сопряжения с шиной промышленной управляющей системы. Поскольку встраиваемый компьютер размещается в корпусе оборудования управляющей системы, к нему предъявляются повышенные требования. При этом возникают ограничения, прежде всего на объём машины. Встраиваемые компьютеры используют стандартные ISA/EISA и нестандартные фирменные шины. Из фирменных шин наибольшее распространение получили шины STD32 - 32-разрядный вариант шины STD80 и G-64/G-96. Последний стандарт ориентирован на платы формата Eurocard 34. После начала производства плат с процессором Pentium в промышленных компьютерах активно внедряется шина PCI и ее производные, которые позволяют улучшить показатели видеосистем и выводят промышленные компьютеры на новый уровень.

Внешний интерфейс контроллеров и промышленных компьютеров

Промышленный компьютер управляет оборудованием и технологическим процессом, основываясь на информации, получаемой от объекта управления. Функции управления реализуются, в свою очередь, за счёт выходных информационных потоков. Внешний интерфейс контроллера можно разделить на три части: интерфейс связи с объектом управления (или устройство связи с объектом - УСО), сетевой интерфейс, служащий для сопряжения с другими ПК, контроллерами и системами управления, и, наконец, интерфейс связи с оператором (см. рис. 2.2.).

Интерфейс оператора - это обычно какая-либо клавиатура и дисплей, состоящий из нескольких строк знакосинтезирующих индикаторов. Большинство клавиатур строится как матрица механических контактов, которые при нажатии замыкают соответствующие строки с соответствующими столбцами. Существует много типов дисплеев. Два основных класса: активные и пассивные. Принцип работы пассивного дисплея основан на отражении света внешнего источника. Так работают жидкокристаллические дисплеи (LCD- дисплеи). Они потребляют очень небольшой ток, примерно 10 мА на 4-строчный дисплей. LCD - дисплеи с подсветкой фона потребляют значительно больше энергии. Катодолуминесцентные дисплеи имеют высокую яркость свечения, позволяют выводить символы разных цветов и не нуждаются во внешнем источнике света. Такой дисплей потребляет 500-600 мА от источника питания 5 В.

Интерфейс связи с объектом управления включает в себя датчики измеряемых величин, преобразователи сигналов датчиков в аналоговую и цифровую формы и устройства управления технологическим оборудованием. Устройства ввода и вывода могут быть выполнены на одном кристалле микроконтроллера (как в MCS-51, где различные модификации предусматриваются составом БИС) и в виде функционально законченных модулей (плат расширения), имеющих интерфейс для подключения к системной шине контроллера. Перечень модулей расширения для промышленных контроллеров огромен, так как каждая фирма-производитель стремится максимально удовлетворить запросы пользователя. Кроме того, модули, идентичные по функциональным характеристикам, имеют несколько конструктивных модификаций в зависимости от условий эксплуатации. Например, краткое описание характеристик модулей сопряжения промышленного компьютера семейства PCL фирмы Advantech [22]: PCL-720 - модуль дискретного ввода/вывода и универсальный таймер, 32 линии ввода/вывода и 3-канальный таймер; PCL-730 - модуль дискретного ввода/вывода с гальванической развязкой, 32 изолированных канала ввода/вывода (напряжение пробоя изоляции 2500 В); PCL-722 - 144 канала

ввода/вывода с возможностью группового обмена; PLC-830 - модуль 10-канального таймера/счётчика; PCLD-788 - модуль мультиплексора (коммутатор 16 каналов на 1); PCLD-774 - расширяемый модуль аналогового ввода. Этот список далеко не полон.

Сетевой интерфейс объектного контроллера включает модули поддержки стандартных интерфейсов - RS-232, TTY (токовая петля 20 мА), RS-422A, RS-485, CAN, модули поддержки сетевых интерфейсов - Ethernet и другие, а также различные преобразователи интерфейсов. Рассмотрим основные характеристики коммуникационных интерфейсов.

RS-232. Это наиболее широко распространённый интерфейс, известный также как интерфейс V.24. Стандарт RS-232 первоначально был создан для соединения терминала и компьютера. В настоящее время многие периферийные устройства используют этот интерфейс. Интерфейс включает 13 линий и служит для соединения двух устройств, хотя для передачи данных достаточно 3 линий. Устройства могут обмениваться информацией одновременно, то есть работать в дуплексном режиме. Основные характеристики интерфейса RS-232: стандарт - EIA RS-232C, CCITT V.24; максимальная скорость обмена - 19200 bps (бит в секунду); максимальная дальность связи - 15 м; физический интерфейс - передача цифровых сигналов уровнями напряжения. Уровню логического "0" соответствует напряжение -3 В...-15 В, логической "1" соответствует уровень +3 В...+15 В; количество передатчиков - 1; количество приёмников - 1; принцип работы - полный дуплекс, point-to-point.

TTY (токовая петля 20 мА). Первоначально был создан для управления телетайпом. В настоящее время используется для управления программно-логическими контроллерами, дисплеями и принтерами. К достоинствам интерфейса можно отнести более легкое использование тока для передачи данных, чем напряжения. Это делает интерфейс надёжным, особенно учитывая его высокую защищённость от электрических помех. Используются две пары проводов: одна для передачи и одна для приёма. Интерфейс TTY полнодуплексный. Он особенно эффективен для передачи данных на большие расстояния в условиях сильных электрических помех. Недостаток - низкая скорость обмена. Характеристики: стандарт - TTY; максимальная скорость обмена - 9600 bps (бит в секунду); максимальная дальность связи - 1000 м; физический интерфейс - токовая петля; передатчик - 1; приёмник - 1; принцип работы - полный дуплекс, point-to-point.

RS-422-A. Этот симметричный интерфейс был принят в 1975 году стандартом EIA RS-422-A, как интерфейс межмашинного обмена. Характеристики: стандарт интерфейса - EIA RS-422-A, CCITT V.11; максимальная скорость обмена - 10 Mbps (Мбит в секунду); максимальная дальность связи - 1200 м; физический интерфейс - дифференциальное напряжение, витая пара; передатчик - 1; приёмник - до 10; принцип работы - полный дуплекс, point-to-point.

RS-485. Интерфейс RS-485 - наиболее широко распространённый в промышленных системах управления, базирующийся на двухстороннем, симметричном способе передачи данных в линии типа "витая пара". Он поддерживает многоточечное соединение в сети до 32 узлов и обеспечивает передачу данных на расстояние до 1200 м. Добавив один расширитель интерфейса RS-485, можно увеличить дистанцию на 1200 м и подсоединить ещё 32 узла и т.д. Характеристики: стандарт интерфейса - EIA RS-485; максимальная скорость обмена - 10 Mbps (Мбит в секунду); максимальная дальность связи - 1200 м; физический интерфейс - дифференциальное напряжение, витая пара; передатчик - до 32; приёмник - до 32; принцип работы - полудуплекс, multi-drop. Модули расширения, поддерживающие стандартные вышеописанные интерфейсы, обычно являются универсальными и программируются программным способом или перемычками.

Также применяется приборный интерфейс стандарта *IEEE-488*. Например, модуль PCL-848A/B фирмы Advantech, позволяет легко управлять 14 устройствами, передавая данные со скоростью 300 кбайт в секунду в режиме прямого доступа к памяти и 67 кбайт в секунду в режиме программного обмена.

Интерфейс локальных сетей. Для создания межкомпьютерных, межконтроллерных, смешанных сетей разных топологий служат 8-, 16- и 32-битные сетевые адаптеры, например, стандарта Ethernet. Линия связи выполняется на витой паре или коаксиальном кабеле. Характеристики интерфейсной карты PCL-843 фирмы Advantech: поддерживает стандарты IEEE 802.3, 10BASE-2 (передача по коаксиальному кабелю), 10BASE-T (передача по витой паре), скорость передачи - до 10 Mbps (Мбит в секунду), 8- и 16-битный режим обмена, 16 Кбайтный высокоскоростной внутренний буфер, максимальная дальность связи - 185 м (коаксиальный кабель), 100 м (витая пара), предусмотрена установка ПЗУ удалённой загрузки. В этом случае можно построить бездисковую систему, которая после подачи питания автоматически загрузит и сконфигурирует драйверы сетевой поддержки.

I²C-bus. В последнее время для построения интеллектуальных цепей автоматического управления используется коммуникационная шина I²C-bus, разработанная фирмой Philips Semiconductor. Она является дешёвой альтернативой интерфейса RS-232. Шина I²C (Inter Integrated Circuit bus) является двунаправленной, 2-проводной (линия синхронизации и линия последовательных данных). Использование шины позволяет строить сети различной конфигурации, в том числе и одноранговые.

Лекция 6. Классификация объектных контроллеров

Расширение сфер применения ПК породило разнообразие структур, конструктивных решений, подходов к объектной ориентации и программированию в процессе эксплуатации. Естественное стремление пользователей и производителей ПК к стандартизации, в свою очередь, вызывало к жизни различные попытки классификации ПК [2,3,12,22,23,24]: от самой простой, основанной на подсчете количества входов/выходов (микро - с количеством входов/выходов до 64, малые - от 65 до 256, средние - от 257 до 512 и большие - свыше 512), до сложных, классифицирующих ПК по функциональным признакам (например, регулирующий микропроцессорный контроллер – Ремиконт, логический - Ломиконт и диалоговый - Димиконт). В основу классификации объектных контроллеров могут быть положены следующие признаки, характеризующие степень или способ объектной ориентации аппаратных средств:

- Степень аппаратной избыточности.
- Способ объектной ориентации аппаратной части.
- Архитектура микроконтроллера (процессора).
- Конструктивное исполнение.
- Возможности аппаратной адаптации в процессе эксплуатации.

Степень аппаратной избыточности является одной из основных технико-экономических оценок при выборе ОПК и характеризует достигнутый на данной стадии жизненного цикла (проектирование, наладка, эксплуатация) уровень аппаратной оптимизации (от одноплатных встраиваемых контроллеров без избыточности до универсальных модульно - адаптируемых контроллеров с высоким уровнем избыточности).

Способ объектной ориентации аппаратной части определяет структурно-топологическую реализацию системы входов-выходов (встроенная в кристалл микроконтроллера, например, модификации микроконтроллера типа MCS-51 [14]; или на плату одноплатного ОПК, например, в одноплатных контроллерах типа PCA [22]; или распределенная по функциональным модулям входов-выходов, например, в универсальных промышленных контроллерах типа MicroPC [25,26]; или сетевая, например, при пространственно - распределенном размещении модулей типа ADAM [27], и т.п.) и иерархический уровень решения задачи объектной ориентации (на уровне алгоритмов управления в специализированных контроллерах, например, типа DICON и dTRON; на уровне схемных решений в одноплатных ОПК [28]; на уровне виртуального представления технологического объекта в панельных контроллерах типа MiPC-50 [22] и т.п.).

Архитектура микроконтроллера (процессора) однозначно определяет вычислительные возможности ОПК, в том числе вычислительную производительность, объем адресуемой памяти, возможности работы в режиме реального времени (количество и тип прерываний), возможности системной магистрали.

Конструктивное исполнение связано с возможностями доступа к элементам аппаратных средств для осуществления объектной ориентации (бескорпусные конструкции одноплатных ОПК с доступом на уровне СБИС, например, PCA; универсальные ОПК на основе монтажных каркасов с доступом к функциональным модулям, например, MicroPC, MiPC-50, MIC - 2000, PR -100, DV 105; без доступа к элементам в закрытых корпусах специализированных контроллеров, например, DICON и dTRON, ADAM 4000).

Возможности аппаратной адаптации в процессе эксплуатации должны быть обеспечены средствами модернизации, модификации и, наконец, замены элементов объектной ориентации аппаратных средств (элементы, модули, коммуникационные элементы, разъемы, интерфейсы, номенклатуры заменяемых модулей и т.п.).

Исходя из перечисленных классификационных признаков, аппаратуру ОПК можно отнести к четырем классам: специализированные технологические контроллеры, одноплатные объектные контроллеры, универсальные модульно-адаптируемые контроллеры и контроллеры с сетевой структурой (табл.3.1).

Далее в разделах главы представлены описания ОПК в соответствии с данной классификацией: специализированные технологические контроллеры, одноплатные объектные контроллеры, универсальные модульно-адаптируемые контроллеры и контроллеры с сетевой структурой. Отдельно рассмотрены вопросы объединения объектных контроллеров в сетевых структурах автоматизации.

Таблица 3.1

Критерии	Одноплатные объектные контроллеры	Специализированные технологические контроллеры	Контроллеры с сетевой структурой	Универсальные модульно-адаптируемые контроллеры
Архитектура микроконтроллера (процессора)	однокристалльный микроконтроллер или универсальный процессор	однокристалльный микроконтроллер или универсальный процессор	однокристалльный микроконтроллер или универсальный процессор	однокристалльный микроконтроллер или универсальный процессор
Способ объектной ориентации аппаратной части	уровень СБИС	уровень модулей	уровень модулей	уровень модулей
Возможности аппаратной адаптации в процессе эксплуатации	не предусмотрены	не предусмотрены	предусмотрены	предусмотрены
Степень аппаратной избыточности	без избыточности	без избыточности	без избыточности	с избыточностью
Конструктивное исполнение = реализация аппаратного доступа	без доступа	без доступа	без доступа	с доступом

Лекция 7. Технологии программирования объектных контроллеров

Современная система автоматизации технологических процессов любого уровня и сложности уже не может быть представлена без программируемых контроллеров (ПК) или промышленных компьютеров. Нормой становится понятие программируемого технологического оборудования, процесса или автоматизированной технологии. В

соответствии с изменением места ПК в автоматизации изменился и подход к их программированию. От используемых традиционно универсальных алгоритмических языков (Паскаль, Си, Модуль, Бейсик и др.) и соответствующих им инструментальных средств путь лежит к созданию и применению специализированных технологий программирования, то есть средств задания программ управления конкретными технологическими объектам [43].

Под термином технологии программирования при этом подразумевается очень широкий диапазон понятий: от использования языка команд процессора микроконтроллера до диалоговых систем визуального проектирования программ АСУ ТП, от программирования последовательностей действий оборудования до программирования автоматизированной технологии. Многообразие средств программирования отражает и современное понимание объектно-ориентированного подхода к проектированию сложных программных систем, когда объектное представление контроллера соответствует иерархическому уровню абстракции решаемой задачи. Следует отметить, что такой подход к программированию ОПК является столь новым, столь и очевидным. Еще в [44] указывалось, что развитие речевых, оптических, сенсорных (добавим сюда графических) средств ввода-вывода данных и языков спецификаций наметили новую линию языкового интерфейса - качественно новых языков программирования в терминах и непосредственных операциях прикладных предметных областей. Для ОПК использование языкового интерфейса в терминах и непосредственных операциях либо объекта автоматизации, либо функций управления является одним из общих требований. В основу классификации специализированных языков или средств программирования ОПК могут быть положены следующие признаки:

- иерархический уровень описания задачи,
- степень декомпозиции ПК,
- инструментальные средства,
- архитектура исполняемой системы.



Рис. 4.1. Классификация средств программирования ПК по иерархическим уровням описания задачи

Иерархический уровень описания задачи отражает классификацию средств программирования с точки зрения абстракции пользователя ПК. Задача управления технологическим процессом или объектом на различных стадиях жизненного цикла ПК (проектирование, внедрение, эксплуатация) формулируется и выполняется специалистами различных предметных областей. Соответственно используемое описание задачи должно быть максимально приближено к уровню знаний и терминам предметной области каждого специалиста. С этой точки зрения различают (рис.4.1) языки описания технологического процесса (объекта) на уровне оператора - технолога, описания системы автоматизации объекта на уровне инженера по автоматизации, описания алгоритмов или процедур программирования на уровне программиста - системотехника, описания команд микропроцессора на уровне программиста - схемотехника.

Как нетрудно заметить, такая классификация отражает уровень абстракции описываемого объекта. Если объектом описания (анализа) является технологический объект,

естественно язык его описания должен в большей степени отражать предметную область конкретной технологии (технологический стандарт). Например, для технологического процесса дозирования в качестве элементов языка могут использоваться понятия бункер, клапан, привод, трубопровод, рецепт, доза, компонент и т.п. Типичными примерами таких языков являются роботоориентированный язык для программирования движения робота [45] или графический язык визуального программирования АСУ ТП [9]. Если объектом рассмотрения является система автоматизации, то элементы языка должны соответствовать виду и сути, решаемых на данном уровне задач. В распоряжение инженера по автоматизации должны быть предоставлены средства описания структуры системы, задач управления, алгоритмов взаимодействия с автоматизируемым объектом, изменяемых и контролируемых параметров и т.п. (например, графический язык "Диаграмма функциональных блоков", где функциональные блоки - это программные объекты, представляющие специализированные функции управления [4]). Уровень программиста-системотехника соответствует выбору в качестве объекта проектирования (исследования) комплекса программ системы автоматизации. Используемые на этом уровне языки программирования (а это чаще всего алгоритмические или объектно-ориентированные языки программирования [9,46] Си, Си++, семейство языков Паскаль, аналогично Бейсик, Форт, Модула, Ада и т.п.) позволяют эффективно решать задачи реализации сложных вычислительных и управляющих процедур, организации вычислительных процессов в автономных или сетевых структурах АСУ ТП и т.п. И, наконец, рассмотрение в качестве объекта собственно микроконтроллера или одноплатного ОПК предрасполагает к языкам описания команд процессора или ассемблера данной конкретной архитектуры [14].

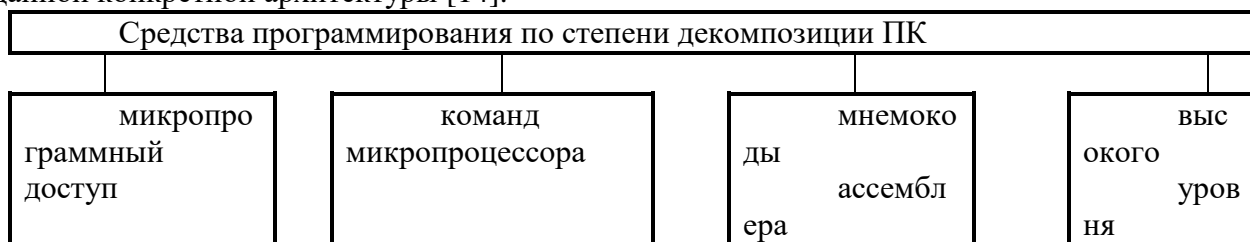


Рис. 4.2. Классификация средств программирования по степени декомпозиции

Степень декомпозиции ПК определяет возможность доступа пользователя к аппаратным ресурсам. ПК представляется в этом случае в виде взаимосвязанных элементов, модулей или контроллеров. Причем характер связи может быть задан (изменен) программно. Средства программирования должны соответствовать степени детальности описания аппаратных средств ПК и программному интерфейсу декомпозируемых элементов или модулей. Различают (рис.4.2) микропрограммный доступ к системе команд микропроцессора, программирование с помощью команд микропроцессора, программирование на языках ассемблера, программирование с применением средств высокого уровня. Микропрограммный доступ реализуется особенно эффективно для ПК, использующих секционные МП. Последние позволяют пользователю проектировать ПК с микропрограммными устройствами управления, при этом возможна эмуляция любой системы команд по выбору пользователя [47,48]. Программирование в кодах процессора исторически является первым способом программирования. Достоинством этого языка является требование знания системы команд и архитектуры МП, а для выполнения составленных таким образом программ не требуется ни каких трансляторов. Языки ассемблера или языки символического кодирования существенно облегчают труд программиста по сравнению с предыдущими языками, позволяя достигнуть практически аналогичного результата в реализации технических возможностей МП. Транслятор с этого языка, т.е. программа-ассемблер достаточно компактна. На ассемблере программируются, как правило, системные программы, исполняемые файлы, драйверы модулей и блоков и другие наиболее ответственные программы, обеспечивающие взаимосвязь программных

модулей [14]. Программирование с применением средств высокого уровня предполагает декомпозицию ПК на модули, поддерживающие программное взаимодействие. В качестве собственно языков программирования в этом случае могут быть использованы алгоритмические или объектно-ориентированные языки Си, Си++, семейство языков Паскаль и т.п.

Инструментальные средства ПК в значительной степени определяются стадией жизненного цикла программного обеспечения (в общем виде: анализ требований, проектирование, кодирование, тестирование и отладка, эксплуатация и сопровождение) и требованиями пользователя. Инструментальные средства и технологии программирования контроллеров достаточно жестко связываются с языковыми средствами и, собственно, аппаратными средствами ПК. С этой точки зрения средства программирования подразделяются на автономные или ориентированные на встроенные в ядро ОПК инструментальные средства программирования (встроенные или подключаемые панели программирования, типы и объем памяти, методы компиляции и т.п.), на использующие универсальные инструментальные средства программирования (инструментальные технические, программные и языковые средства), специализированные автоматизированные технологии программирования ОПК (в более общем виде получившие название CASE (Computer-Aided Software/System Engineering) - технологий), и, наконец, использующие для программирования SCADA (Supervisory Control And Data Acquisition) системы, то есть технологии визуального программирования (рис.4.3).

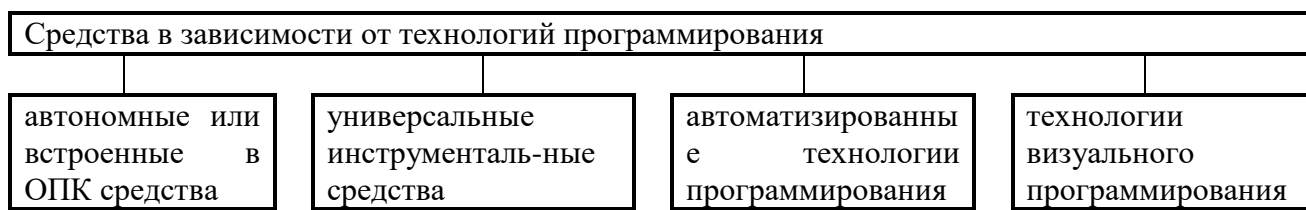


Рис. 4.3.Классификация средств программирования по используемым технологиям программирования

Использование технологий программирования, ориентированных на автономные или на встроенные в ОПК средства программирования, характерно для специализированных технологических контроллеров. Именно такие ПК чаще всего оснащаются встроенными или подключаемыми панелями программирования, имеют резидентные интерпретаторы языков (команд), характеризуются доступным интерфейсом пользователя при ограниченных возможностях. Наиболее типичным нам представляется программирование специализированного технологического контроллера типа Ремиконт Р-130, достаточно широко распространенного в России и программируемого с использованием резидентной библиотеки стандартных алгоритмов [28,49]. Практически большинство "малых" ОПК предполагает использование такой технологии программирования. В качестве языков могут применяться уникальные (аналогично Ремиконт Р-130), но чаще это стандартизованные языки типа уже упоминаемого графического языка "Диаграмма функциональных блоков" или графического языка "Диаграмма цепей".

Использование универсальных инструментальных средств программирования (инструментальные технические, программные и языковые средства) характерно для работы на начальных стадиях жизненного цикла программного обеспечения ПК при выборе в качестве объекта проектирования (исследования) комплекса программ системы автоматизации. Такой инструментарий позволяет эффективно решать задачи реализации сложных вычислительных и управляющих процедур, организации вычислительных процессов в автономных или сетевых структурах АСУ ТП, а также программирования однокристальных микропроцессоров. Это могут быть как машинно-ориентированные языки типа ассемблера, так и универсальные алгоритмические или объектно-ориентированные языки. В качестве инструментальных технических средств обычно используются рабочие

станции на базе IBM - совместимых компьютеров, оснащаемые инструментальными программно - языковыми пакетами [2,41].

Специализированные автоматизированные технологии программирования ОПК (CASE - технологии) представляют собой совокупность методологий анализа, проектирования, разработки и сопровождения сложных систем программного обеспечения, поддержанную комплексом средств автоматизации [50].

Основная цель CASE состоит в том, чтобы отделить проектирование программ от их кодирования и последующих этапов разработки, а также позволить разработчикам не вникать во многие детали среды разработки и функционирования программ. CASE предоставляет пользователям в конкретной предметной области наглядные, строгие и интуитивно понятные языки программирования. Принципиальным представляется графическая ориентация CASE, позволяющая описывать программы в виде двумерных схем, сокращающих объем описания задачи и обладающих простой и ясной структурой. Предметная направленность CASE обеспечивается наличием базы данных проекта, где находятся подготовленные для данной области применения структурные диаграммы, модели данных, организации, обработки, исходные коды, логика обработки и т.п. На этой основе осуществляется интеграция CASE - средств: общий пользовательский интерфейс по всем средствам, передача данных между всеми средствами, передача данных и средств между различными платформами, интеграция этапов проектирования и т.п. Среди других достоинств CASE следует отметить возможность прототипирования (построения быстрых прототипов системы), автоматическую генерацию документации проекта, автоматическую верификацию и контроль проекта на полноту и состоятельность на ранних этапах разработки, автоматическую кодогенерацию на основе целевого языка и, наконец, сопровождение и реинжиниринг проекта. Применительно к ОПК CASE - технологии наиболее полно реализованы в программных интегрированных системах ISaGRAF [51] и Ultralogic [52].

Использование для программирования ОПК SCADA - систем, то есть технологии визуального программирования, связано с применением в ПК процессоров высокой производительности (i80386, i80486 и т.п.) и перенесением на нижний уровень автоматизации технологий верхнего уровня. Поэтому первоначально предназначенные только для построения человеко-машинного интерфейса (ММИ) верхнего уровня АСУ ТП [53,54,55,56] уже в настоящее время интенсивно внедряются в сфере ОПК [57]. Разработка прикладной программы автоматизации в SCADA осуществляется в среде визуального (графического, мультимедийного и видео) представления объекта и средств автоматизации. Понятие язык программирования полностью трансформируется в понятие язык проектирования, а точнее язык взаимодействия оператора с ЭВМ в привычной для него предметной среде [58,59,60]. Предметная среда обеспечивается наличием соответствующих библиотек графических примитивов, экранных форм представления информации, формул и алгоритмов математической обработки и управления, драйверов и т.п. На Российском рынке SCADA - систем АСУ ТП следует отметить, прежде всего ТРЕЙС МОУД, разрабатываемую и поставляемую фирмой AdAstra Research Group [56,57]; Genesis, распространяемую в России фирмой ПРОСОФТ [58,59], Wonderware InToch, распространяемую в России фирмой PLC Systems [60].

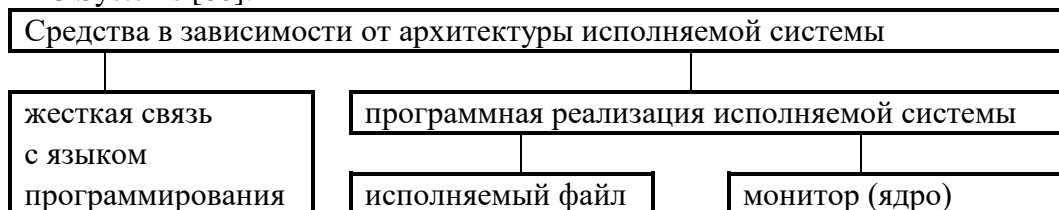


Рис. 4.4. Классификация средств программирования в зависимости от архитектуры исполняемой системы ПК

Архитектура исполняемой системы, строго говоря, не основа классификации, а ограничения, накладываемые на выбор средств программирования ОПК исполняемой системой, то есть ресурсами ПК, выделяемыми для решения поставленной задачи (рис.4.4). Возможно использование всех ресурсов ПК (аппаратных и программных) в качестве архитектуры исполняемой системы, то есть жесткая связь с языком программирования; возможна и только программная, то есть гибкая связь с языком программирования. К первой группе относят языки программирования, формирующие архитектуру микроконтроллера, например, Модула - ориентированные или Форт - ориентированные структуры микроконтроллеров [1], где встроенными средствами поддержки языка программирования являются и процессор, и микропрограммы, и программный монитор и т.д. Программная реализация исполняемой системы связана с выбором принципа перевода программы, написанной на языке программирования в коды (команды процессора). В случае ассемблера или написания программы в командах процессора исполняемая система представляет собой исполняемый файл. Применение различного вида интерпретаторов, компиляторов или трансляторов для преобразования систем команд исполняемая система должна содержать монитор (или ядро), реализующий указанную функцию.

В заключение хочется обратить внимание на постоянно упоминаемый признак: стандартный язык или стандарт на языки. Стандарты устанавливают общий ориентир развития, основные правила программирования, рекомендуемый перечень языков. Следование стандарту позволяет расширить возможности как проектанта, так и пользователя. Отклонение от стандартов для проектантов или пользователей оправданы лишь для широкомасштабных проектов, обещающих качественный скачок или большие возможности [48]. Именно поэтому представляет интерес подбор языков программирования для международного стандарта на языки программирования контроллеров ИЕС 1131-3 (МЭК 1131-3), введенного в 1992 году [4]. Стандарт позволяет применить при программировании контроллеров современные технологии программирования в сочетании с классическими языками для программирования контроллеров. Он предоставляет в распоряжение разработчика пять языков программирования, а именно:

графический язык "Последовательные функциональные схемы" (международная версия французского стандарта "Графсет"), описывающий логику программы на уровне чередующихся процедурных шагов и переходов;

графический язык "Диаграмма функциональных блоков", повсеместно используемый в Европе и позволяющий строить комплексную процедуру, состоящую из различных функциональных библиотечных блоков;

графический язык "Диаграмма цепей", распространенный в основном в США, язык релейных диаграмм для описания логических выражений любой сложности;

язык структурированного текста (из семейства языков Паскаль), реализующий гибкие процедуры обработки данных;

язык инструкций, относящийся к классу ассемблеров, для создания оптимизированных процедур.

Первые три графических языка ориентированы на рассмотрение системы автоматизации как объекта исследования на уровне предметной области инженера по автоматизации. При необходимости более детальной декомпозиции (например, разбиении системы автоматизации на подсистемы) осуществляется переход к языку структурированного текста (Паскаль) и решение задач на уровне инженера - программиста. Дальнейшая декомпозиция задачи автоматизации потребует привлечения языка программирования более низкого уровня, а именно языка инструкций (ассемблер) для исследований на уровне программиста - схемотехника. Выбор технологии программирования для данных языков и поддержка исполняемыми системами весьма разнообразна: от встроенных средств перепрограммирования контроллеров до универсальных инструментальных средств и интеграции их в новые CASE - технологии, например в ISaGRAF [51] и Ultralogic [52]. Стандартизованы таким образом достаточно

универсальные языковые средства для решения задач программирования системы автоматизации как объекта исследования различной степени декомпозиции. Программирование специализированных технологических контроллеров и программирование ОПК с помощью SCADA - систем - две полярные границы предметной области автоматизируемых технологий, и стандартизация их языковых средств нас еще ожидает.

В последующих разделах данной главы будут рассмотрены следующие наиболее типичные случаи применения языков программирования ОПК:

программирование одноплатных ОПК на основе однокристальных микроконтроллеров семейства MCS-51 на языке ассемблер с применением универсальных инструментальных средств;

программирование специализированных технологических контроллеров на примере Ремиконт Р - 130 со встроенными инструментальными средствами;

программирование универсальных ОПК графическими языками стандарта МЭК 1131-3 с применением элементов CASE - технологии инструментальной системы Ultralogic;

программирование ОПК с помощью SCADA - системы.

Лекция 8. Одноплатные объектно-ориентированные контроллеры

Появление однокристальных микроЭВМ и однокристальных микроконтроллеров явилось следствием совершенствования полупроводниковых технологий и схемотехники. Высокая надежность, низкое энергопотребление и небольшие размеры однокристальных микроконтроллеров позволили создать на их основе средства прямого управления технологическим оборудованием. Во многих случаях к ним предъявляются очень высокие требования по скорости реакции на внешние события (внешние прерывания) и скорости переключения с одной программы на другую (внутренние прерывания). Важным требованием является также высокая надёжность, поскольку отказ устройства управления может привести к выходу из строя дорогостоящего оборудования и порче продукции. Все эти требования привели к созданию одноплатных контроллеров, которые конструктивно объединяли на одной плате вычислительный блок, интерфейсные модули и модули ввода/вывода, то есть к выполнению ОПК в виде неизменяемого ядра, встраиваемого в технологический объект. Встраивание ОПК, наряду с оптимизацией режимов работы автоматизируемого технологического объекта, преследует цель оптимизации стоимости. Технологический объект в течение всего жизненного цикла не изменяется, поэтому средства адаптации к физическим сигналам объекта и исполняемый файл прикладной программы управления объектом также остаются неизменными. Средства адаптации к физическим сигналам объекта и исполняемый файл прикладной программы для такого случая, как уже отмечалось, вносятся в состав ядра. Программная поддержка работы перечисленных аппаратных средств осуществляется с помощью резидентного загрузочного модуля (BIOS) и программы - монитора реального времени. Наличие программ драйверов и средств программирования в ядре ОПК позволяет реализовать программный доступ к аппаратным ресурсам с целью решения задач автоматизации. Программирование одноплатных контроллеров выполняется обычно с помощью внешнего пульта или на специальном инструментальном вычислительном комплексе. Разработанная и отлаженная программа заносится в ПЗУ или ППЗУ контроллера.

Анализ химико-технологического оборудования показывает, что в основном (более 80 %) оно представляет собой индивидуальные относительно несложные машины и аппараты [28]. Использование для их автоматизации одноплатных ОПК является и технически, и экономически оправданным. Принципиально возможно два решения задачи автоматизации технологического объекта с помощью одноплатных ОПК: разработка под конкретный объект (проект) индивидуально ориентированного контроллера или выбор из ряда универсальных одноплатных ОПК контроллера, наименее избыточного с точки зрения конкретного объекта (проекта).

В первом случае аппаратная часть контроллеров создаётся только с учетом требований конкретного объекта автоматизации и имеет средства индикации и управления, также ориентированные только на технологические особенности объекта. Оптимизация аппаратуры контроллера, как правило, приводит к небольшому количеству дискретных входов/выходов, каналов аналогового, частотного ввода/вывода, наличию интерфейса для связи с контроллерами верхнего уровня, схемы управления датчиками, исполнительными механизмами и т.д. Требования к управлению технологическим оборудованием определяют тип и разрядность микропроцессора, объём оперативной памяти и памяти программ, что в свою очередь определяет выбор средств программирования и отладки, оптимально реализующих законы управления. Обычно в качестве основы ОПК в этом случае выбирают однокристалльный микроконтроллер. Он имеет ограниченный объём памяти, логически и физически разделённую память программ и данных, систему команд, ориентированную на выполнение алгоритмов управления. Функционально микроконтроллер объединяет в себе 8- или 16-битный центральный процессорный элемент, диспетчер управления прерываниями, памятью, встроенные таймеры, набор встроенных портов ввода-вывода, ЦАП, АЦП, схемы частотного вывода, канал последовательного ввода-вывода, память программ и данных. Это позволяет использовать его в тех системах, где ранее использовали одноплатные ЭВМ и схемы на жёсткой логике. Рассмотрим более подробно архитектуру семейства микроконтроллеров MCS-51, созданного фирмой Intel, (в России ему соответствует семейство микроконтроллеров типа 1816XX51).

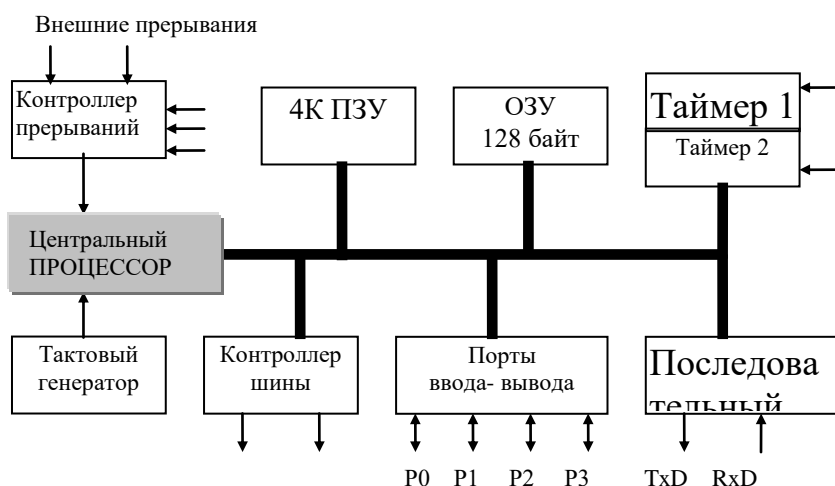


Рис. 3.5 Структурная схема микроконтроллера MCS-51

На рис.3.5 представлена структурная схема кристалла MCS-51.

Центральный процессор микроконтроллера имеет следующие технические характеристики [14]: разрядность АЛУ - 8 бит; число команд - 111; длина команд - 1,2 или 3 байта; число регистров общего назначения (РОН) - 32; число прямо адресуемых битовых переменных - 128; число прямо адресуемых битов в области регистров специального назначения - 128;

максимальный объём памяти программ - 64 кбайт; максимальный объём внешней памяти данных - 64 кбайт; максимальный объём внутренней памяти данных - 256 байт; методы адресации операнда - регистровый, косвенный, прямой, непосредственный. На частоте 12 МГц обеспечивается следующее время выполнения команд, мкс: сложение - 1, пересылка "регистр - внешняя память" - 2, умножение/деление - 4.

Прочие узлы микроконтроллера: внутренняя память программ масочного или перепрограммируемого типа объёмом от 4 до 32 кбайт, расположенная на кристалле (в некоторых модификациях она отсутствует); не менее чем 128-байтное резидентное ОЗУ данных, которое используется для организации банков РОН, стека, хранения данных пользователя; 32 двунаправленные интерфейсные линии ввода-вывода, индивидуально настраиваемые на ввод и вывод, объединены в четыре 8-разрядных порта; два 16-битных многорежимных таймера/счётчика, используемые для подсчёта внешних событий, организации временных задержек и тактирования коммуникационного порта; двунаправленный дуплексный асинхронный приемопередатчик (UART), предназначенный для организации каналов связи между микроконтроллером и внешними устройствами с широким диапазоном скоростей передачи информации. Имеются средства для аппаратно-

программного объединения микроконтроллеров в связанную систему; приоритетная двухуровневая система прерываний, поддерживающая не менее 5 векторов прерываний от 4 внутренних и 2 внешних источников событий; встроенный тактовый генератор.

Некоторые кристаллы имеют ряд специализированных узлов для управления двигателями (80C51F A,B,C) и обработки аналоговых сигналов (80C51GB). В прил.1 [30] приведены основные функциональные характеристики выпускаемых на сегодняшний день фирмой Intel микроконтроллеров. Некоторые микроконтроллеры имеют узкоспециальное назначение. К ним относятся связные контроллеры 8X152 и 8XC51SL. Как видно из данных приложения, уже на стадии выбора микроконтроллера может и должна быть осуществлена оптимальная объектная ориентация выбором соответствующего микроконтроллера из ряда микроконтроллеров с большим разнообразием характеристик в части аппаратного согласования. После выбора микроконтроллера на уровне стандартизованных схемных решений осуществляется дополнение функциональными и коммуникационными элементами, а также элементами интерфейса пользователя. Примером, иллюстрирующим такой подход, может служить приведенная на рис. 3.6. структурная схема ОПК для опроса и предварительной обработки сигналов аналоговых датчиков температуры. ОПК выполнен на базе микроконтроллера 80C51GB, имеющего встроенный 8-канальный АЦП (прил. 1).

Линеаризация и масштабирование сигналов аналоговых датчиков выполняется табличным преобразованием значений сигнала. Таблицы преобразования расположены во внешнем ППЗУ. В качестве системного интерфейса контроллера используется шина данных, подключенная к порту P0 микроконтроллера.

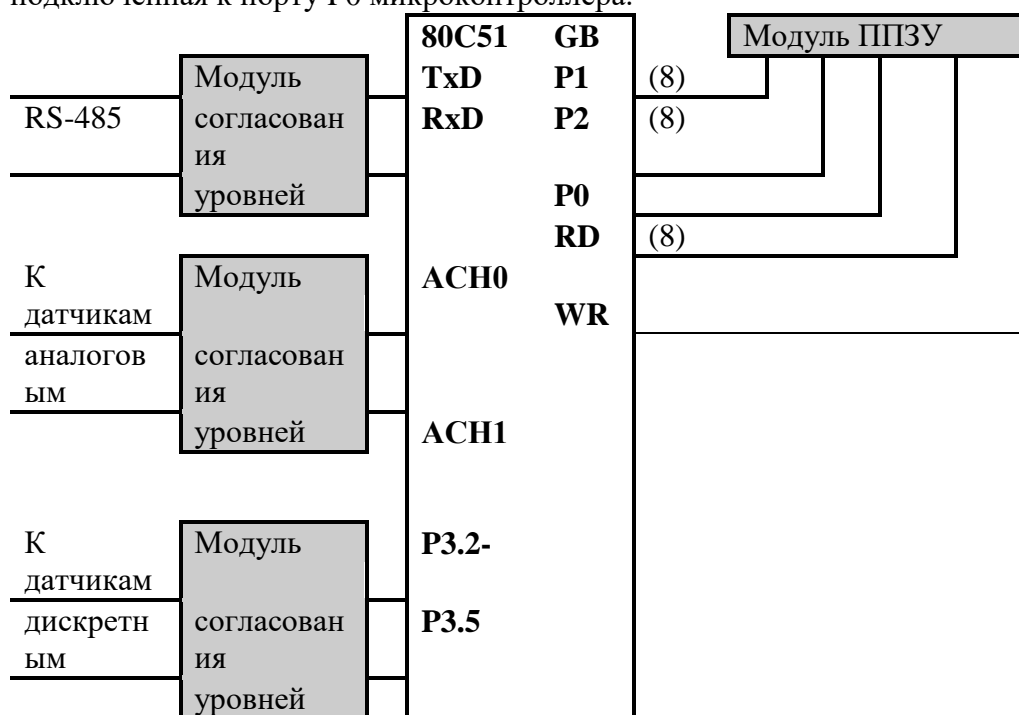


Рис 3.6. Структурная схема ОПК ввода и преобразования сигналов датчиков температуры

Шина адреса подключена к портам P1 и P2, шина управления - порт P3. Локальный интерфейс связи контроллеров RS-485 подключен через модуль преобразования уровней к линиям встроенного интерфейса последовательного обмена. Аналоговые датчики через модуль преобразования уровней подключены к линиям порта P4. Шина адреса включает линии портов P1 (младший байт адреса) и P2 (старший байт адреса), шина данных - порт P0. Дискретные датчики подключены через модуль преобразования уровней к линиям порта P3.

В таблице 3.3 в качестве иллюстрации приведены технические характеристики объектно - ориентированных контроллеров серии E29 [31]. Все ОПК построены на базе однокристальных микроконтроллеров серии K1816.

Например, одноплатный ОПК типа E29.21 представляет собой контроллер - программатор автоклава [32], осуществляющего загрузку - выгрузку продукции стерилизации, вывод на режим по определенной программе, поддержание заданных характеристик стерилизации (температуры, давления, уровня и т.д.), завершение стерилизации (предварительное охлаждение, слив, охлаждение). При этом оператор задает рецепт стерилизации для конкретной продукции, контролирует параметры процесса и работу установки и, в случае необходимости, изменяет характеристики или непосредственно управляет автоклавом на той или иной стадии процесса.

ОПК имеет необходимые для согласования с объектом входные и выходные преобразователи, реализует объектный диалоговый интерфейс с оператором, предоставляя оператору возможность задавать рецепт, осуществлять контроль параметров и вести непосредственное управление автоклавом в терминах технологического процесса. Кроме этого ОПК содержит обширную библиотеку рецептов, организует протоколирование параметров процесса, обладает энергонезависимой памятью, реализует с высокой точностью измерение и регулирование параметров и, наконец, осуществляет самодиагностику. И все это реализовано на одной плате размером 300*200мм, помещенной в корпус и устанавливаемой непосредственно на объекте.

Непрерывное снижение стоимости аппаратных средств ОПК и всё более жесткие требования к программному обеспечению сужают область применения одноплатных ОПК на основе микроконтроллеров.

Наиболее вероятно, что они будут использоваться как встроенные контроллеры датчиков, первичных преобразователей и системы автоматизации несложных технологических объектов. Их место на рынке одноплатных контроллеров занимают ОПК, собираемые на микросхемах с высокой степенью интеграции, имеющие широкие функциональные возможности, приближающиеся к возможностям машин автономного исполнения. Их относят к *ряду универсальных одноплатных ОПК*. Типичный представитель подобных устройств - встроенный одноплатный контроллер EPC-6 фирмы Radisys на базе шины VME [33]. На плате находятся: процессор 80386SX-20МГц, гнездо для математического сопроцессора, память ёмкостью до 4Мбайт на динамических схемах, 8 Кбайтное статическое ОЗУ с батарейным питанием, кэш-память команд и данных ёмкостью 16Кбайт, ППЗУ ёмкостью 512 кбайт на схемах флэш-памяти, блок сопряжения с шиной VME и гнездо шины расширения EXM (компактный вариант шины ISA). Гнездо EXM можно использовать для подключения сетевого адаптера Ethernet, адаптера VGA, дополнительного последовательного порта или модема. Другой пример - одноплатные промышленные контроллеры серии РСМ-3860 фирмы Advantech [34], имеющие аналогичные технические характеристики.

Универсальный одноплатный контроллер позволяет решать все необходимые задачи управления - контроль, индикацию, сигнализацию, регулирование, стабилизацию, управление по критериям оптимальности и т.д. При этом аппаратная часть контроллера должна быть универсальной, что экономически целесообразно при массовом выпуске. Реализация же необходимых алгоритмов управления для каждого типа и вида оборудования происходит на программном уровне записью в энергонезависимое ПЗУ. Универсальность такого контроллера предполагает также возможность обработки и выдачи дискретных и аналоговых входных и выходных сигналов, возможность одновременного формирования логических, программно-временных и непрерывных законов управления.

Стремление производителей сделать такой одноплатный контроллер привлекательно дешевым для пользователя приводит к созданию ряда одноплатных контроллеров, отличающихся в основном узлами объектной ориентации. Подобная унификация целесообразна только в таких классах объектов, в которых из сравнительно небольшого числа разновидностей элементов предстоит проектирование и изготовление большого числа разнообразных систем. Именно эти разновидности элементов и подлежат унификации. Поэтому в общем случае универсальные одноплатные ОПК принципиально обладают

аппаратной избыточностью. Однако выпуск их крупными сериями и наличие стандартных инструментальных средств программирования делают этот вариант автоматизации для ряда технологических объектов экономически привлекательным по сравнению с индивидуальной разработкой специализированных одноплатных ОПК.

Лекция 9. Язык ассемблера для однокристалльного микроконтроллера

В ОПК аппаратные средства и программное обеспечение существуют в форме единого аппаратно-программного комплекса. При проектировании контроллеров приходится решать сложную задачу оптимального распределения функций контроллера между аппаратными средствами и программным обеспечением. Весь цикл разработки ОПК можно рассматривать как последовательность трех этапов:

- анализ задачи управления и определение структуры аппаратных средств;
- разработка прикладного программного обеспечения;
- отладка программного обеспечения в прототипе контроллера.

При решении задач проектирования ОПК наиболее эффективными (по быстродействию и компактности) являются программы, написанные на архитектурно-ориентированном языке - ассемблере. Языки такого типа включены в стандарт по языкам программирования контроллеров под названием "языков инструкций" для создания эффективных оптимизированных программ управления технологическими процессами.

После определения структуры контроллера наступает стадия разработки программного обеспечения, которая, в свою очередь, разбивается на два различных этапа:

- постановка задачи программы и разработка исходного текста программы;
- получение программы в машинных кодах для работы в микроконтроллере.

Первый этап практически не поддается формализации и не может быть автоматизирован. Именно на этом этапе разработчик сталкивается с наибольшим количеством трудностей. С другой стороны, качество разработанной программы определяется решениями, принятыми именно на этом этапе.

Второй этап, как правило, отработан достаточно хорошо в системах разработки программного обеспечения ПК в виде текстовых редакторов, трансляторов с языка ассемблера, программ-документаторов, программ-отладчиков, работающих на персональных ЭВМ в различных операционных средах. Такие ЭВМ, используемые на втором этапе и, как правило, не совместимые по системе команд с ПК, называют инструментальными ЭВМ.

Язык "символического кодирования" - ассемблер учитывает архитектуру однокристалльных микроконтроллеров с позиции их программирования. В качестве иллюстративной базы воспользуемся семейством *однокристалльных микроконтроллеров MCS-51* [14,30], созданных фирмой Intel. Архитектура MCS-51 рассмотрена в разделе 3.3. Подробно система команд MCS-51 приведена в прил. 4. Различные модификации микроконтроллеров для программиста отличаются только количеством регистров специальных функций, оставляя неизменной систему команд.

Система команд микроконтроллеров MCS-51

Система команд предоставляет большие возможности обработки данных, обеспечивает реализацию логических, арифметических операций, а также управление в режиме реального времени. В системе команд рассматриваемого микроконтроллера, наряду с командами, выполняющими действия с байтами данных, есть большое количество битово-ориентированных команд и адресуемых битовых полей в памяти данных, позволяющих эффективно работать с дискретными датчиками и исполнительными механизмами. Все программно-доступные узлы микроконтроллера сведены в специальную область памяти данных - регистры специальных функций (SFR), что позволяет адресоваться к ним также, как и к ячейкам памяти.

С позиций программирования все основные структурные элементы рассматриваемого семейства: АЛУ, ПЗУ, ОЗУ, регистры специальных функций, внешние шины - имеют

байтовую организацию. Двухбайтовые данные используются только регистром-указателем и счетчиком команд. Все данные хранятся в резидентном (расположенном на кристалле) ОЗУ ёмкостью 128 байт и во внешнем ОЗУ, подключаемом к микроконтроллеру через системную шину, организованную на выводах порта P0 (см. гл. 3.3). Суммарный размер внешней памяти команд и данных может достигать 128 Кбайт.

Важным элементом программной структуры MCS-51 является аккумулятор А - служебная 8-разрядная ячейка памяти, используемая в качестве регистра операнда и регистра результата.

Синтаксис большинства команд ассемблера состоит из *мнемонического обозначения функции*, выполняемой командой, вслед за которым идут *операнды*, указывающие *методы адресации и типы данных*. Различные типы данных или режимы адресации определяются установленными операндами, а не изменениями мнемонических обозначений команды.

Набор команд MCS-51 поддерживает следующие **методы адресации**:

непосредственная адресация, когда сам 8-битный операнд (константа) располагается непосредственно в команде, например:

MOV A,#09H

Здесь MOV – обозначение функции команды (в данном случае – передача данных); А - операнд-приёмник (в данном случае - аккумулятор), в котором находятся данные после выполнения команды; #09-числовая константа;

прямая адресация, когда адрес операнда содержится в самой команде (указывается, как номер ячейки резидентного ОЗУ, или как один из 128 прямо адресуемых битов, расположенных в ячейках ОЗУ с адресами 20H-2FH), например:

MOV A,09

Здесь 09-адрес ячейки ОЗУ, в которой находятся данные, пересылаемые в аккумулятор;

регистровая адресация, для обращения к восьми регистрам выбранного банка регистров. Регистры - это 16 ячеек памяти, обозначаемых именами с R0 ... R7 и объединенных в два банка RB0 и RB1 (по 8 регистров в каждом). В программе специальной директивой прямо указывается, с каким банком регистров в данный момент надо работать. Пример:

MOV A,R5

Здесь R5-операнд-источник данных (регистр R5);

косвенно-регистровая адресация, используемая в основном для обращения к ячейкам внешнего ОЗУ, хотя применима и к резидентному ОЗУ. Адрес операнда в этом режиме адресации указывается косвенно, как содержимое индексного регистра. В качестве индексных регистров могут использоваться регистры R0 или R1:

MOV A,@R0

@-обозначение косвенного метода адресации. В регистре R0 содержится адрес операнда-источника данных.

Систему команд можно условно разбить на четыре группы:

- команды передачи данных;
- арифметические команды;
- команды логических операций;
- команды передачи управления.

Команды передачи данных

Большую часть команд данной группы составляют команды передачи и обмена байтов. Команды пересылки бит представлены в группе команд битовых операций (см. прил.4). Возможный обмен

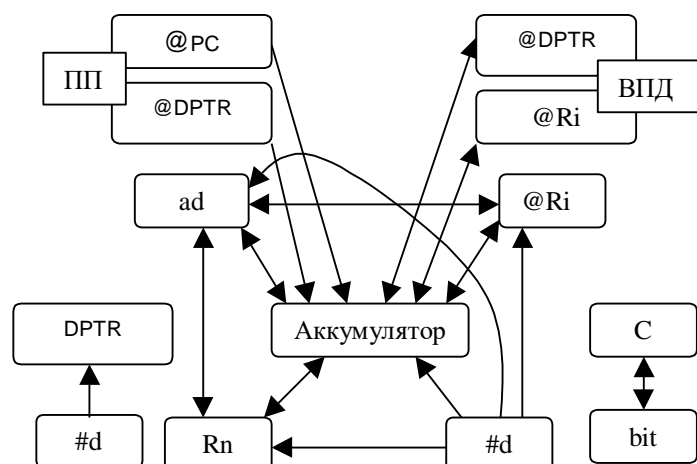


Рис.4.5. Пути передачи данных в MCS-51.

PC-счетчик адресов программы; DPTR-регистр-указатель данных; ad-адрес байта памяти; Ri-индексный регистр; Rn-регистр общего назначения; #d-числовая константа; C-разряд переноса при выполнении арифметических и сдвиговых операций; bit-бит информации. ПП-память программ, ВПД-внешняя память данных.

данными между различными операндами представлен в виде графа на рис.4.5. Ниже приведены примеры команд передачи данных:

```
MOV A,#10 ; передача в аккумулятор
MOV A,#0CFH ; константы
MOV R2,A ; передача данных в регистр общего назначения
MOV R5,#10101010B
MOV R1,#10 ; передача косвенно - регистровым методом адресации
MOV @R1,#0EEH ; в индексном регистре R1 - адрес операнда-приемника
MOV A,@R1
Примеры использования прямой адресации в командах передачи данных:
MOV 30H,R7
MOV 31H,@R0
MOV @R1,31H
MOV 80H,37H
```

Арифметические команды

В наборе команд имеются следующие арифметические операции: сложение, сложение с учетом флага переноса, вычитание с займом, инкрементирование, декрементирование, сравнение, десятичная коррекция, умножение, деление. Все арифметические операции, кроме инкрементирования и декрементирования, влияют на определенные биты: переполнения, переноса, промежуточного переноса, бит четности в *слове состояния процессора* – специальной ячейке резидентного ОЗУ.

Пример сложения содержимого регистра R0 и аккумулятора A:

```
ADD A, R0
```

То же, но с учетом флага переноса:

```
ADDC A, R0 ; учет результата сложения младший байт
```

Умножение числа 12 на константу 173:

```
MOV A,#12
```

```
MOV B,#173 ; загрузка множителя
```

```
MUL AB ; умножение
```

В результате, в B - младший байт произведения, в A старший байт произведения.

Команды логических операций

Система команд микроконтроллера позволяет реализовать логические операции “И”, “ИЛИ”, “ИСКЛЮЧАЮЩЕЕ ИЛИ” на регистре-аккумуляторе и байте-источнике, в качестве которого может быть регистр общего назначения, прямо или косвенно адресуемая ячейка памяти, константа.

Примеры операции "И":

```
ANL A,R3
```

```
ANL A,#0
```

```
ANL 20H,#15
```

Примеры операции "ИЛИ":

```
ORL A,@R0
```

```
ORL 30H,#55H
```

Примеры операции "ИСКЛЮЧАЮЩЕЕ ИЛИ":

```
XRL A,#30H
```

```
XRL A,R3
```

```
XRL 30H,#55H
```

Существуют логические операции, выполняемые только на одном аккумуляторе: сброс и инвертирование содержимого аккумулятора, циклический сдвиг влево и вправо, обмен местами старшей и младшей тетрад аккумулятора:

Пример команды:

```
CLR A ; сброс аккумулятора
```

Команды передачи управления

К данной группе команд относятся команды, обеспечивающие условное и безусловное ветвление программы, вызов подпрограмм и возврат из них, а также команда пустой операции NOP. Можно выделить три разновидности команд ветвления:

длинный переход и вызов подпрограмм, позволяющие осуществлять переход в любую точку адресного пространства памяти программ объемом 64 Кбайт. Длинный переход содержит в мнемокоде программы букву L:

LJMP START ; переход к строке программы с меткой START

LCALL TIME ; вызов подпрограммы TIME

абсолютный переход, работающий в пределах одной страницы памяти программ размером 2048 байт. Команда абсолютного перехода занимает в памяти программ меньше места, чем длинный переход. Такие команды содержат в мнемокоде букву A:

AJMP LAB1

ACALL.SSHIFT

относительный переход, обеспечивающий передачу управления в пределах $-128 \div +127$ байт относительно адреса следующей команды. Существует одна команда безусловного короткого перехода SJMP.

Развитая система условных переходов позволяет осуществлять ветвление по состоянию "0" или "1" прямо адресуемых битов, по равенству или неравенству нулю аккумулятора.

Полный перечень команд приведен в прил.4.

Лекция 11. Программирование специализированных технологических контроллеров

Программирование специализированных технологических контроллеров (СТК) осуществляется непосредственно на самом СТК с помощью встроенного или подключаемого (или и то и другое вместе) пультов. Программирование СТК сводится к объединению готовых программ, находящихся в ПЗУ, в одну общую программу с помощью специальных процедур кодирования и конфигурирования. Рассмотрим особенности программирования СТК на примере широко распространенного в России регулирующего микроконтроллера типа Ремиконт Р-130 [3,28,49]. При наличии обширной библиотеки готовых программ программирование позволяет сравнительно просто реализовывать в микроконтроллере Р-130 разнообразные алгоритмы регулирования, расчета систем управления, имитационного моделирования динамических систем.

Как уже отмечалось ранее в гл. 3, специализированные технологические контроллеры могут быть представлены в виде виртуальных структур, реализующих конкретные алгоритмы управления технологическими процессами. Часть элементов такой структуры организуется аппаратно, часть программно. Но неизменно контроллер в целом представляет пользователю (в частности, инженеру по автоматизации) возможность взаимодействовать со средствами контроллера в привычной для него среде. Как объект исследования Ремиконт Р-130 организует доступ пользователя к структуре и параметрам алгоритма управления каким-либо параметром или функцией технологического процесса. Поэтому средства его программирования используют терминологию, математический аппарат и инструментарий, соответствующий решению задачи пользователем на уровне описания алгоритма управления.

Средства программирования Ремиконт Р-130 включают пульт настройки, с помощью которого реализуется программирование, библиотеку алгоритмов, размещенную в ПЗУ и недоступную для корректировки пользователю, и, наконец, алгоблоки (структурированное ЭнОЗУ), в которые с помощью пульта настройки и заносятся алгоритмы из библиотеки в процессе программирования (см. рис.4.8).

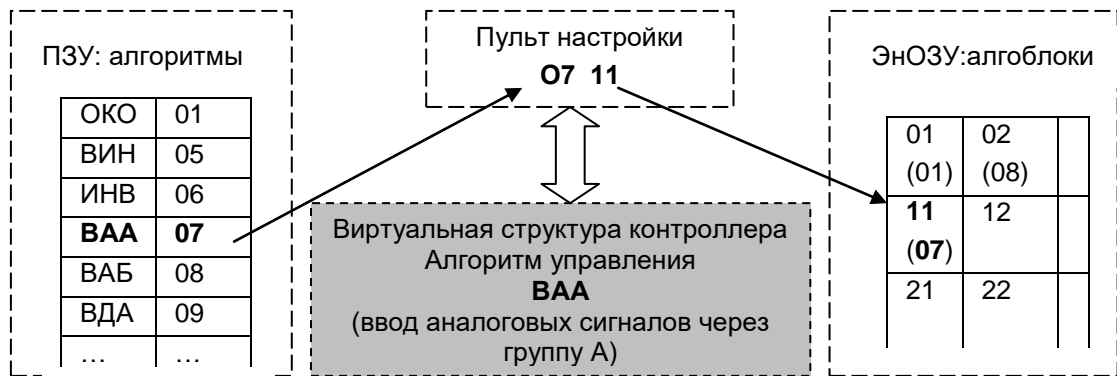


Рис.4.8. Схема технологического программирования контроллеров Р-130

Язык технологического программирования имеет два пользовательских слоя. Один - для представления инженеру по автоматизации информации об алгоритме управления объектом в виде схемы из библиотечных алгоритмов. Именно благодаря такому представлению и формируется виртуальная структура контроллера в виде алгоритма. Второй, в виде двузначных кодов, используемых при вводе программы с помощью пульта настройки в контроллер, является своеобразной платой за примитивный пульт настройки. Поэтому пользователю приходится оперировать одновременно удобными и понятными аббревиатурами типа ВАА (ввод аналоговых сигналов группы А) и цифрами (соответственно 07).

Любая программа, реализуемая с помощью языков технологического программирования, содержит ряд обязательных алгоритмов, например, для микроконтроллера Ремиконт Р-130 при реализации функции регулирования:

- ОКО (оперативный контроль регулирования);
- ЗДН (изменение задания регулятору);
- РУЧ (ручное управление);
- ВАА (ввод аналоговых сигналов группы А);
- АВА (вывод аналоговых сигналов группы А);
- РАН (регулирование аналоговое) или РИМ (регулирование цифровое).

Перечисленные алгоритмы, наряду с алгоритмами лицевой панели, ввода - вывода, регулирования, динамического преобразования, статического преобразования, аналого - дискретного преобразования, логических операций, дискретного управления и группового непрерывно - дискретного управления, реализованы в виде программных модулей и размещены в первом разделе алгоритмов резидентной библиотеки микроконтроллера Ремиконт Р-130, во второй раздел которой включены алгоритмы стандартных конфигураций регулятора аналогового РЕГА и регулятора импульсного РЕГИ (прил. 6).

В общем случае алгоритм имеет входы, выходы и функциональное ядро. Входы алгоритма подразделяют на сигнальные (принимающие сигналы для обработки в соответствии с функцией алгоритма) и настроечные (управляющие параметрами настройки). На выходе алгоритма формируются сигналы, являющиеся результатом обработки алгоритмом входных сигналов. Библиотечный алгоритм имеет три реквизита: библиотечный номер, модификатор и масштаб времени.

Библиотечный номер является основным параметром, характеризующим свойства алгоритма, и обозначается двузначной цифрой.

Модификатор обычно задает число однотипных операций, которые может выполнять

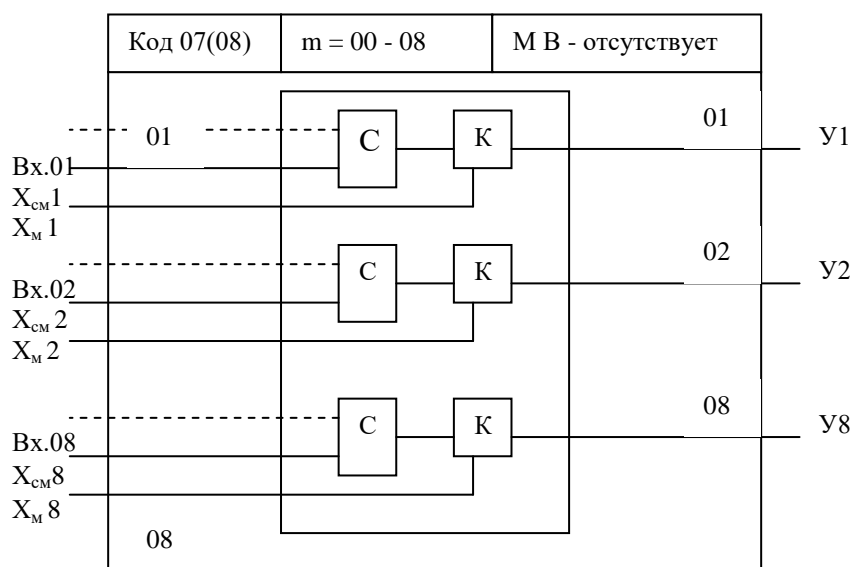


Рис. 4.9. Функциональная схема алгоритма ВАА (ВАБ)

один алгоритм. Например, в сумматоре модификатор задает число суммирующих входов, в программном задатчике - число участков программы и т.п.

Масштаб времени задает размерность для временных сигналов или параметров в алгоритмах, работающих в режиме реального времени (секунды - минуты или минуты - часы).

На рис.4.9 представлено графическое изображение алгоритма ввода аналоговых сигналов.

Алгоритм применяется для связи функциональных алгоритмов с аппаратными средствами аналогового ввода. Для связи с аналоговыми входами группы А и Б используются соответственно алгоритмы ВАА и ВАБ. Каждый алгоритм обслуживает до 8 аналоговых входов (8 каналов). Число каналов задается модификатором m. Каждый канал связан с соответствующим по номеру аналоговым входом контроллера. К входному аналоговому сигналу добавляется сигнал смещения X_{см}, полученная сумма умножается на коэффициент К. Выходной сигнал канала равен $Y = (X_{вх} + X_{см}) * K$. Время обслуживания одного канала аналогового ввода равно 40 мс. Масштаб времени отсутствует.

Своеобразный **синтаксис языка технологического программирования** представляет собой свод правил и процедур программирования, то есть правил размещения алгоритмов в алгоблоках и взаимосвязи алгоблоков, а также процедур ввода информации в контроллер для его программирования.

Размещение алгоритмов в алгоблоках подчиняется двум правилам:

- Любой алгоритм можно помещать в любой по номеру алгоблок.
- Один и тот же алгоритм можно помещать в разные алгоблоки.

Исключение делается для алгоритмов ОКО (ОКЛ), помещаемых только в первые четыре алгоблока в соответствии с номером контура, обслуживаемого данным алгоритмом ОКО (ОКЛ). Для алгоритмов ввода - вывода информации имеется ограничение на кратность их использования в пределах контроллера. Возможности конфигурирования, то есть связывания алгоблоков, не зависят от алгоритма, помещаемого в алгоблок, и определяются следующими правилами:

- Любой вход любого алгоблока можно связать с любым входом выходом алгоблока или оставить свободным.
- На любом свободном входе любого алгоблока можно вручную задавать сигнал в виде константы (в режиме программирования) или коэффициента (в режиме программирования или работы).
- На любом входе любого алгоблока сигнал можно инвертировать.

В исходном состоянии все входы алгоблоков являются свободными; на них заданы константы, значения которых зависят от алгоритма; инверсия отсутствует. В процессе конфигурирования для связываемых входов - выходов алгоблоков задаются номера алгоблока и номера входов - выходов, а для свободных входов - является ли сигнал

константой или коэффициентом. Таким образом, осуществляется взаимное соединение входов-выходов алгоблоков или, что, то же самое, записанных в них алгоритмов в единую структуру, реализующую математическую операцию или задачу управления.

Обслуживание алгоблоков ведется циклически с постоянным временем цикла, последовательно с первого по последний алгоблок. В оставшееся после обслуживания алгоблоков время цикла выполняется самодиагностика контроллера.

Программирование Ремиконт Р-130 состоит из выполнения следующих этапов:

- Составление схемы размещения алгоритмов в алгоблоках на основании общего алгоритма управления технологическим объектом и перечня в библиотеке алгоритмов.
- Построение блок - схемы взаимосвязей алгоблоков в соотношении со структурой программируемой задачи и общим алгоритмом управления технологическим объектом.
- Составление программы задачи.
- Введение программы в память в режиме программирования.

Составим схему размещения алгоритмов в алгоблоках для контроля параметра технологического процесса и построим блок - схему взаимосвязей алгоблоков в соответствии со структурой программируемой задачи.



Рис.4.10. Схема размещения набора алгоблоков

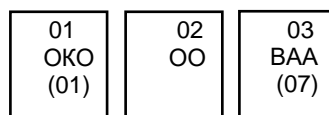


Рис.4.11. Блок-схема взаимосвязи алгоблоков

Программа должна осуществлять контроль изменения аналогового входного сигнала, характеризующего параметр технологического объекта и выводить значение параметра на индикатор лицевой панели контроллера для наблюдения оператором. На рис.4.10 приведена схема размещения набора алгоблоков такой программы, включающая набор алгоблоков, их обозначение и код алгоритмов: в частности, в алгоблок 01 помещен алгоритм ОКО - номер 01 в библиотеке алгоритмов, алгоблок 02 - свободен (например, зарезервирован для второго контура регулирования), в алгоблок 03 задействован алгоритм ВАА - номер 07 в библиотеке алгоритмов.

На основе параметров используемых алгоритмов строится блок-схема соединения связанных входов и выходов алгоблоков (рис.4.11).

Для составления программы предварительно по блок-схеме определяются номера алгоблоков-приемников и алгоблоков-источников информации. Поскольку некоторые алгоблоки могут одновременно являются и приемниками и источниками, то их номера в программе повторяются. Пример составления и оформления подобной программы в соответствии с блок-схемой на рис.4.11. представлен в табл.4.1. По существу программой является последовательность информации для занесения в память микроконтроллера с помощью пульта настройки (ПН) при программировании.

Программа одноконтурной АСР

Таблица 4.1

№ алго- блока прием.	Алго- ритм и код	№ входа алгоблока приемника	№ алго- блока источ.	Алго- ритм и код	№ выхода алгоблока источника	Представление на ЦИ ПН		
01	ОКО (01)	02	03	ВАО (07)	01		01	02
						,	03	01

Программирование Ремиконт Р-130 проводится с помощью пульта настройки и, несмотря на свою простоту, требует проведения в строгой последовательности ряда подготовительных этапов, связанных с выполнением 8 следующих процедур: тестирование; установка приборных параметров; задание системных параметров; размещение алгоритмов в алгоблоках; конфигурирование; настройка параметров алгоблоков; задание начальных условий; работа с ППЗУ.

В процедуре тестирования проверяется работоспособность устройств микроконтроллера - ПЗУ, ОЗУ, интерфейсный канал, пульт настройки, лицевая панель и т.д. Процедура "приборные параметры" предназначена для задания общих параметров контроллера, таких как комплектность, диапазон времени, "обнуление", время цикла и другие. В процедуре "алгоритм" устанавливаются коды алгоритмов и параметры-модификаторы, масштаб времени. При выполнении процедуры конфигурирования задаются взаимосвязи алгоблоков между собой и входными (выходными) устройствами микроконтроллера. При настройке устанавливаются значения параметров алгоритмов - константы и коэффициенты. В процедуре "начальные условия" задаются значения сигналов на выходах алгоблоков, с которыми эти блоки начнут работать при переходе в рабочий режим. Процедура "системные параметры" дает возможность установить логические номера контроллеров в локальной сети "Транзит" и режим работы интерфейса. Процедура "работа с ППЗУ" выполняет запись в память составленной программы и обратно в ОЗУ.

Собственно программирование начинается с установки приборных параметров. Выполняемая в процедуре "приборные параметры" последовательность операций и характеризующие их параметры, являются общими для всех алгоблоков, т.е. для контроллера в целом. В процедуру "приборные параметры" входят следующие операции: обнуление контроллера (или установка стандартной конфигурации); установка комплектности; установка кода запрета изменения параметров и диапазона временных параметров; задание времени цикла работы контроллера; контроль ресурса 1-й и 2-й областей ОЗУ; контроль состава и версии библиотеки алгоритмов.

В первой операции обнулению соответствует код 00, стандартной конфигурации - аналоговое ПИД-регулирование - код 01 и импульсное регулирование - 02. В операции "комплектность" устанавливается код, определяемый модификацией контроллера, в зависимости от числа входных и выходных аналоговых и дискретных сигналов. В операции "установка кода запрета" используются код запрета -00 и код разрешения изменения параметров -01. Диапазоны временных параметров задаются следующими кодами: 00 - с, мин (младший диапазон), 01-мин, ч (старший диапазон). Это означает, что все временные параметры, например, время интегрирования Тп, дифференцирования Тд, постоянная времени фильтрации Тф измеряются в заданном масштабе. Операция "задание времени цикла" позволяет устанавливать одинаковое для всех алгоблоков контроллера время цикла в диапазоне 0.2 - 2 с с шагом 0.2 с.

В процедуре "алгоритм" происходит "заполнение" алгоблоков программами из библиотеки алгоритмов. В общем случае для каждого алгоблока задаются 3 параметра: код алгоритма, его модификатор (М) и масштаб времени (МВ). В некоторых алгоритмах модификатор и (или) масштаб времени могут отсутствовать. После обнуления контроллера алгоблоки можно заполнять только подряд, начиная с первого. Если какие-либо алгоблоки нужно зарезервировать для последующего расширения алгоритмической структуры, в них следует ввести алгоритм с кодом 00.

В процедуре конфигурирования осуществляется программирование заданного взаимного соединения входов-выходов алгоблоков в единую структуру. Алгоблок, принимающий на входе информацию, называется алгоблок-приемник, передающий информацию, - алгоблок-источник. Входы алгоблока-приемника подразделяются на 2 типа: связанные и свободные. Связные входы подключаются к одному из выходов алгоблока-источника. На свободных входах задаются параметры настройки, которые также делятся на 2 группы: константы и коэффициенты. Константы устанавливаются в режиме программирования и не могут изменяться (но могут контролироваться) в режиме работы. Коэффициенты устанавливаются и могут изменяться как в режиме программирования, так и в режиме работы. Конкретные значения параметров на свободных входах алгоблока задаются в процедуре "настройка". При конфигурации также определяются, поступает ли сигнал, значение параметра прямо или инверсно, т.е. с изменением знака или нет. Определение алгоблоков-приемников и источников зависит от структуры программируемой задачи; определение типа входа - связный, свободный - следует из описания алгоритмов. Необходимо также отметить, что некоторые алгоритмы могут не иметь конфигурируемых входов или выходов, например, алгоритмы ввода и вывода информации (ВАА и АВА). В процедуре настройки параметров устанавливаются числовые значения параметров на свободных входах алгоблока.

Режим "работа" контроллера содержит процедуры, предназначенные для контроля параметров и сигналов: процедура обнаружения ошибок и отказов; контроль приборных параметров; контроль входных сигналов; контроль выходных сигналов; контроль и изменение параметров алгоблоков; калибровка сигналов.

Последовательность выполнения операций в перечисленных процедурах приведена в прил.7.

Таким образом, программирование с помощью встроенных средств (библиотека алгоритмов и алгоблоки) и пульта настройки позволяет сравнительно просто реализовывать в микроконтроллере Р-130 разнообразные алгоритмы управления. При этом доступ пользователя к структуре и параметрам алгоритма управления, каким - либо параметром или функцией технологического процесса, ведется в привычной среде (с использованием терминов и математического аппарата на уровне описания алгоритма управления).

Лекция 12. Универсальные модульно - адаптируемые контроллеры

Одним из наиболее эффективных методов получения оптимального соотношения стоимость / производительность для встраиваемого контроллера является задание конфигурации контроллера в процессе его наладки самим пользователем. Архитектура встраиваемого контроллера должна предоставлять такую возможность. Общее название класса таких контроллеров - универсальные модульно-адаптируемые. Конструктивно архитектура этого класса контроллеров реализует доступ к функциональным модулям на основе мезонинных технологий или с помощью монтажных каркасов.

Наглядным воплощением идеи мезонинных технологий является встраиваемый одноплатный компьютер MVME162 фирмы Motorola. Особенностью этого мощного одноплатного компьютера является возможность широкого выбора требуемых характеристик и функциональных частей за счет использования стандартных мезонинных технологий ввода/вывода. Мезонинная технология подразумевает наличие специальной локальной шины на плате компьютера и набора функциональных модулей расширения [35,36]. Современный мезонинный модуль имеет небольшие размеры (обычно 45 x 99мм) и два разъема. Один разъем служит для подключения к локальной шине платы-носителя, атрибуты другого определяются функциональной принадлежностью платы. На таких платах можно разместить многоканальный 16-разрядный АЦП, 40 каналов ввода/вывода, многоканальный интерфейс RS-232C/ 422/ 485, модуль оперативной или постоянной памяти, математический сопроцессор и т.д. Сейчас известно около 10 мезонинных технологий.

Наибольшую популярность получили модули фирм PEP Modular Computer - MODPACK и фирмы Industry Pack - IP. В настоящее время фирмы-производители IP- контроллеров выпускают более 235 моделей, которые могут использоваться практически в любой отрасли промышленности. Платы-носители мезонинных модулей IP существуют практически для всех магистрально-модульных интерфейсов: VMEbus, PCI, VXI, VESA, ISA, G64/G96, Nubus. Плата-носитель может иметь собственные процессоры или нет. Беспроцессорные платы предназначаются, как правило, для простого наращивания каналов ввода/вывода. Типичная плата-носитель VME форматом 6U содержит 4 мезонинных модуля. Чтобы оценить возможности такой конструкции достаточно перечислить атрибуты одноплатного компьютера, собранного из стандартных IP- модулей: CPU, ОЗУ, Flash ППЗУ, Ethernet, SCSI, VME64 интерфейс, 2 канала RS-232C, набор таймеров. Учитывая открытый характер мезонинных спецификаций, можно разработать любую плату-носитель IP со встроенным процессором любого типа. При этом можно создать контроллер, наиболее полно соответствующий классу решаемых задач.

Как правило, вместе с мезонинными платами поставляют драйверы операционных систем, которые обычно работают на платах-носителях. Это операционные системы реального времени - OS9, pSOS+, VMEexec, VxWorks, LynxOS и традиционные - MS-DOS, Windows, Windows 95, UNIX, OS7.

К новейшим технологиям мезонинных модулей относятся, прежде всего мезонины, использующие локальную шину PCI. Они получили название PMC (PCI+MEZZANIN CARD = PMC) и используются, в основном, как стандартное расширение одноплатных VME-контроллеров, построенных на базе семейства RISC - процессоров: PowerPC, Alpha, Mips R4600 и др. Номенклатура модулей PMC пока невелика и содержит только скоростные интерфейсы, но она быстро расширяется.

Если мезонинные технологии еще только завоевывают рынок (хотя и успешно), то традиционно модульно адаптируемые промышленные контроллеры и компьютеры строятся с помощью монтажных каркасов. Пример - MicroPC, ADAM-5510, MiPC-50, MIC-2000, PR - 100, DV 105, DL 105 или отечественные серии Ломиконт, TCM51, E29.17.

Монтажные каркасы обычно представляют собой пассивные 4- или многослойные платы с несколькими слотами для установки процессорных или согласующих модулей. Магистраль, объединяющая модули, может быть из ряда универсальных (например, уже упомянутые VMEbus, PCI, VXI, VESA, ISA, G64/G96, Nubus) или закрытая шина данной фирмы. Ранее уже отмечалось, что перечень модулей расширения для промышленных контроллеров огромен, так как каждая фирма-производитель стремится максимально удовлетворить запросы пользователя. Часть фирм-производителей предлагает пользователю, наряду с индивидуальными каркасом и шиной, также и индивидуальный конструктив модулей. Такой подход ограничивает возможности аппаратной адаптации только перечнем модулей, предлагаемым такими фирмами. Чаще всего перечень невелик и, в свою очередь, ориентирован на конкретные области применения.

Например, **контроллер серии ADAM-5510** фирмы Advantech предназначен для работы в территориально распределенных системах управления и сбора данных в качестве автономного контроллера [37]. Он работает с аналоговыми и дискретными сигналами, производит первичную обработку информации. Имеется возможность гибко конфигурировать состав системы в зависимости от вида и количества контролируемых параметров. Устройства этой серии легко объединяются в многоточечную сеть на базе интерфейса RS-485. Системное ядро, несмотря на небольшие размеры, предоставляет пользователю много возможностей. Модульная конструкция, в свою очередь, обеспечивает широкие возможности при конфигурации системы.

Системный блок ADAM-5000 (рис 3.9) содержит плату центрального процессора на базе 16-разрядного микропроцессора 80188 с ОЗУ объемом 32 Кб и флэш-ПЗУ объемом 128 Кб для записи управляющей программы, преобразователь питания, 4 слота расширения для

подключения дополнительных модулей, сторожевой таймер, встроенный коммуникационный порт RS-232 и 1 встроенный коммуникационный порт RS-485.

Модули ввода/вывода. Одна плата центрального процессора ADAM-5000 может поддерживать до 64 каналов ввода/вывода в четырех слотах расширения. Эти каналы могут быть назначены как входные или выходные. Серия ADAM-5000 содержит различные типы модулей ввода/вывода. Модуль дискретного ввода ADAM-5051 имеет 16 каналов и работает с сигналами до 30 В постоянного тока. У модуля дискретного вывода ADAM-5056 -также 16 каналов вывода с открытым коллектором, коммутирующих цепи с напряжением до 30В и мощностью до 450 мВт, что позволяет подключать к нему различные типы исполнительных устройств. Специальный модуль релейных выходов ADAM-5060 имеет 6 каналов выходных реле, предназначенных для коммутации цепей постоянного и переменного тока мощностью до 60 Вт. Модуль ввода аналоговых сигналов ADAM-5017 обеспечивает ввод до 8 аналоговых сигналов напряжения в диапазоне от ± 150 мВ до ± 10 В или тока от 0 до 20 мА. Специально для работы с термопарами предназначен модуль аналогового ввода ADAM-5018. Он позволяет подключить до 8 стандартных термопар различного типа. Аналоговые модули имеют разрешение 16 разрядов и могут быть запрограммированы пользователем на работу в различных диапазонах сигналов. Для формирования выходных аналоговых сигналов в диапазоне от 0 до 10 В или от 0 до 20 мА используется 4-х канальный модуль аналогового вывода ADAM-5024.

Рассмотрим более подробно в качестве примера модульно-ориентированных универсальных контроллеров семейство контроллеров MicroPC.

Семейство контроллеров MicroPC - это семейство PC-микроконтроллеров, процессорных плат и модулей расширения с шиной ISA IBM PC- совместимых компьютеров, ориентированное на промышленное применение [25,26]. Фирма-производитель - Octagon Systems (США).

Коротко о MicroPC:

- * Резидентная операционная система MS-DOS 6.0. ПЗУ воспринимается, как электронный диск, поэтому загрузка и работа операционной системы происходит так же, как и в настольном ПК. Совместима с Windows и QNX.
- * Три электронных диска на процессорных платах.
- * Возможность автономной работы без видеомонитора, клавиатуры и дисковых накопителей.
- * Конструктивная совместимость с IBM PC, что позволяет устанавливать платы MicroPC в любой персональный компьютер.
- * Малая потребляемая мощность - отсутствие принудительного охлаждения. Большинство плат семейства работают от одного источника питания 5В.
- * Рабочий диапазон температур от -40° С до +85° С.
- * Выдерживают перегрузки до 5g при вибрации и 20g при ударе.

В настоящее время существует 27 типов плат расширения. В одной системе можно установить до 8 плат расширения. Все платы расширения могут быть вставлены в пассивную объединительную панель или монтажный каркас MicroPC, в настольный ПК, в ПК промышленного назначения.

Пассивная объединительная панель позволяет объединить до 8 модулей без использования монтажного каркаса. Имеет два разъёма для подключения источника питания. 8/16-битная пассивная объединительная панель позволяет объединять 8- и 16-битные модули. Каждая панель имеет одинаковое количество 8- и 16-битных слотов. Выпускаются модификации для объединения 4, 6 и 8 модулей.

Монтажные каркасы служат для размещения и объединения модулей MicroPC. Внешний вид модуля семейства MicroPC представлен на рис.3.11. Каркас серии 52xx позволяет объединять до 8 модулей. Монтажный каркас серии 72xx служит для объединения 8/16-битных модулей MicroPC. Монтажные каркасы предназначены для установки в стойку автоматики типа NEMA или для крепления на стену. Некоторые модули предназначены для

настолярного размещения. Дешёвые монтажные каркасы 52хх предназначены для систем, предъявляющих повышенные требования к объёму, занимаемому системой управления. Уобразная система крепления модулей позволяет эксплуатировать их в условиях повышенных вибраций и частых ударов. Более подробно монтажные каркасы серий 52хх, 727х. описаны в прил. 2.

Процессорные модули семейства MicroPC охватывают широкий спектр приложений: от однокристальных микроконтроллеров, ориентированных на локальное управление простым устройством, до промышленных файл-серверов, ориентированных на управление локальной сетью ПК или на управление сложным и динамичным объектом. Все модули имеют в своём составе сторожевые таймеры, предназначенные для автоматического восстановления работоспособности системы управления в случае сбоя или “зависания” управляющей программы. Основные характеристики процессорных модулей приведены в прил.2. Эти модули перекрывают широкий диапазон от модулей типа 7000 и 5025х, ориентированных на высокопроизводительные, встраиваемые системы управления (работающие в среде Windows и QNX), до дешёвых микроконтроллеров со средней производительностью типа 6012 и 6024, предназначенных для малых систем сбора данных и управления. Для управления локальными сетями промышленных контроллеров и сложными объектами, требующими передачи и обработки больших массивов данных, предлагаются модули промышленных файл-серверов типа PC-450 и PC-425.

Платы управления PC используются в прикладных программах, где требуется совместимость PC или высокая эффективность (5012 или выше). Совместимость PC часто необходима, чтобы воспользоваться преимуществом дешевого доступного программного обеспечения. Система PC будет стоить больше чем один микроконтроллер, но Вы имеете доступ к видео и дисковой памяти (хранению) и другим функциям.

Модули ввода/вывода и модули, предназначенные для реализации специальных функций, позволяют организовать объектную или интерфейсную связь контроллера, осуществить дополнение его структуры необходимыми средствами памяти, усиления развязки и т.п. Кроме модулей, объединяемых с помощью стандартного 8/16 битного интерфейса ISA, фирма MicroPC выпускает целый ряд дополнительных модулей расширения, которые подключаются к этим модулям с помощью гибкого кабеля и обеспечивают согласование входных и выходных сигналов системы управления по напряжению, току, мощности, по типу коммутационных элементов и т.д.

Практически любая система управления требует наличия одной или нескольких линий **дискретного ввода/вывода**. Все линии дискретного ввода/вывода можно разбить на 3 базовых класса в зависимости от требований к интерфейсу взаимодействия с внешними устройствами: логический уровень (0-5 В), также называемый ТТЛ уровнем; значения тока и напряжения среднего уровня (до 50 В постоянного тока и тока до 100 мА); значения тока и напряжения высокого уровня (до 260 В постоянного и переменного тока и тока до 3 А). Схемы управления большинства промышленных компьютеров используют уровень 0-5 В. Этот интерфейс ввода/вывода позволяет работать с такими устройствами, как внешние переключатели (контактные датчики), простые логические устройства, светодиоды и другие слаботочные устройства. Когда в системе используются программируемые логические контроллеры или другие устройства с 12-вольтовой логикой, необходимо включить преобразователь логических уровней для трансляции уровней сигналов в диапазон 0-5 В. Буферизированный таким образом выходной сигнал компьютера может управлять малогабаритными реле, соленоидами и лампами. Управление высоковольтными цепями постоянного и переменного тока не только требует усиления управляющих сигналов, но и предъявляет высокие требования к изоляции выходных цепей компьютера. Этого можно достичь, используя интерфейсные модули с оптоизоляцией. Модули управляются ТТЛ сигналами и обеспечивают защиту от электрического пробоя при напряжениях до 4000 В. Электрические двигатели, контакторы, коммутаторы переменного тока, тиристорные регуляторы и другие устройства могут безопасно управляться с помощью этих модулей.

Компьютеры семейства MicroPC могут объединять все три класса модулей дискретного ввода/вывода. Большинство устройств требуют для управления ТТЛ уровень. Модули второго класса объединяются с помощью соединительных модулей высоковольтного интерфейса типа ITB-16/8, ITB-8/16 и подключаются к ТТЛ выходам компьютера. Модули семейства G4 (рис.3.12) обеспечивают интерфейс управления для третьего класса. Они объединяются с помощью пассивных соединительных плат MPB-08/16/24 (цифра определяет максимальное число модулей на плате).

Аналоговый ввод/вывод: аналоговый вход считывает значение напряжения с внешнего датчика, аналоговый выход генерирует напряжения для управления таким устройством, как регулятор скорости двигателя, измерительный прибор или привод головок. В то время, как дискретный ввод/вывод имеет только два состояния, аналоговый сигнал может принимать сотни и тысячи значений. Аналоговый интерфейс компьютера может быть описан с помощью следующих характеристик: разрешение; точность; коррекция 0; режимы ввода; диапазон аналогового сигнала; тип преобразования.

Для большинства промышленных применений достаточно точности 10-битного АЦП. Некоторые модули ввода аналоговых сигналов работают с однополярными сигналами (0-5 В), другие - с разнополярными (± 5 В). Внешние усилители с токовым выходом требуют резистора 250 Ом, включенного на аналоговом входе. Резистор позволяет преобразовать ток 4-20 мА в напряжение 1-5 В. Время преобразования аналогового сигнала в цифровую форму должно быть значительно меньше скорости изменения сигнала. Чем более быстротекущий процесс опрашивается с помощью АЦП, тем более высокоскоростной преобразователь необходимо использовать, чтобы получить измерение требуемой точности.

Таймерные функции. Многие процессы требуют измерения и генерации различных частотных сигналов, выдержек времени, счёта событий и т.д. Универсальным средством поддержания этой функции является массив индивидуально программируемых таймеров/счётчиков. В семействе MicroPC эту функцию выполняют модули 5300. Один таймер/счётчик может накапливать цифровые значения до величины 65535. Для увеличения разрядности таймеры/счётчики включаются каскадно, т.е. выход одного счётчика соединяется с входом другого.

Интерфейс последовательного обмена в настоящее время является неотъемлемой частью любого управляющего компьютера. Он используется для организации межкомпьютерной сети, управления принтерами, микроконтроллерами, устройствами считывания штриховых кодов и другими устройствами. Часто функции последовательного интерфейса реализуются программно с помощью линий дискретного ввода/вывода. Наиболее часто используют интерфейс RS-422 и RS-485, как стандарт межкомпьютерного обмена. Модемы и радиомодемы позволяют создавать сети неограниченной удалённости.

Интерфейс оператора - это обычно какая-либо клавиатура и дисплей, состоящий из нескольких строк знаковосинтезирующих индикаторов. Большинство клавиатур строится как матрица механических контактов, которые при нажатии замыкают соответствующие строки с соответствующими столбцами. Существует много типов дисплеев. Два основных класса: активные и пассивные. Принцип работы пассивного дисплея основан на отражении света внешнего источника. Так работают жидкокристаллические дисплеи (LCD- дисплеи). Они потребляют очень небольшой ток, примерно 10 мА на 4-строчный дисплей. LCD- дисплеи с подсветкой фона потребляют значительно больше. Катодолюминесцентные дисплеи имеют высокую яркость свечения, позволяют выводить символы разных цветов и не нуждаются во внешнем источнике света. Такой дисплей потребляет 500-600 мА от источника питания 5 В.

Память. В отличие от настольного персонального компьютера, промышленный контроллер использует память меньшего объёма. Это надо учитывать при расчёте необходимого объёма памяти для накопления данных или регистрации параметров управляемого процесса.

Другие системные функции. Существует еще много различных системных компонентов, необходимых для корректной работы всей системы управления. Батарея

питания энергонезависимого ОЗУ, часы/ календарь, кабели, драйверы устройств и интерфейсные платы являются обязательным компонентом современных контроллеров.

Интерфейс расширения системы. Все процессорные платы семейства MicroPC имеют возможность простой модернизации цепей ввода/вывода, изменения функционального состава (до 7 модулей расширения), оперативного перепрограммирования или наладки при изменении параметров процесса или состава технологического оборудования. Объединительные панели 5205/7-RMH позволяют соединить модули MicroPC и любые другие PC- модули половинного размера. Модули-конструкторы 5970/71 позволяют пользователю создать своё собственное устройство и включить его в систему MicroPC.

Характеристики некоторых функциональных модулей приведены в прил.2.

На рис 3.13 представлен вариант формирования ядра объектного контроллера из модулей MicroPC. Конструктивно контроллер размещается в монтажном каркасе 5204, имеющем 4 шинных позиции для установки модулей и место для размещения блока питания, например, типа 5105. Основой контроллера является модуль центрального процессора 4000 на основе процессора 386CX, обеспечивающий реализацию всех системных функций и непосредственное подключение клавиатуры и принтера. Для подключения стандартного монитора SVGA в состав контроллера включен модуль видеоадаптера SVGA 5420. Адаптер сети Ethernet 5500 обеспечивает обмен информацией с другими абонентами сети.

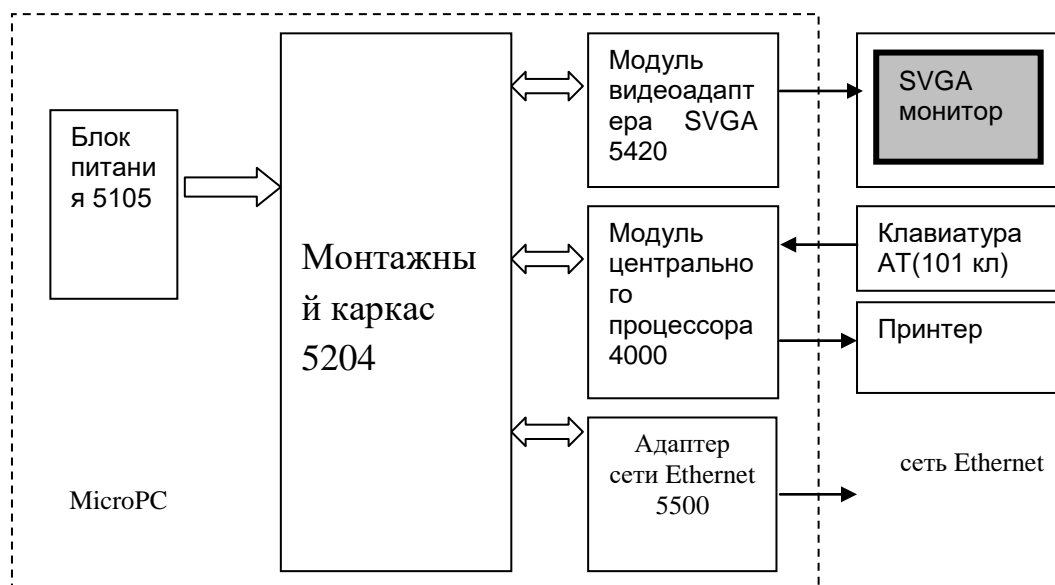


Рис.3.13. Структурная схема контроллера на базе модулей MicroPC

Лекция 13. Контроллеры с сетевой структурой

Объектная ориентация средств контроллера предполагает возможность пространственного разделения модулей входов-выходов для реализации управления технологическими объектами большой протяженности. Решения подобных задач осуществляются в виде внутренних сетей ОПК. Такое название сети получили от внутренней структуры ОПК, к элементам которой сети и относятся [2,27]. Кроме собственно модулей ОПК, оборудование, используемое на нижнем уровне автоматизации технологических процессов, такое как датчики, исполнительные механизмы, передаточные устройства, приводы, также требует организации связи с ОПК. Пространственное разнесение элементов внутренней структуры ОПК определяет требование к работе сети в режиме реального времени, а также повышенные требования к надежности работы. Принципиально возможны два подхода к построению внутренних сетей: индивидуальные закрытые архитектуры сети или открытые стандарты для организации сетевого интерфейса для элементов ОПК различных производителей. Первый подход, например, реализован в контроллерах серии

TSX-7 [2], где для поддержания обмена информацией между элементами ОПК в структуру введены интеллектуальные модули связи. Они включаются по концам канала передачи и обеспечивают ведение обмена, проверку линий связи, диагностику подключенных элементов ОПК. Хотя предлагаемые такими фирмами решения сравнительно недороги, второй подход к организации внутренних сетей ОПК оказывается более привлекательным. Использование стандартизованных интерфейсов для внутренней сети позволяет ориентироваться на перечень разнообразных модулей различных фирм при решении задач объектной адаптации. На рис. 3.14 представлен внешний вид соединения внутренней сети ОПК на базе модулей ADAM - 4000 в стандарте RS - 485 [27]. Поддержка стандартов последовательного интерфейса является неотъемлемым элементом однокристальных микроконтроллеров (прил. 1) и одноплатных контроллеров (разд. 3.2), используемых именно на нижнем уровне автоматизации. Необходимость в открытой, независимой от поставщика системе связи привела к разработке ряда стандартов [38,39,40]. Все чаще для связи внутри ОПК применяют бит-последовательные промышленные шины (bit serial Fieldbus).

Fieldbus - это полностью цифровая двунаправленная многоточечная коммуникационная система, используемая для связи приборов на объектах с системами оператора [21]. Одно из основных свойств системы Fieldbus состоит в том, что она поддерживает двунаправленную связь с множеством переменных величин. Физически в Fieldbus могут использоваться три вида топологий межсоединений: двухточечная, древовидная и многоточечная. Для соединения нескольких низкоскоростных систем Fieldbus с высокоскоростной могут служить устройства сопряжения (мосты).

Тип используемых средств зависит от приложения. Витая пара - наиболее дешевое средство, широко применяемое и удобное для установки. Оптоволоконные средства обеспечивают защиту от электромагнитных помех. Избыточность кабелей представляется необходимой, но способность продолжать работу в случае любых возможных отказов намного важнее. Весьма значима также гальваническая изоляция, поскольку плохая изоляция между сенсорами может стать причиной серьезных проблем. В большинстве случаев сеть Fieldbus должна подводить электропитание к связываемым ею приборам. Важным фактором является техника безопасности на определенных видах производств. Среднее число приборов, подсоединенных к сети Fieldbus, может колебаться от 50 до 200 при длине сети 100 ... 1000 м.

В соответствии с американским стандартом EIA RS-485 вариант с медным проводом является базовым для промышленных приложений. В нём используется двухпроводная витая пара проводов с экранированием и без. Возможны двухпроводные варианты с различными максимальными расстояниями.

Основные характеристики среды передачи RS-485:

Топология сети - линейная шина с оконечными нагрузками на обоих концах.

Передающая среда - экранированная или неэкранированная витая пара.

Количество станций - 32 станции в каждом сегменте без повторителей. С повторителями - до 127.

Максимальная длина шины без повторителя - 200 м при 1500 кбит/с, до 1,2 км при 93,75 кбит/с (экранированная пара).

Скорость передачи - выбирается из ряда 9,6; 19,2; 93,75; 187,5; 500; 1500 кбит/с.

Соединитель - 9-контактный разъём D-Sub.

В качестве примера рассмотрим реализацию сетевой структуры ПК с использованием модулей удаленного сбора данных и управления ADAM - 4000 и среды передачи RS-485 [27].

Модули серии ADAM - 4000 предназначены для организации взаимодействия между управляющим компьютером (контроллером) и датчиками непрерывных и дискретных параметров, а также для выдачи управляющих воздействий на исполнительные механизмы. При этом модули осуществляют обмен командами по каналу RS-485, ввод и нормализацию аналоговых сигналов (ток и напряжение), опрос состояния дискретных входов, фильтрацию

сигналов, вывод аналоговых и дискретных сигналов, аналого-цифровое и цифро-аналоговое преобразования, преобразование шкалы значений в предварительно заданные единицы измерений и формирование и передачу в адрес управляющего компьютера информации, после получения соответствующего запроса по каналу RS-485. Настойки и калибровки модулей осуществляются программным способом путем передачи соответствующих команд. Конструктивно модуль представляет собой функционально законченное устройство, заключенное в пластмассовый корпус (112*60*25мм) с клеммными винтовыми колодками по торцам корпуса (см. рис. 3.14). Номенклатура и краткие характеристики модулей приведены в прил. 3.

Модули серии ADAM-4000 функционально разделяются на три группы: модули аналогового и дискретного ввода – вывода, коммуникационные модули и модуль контроллера сети.

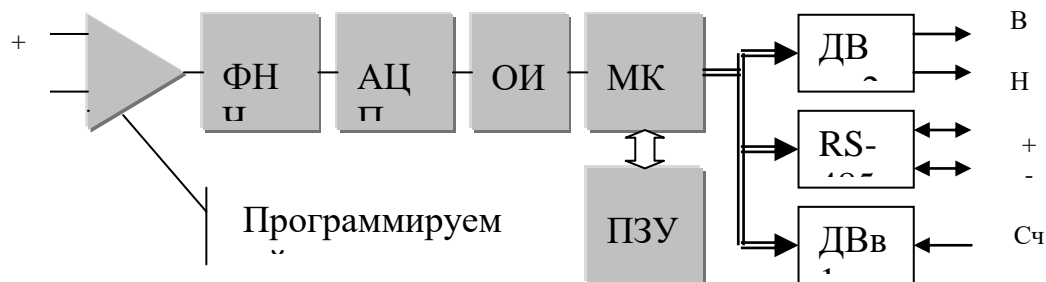


Рис.3.15. Структурная схема модуля аналогового ввода ADAM-4012

На рис.3.15 приведена структурная схема модуля аналогового ввода ADAM-4012 [27]. Входной сигнал, присутствующий на дифференциальном входе модуля, поступает на малошумящий усилитель с программируемым коэффициентом усиления. Последний может принимать значения в диапазоне от 1 до 128. Далее сигнал подвергается низкочастотной фильтрации (до 10 Гц) и поступает на вход сигма-дельта АЦП. Результат АЦ - преобразования через цепи оптоизоляции (ОИ) подается во встроенный микроконтроллер (МК), программное обеспечение которого позволяет осуществлять:

- сравнение значений входного сигнала с предварительно заданными верхним и нижним предельно допустимыми уровнями;
- управление соответствующими дискретными выходами (при превышении верхнего – В или нижнего – Н уровней);
- калибровку шкалы аналогового ввода;
- подсчет импульсов на дискретном входе счета внешних событий (ДВВ);
- анализ команд, поступающих по каналу связи (RS-485);
- передачу в адрес управляющего устройства цифрового отчета при поступлении соответствующего запроса;
- линейризацию сигнала от термопары (для модуля ADAM-4011).

Модули аналогового вывода позволяют

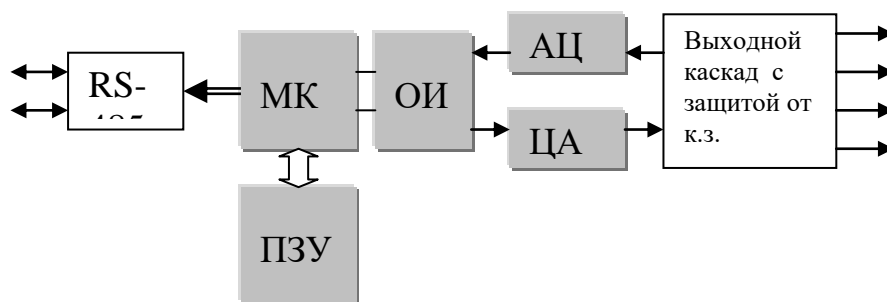


Рис.3.16. Структурная схема модуля аналогового вывода ADAM-4021

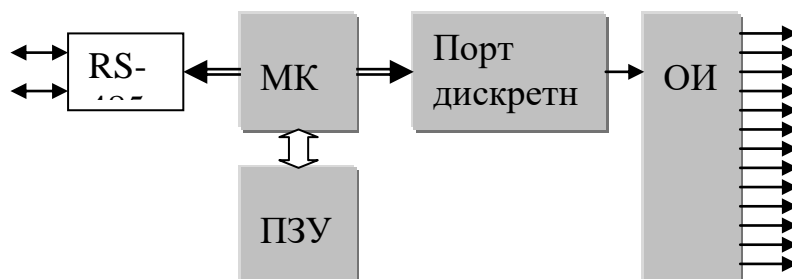


Рис.3.17. Структурная схема модуля дискретного вывода ADAM-4052

формировать сигнал в виде напряжения или тока с заданным уровнем и скоростью изменения. Модули дискретного ввода – вывода предназначены для осуществления контроля положения и управления коммутационными аппаратами, а также для организации взаимодействия с устройствами, уровни входных и выходных сигналов которых совместимы с ТТЛ. Принципы функционирования модулей аналогичны описанному выше. Структурные схемы модулей аналогового вывода и дискретного ввода – вывода приведены, соответственно, на рис. 3.16 и 3.17.

Для организации сетевой структуры из модулей ADAM в серию включен ряд коммутационных модулей. Модуль ADAM-4510 является двунаправленным повторителем общего назначения, который служит для увеличения протяженности линии связи в сети на основе интерфейса RS-485. Модуль ADAM-4520 реализует согласование интерфейсов RS-232/ RS-422/ RS-485 с автоматическим определением направления потока передаваемых данных и гальванической изоляцией. Модуль ADAM-4521 содержит микропроцессор, обеспечивающий управление двумя универсальными приемопередатчиками, что позволяет осуществлять двунаправленный обмен данными, в том числе при различных скоростях передачи. При автоматизации производств с территориально-распределенными технологическими участками в качестве физической среды передачи используется радио канал (РК) или канал оптоволоконной связи (КОВС). Для организации сетей с участками указанной физической среды в серию включены модули ADAM-4530 и 4550 (РК) и ADAM-4540 (КОВС). Например, модуль ADAM-4550 представляет собой радиомодем, в котором применена технология шумоподобного сигнала. Модуль имеет интерфейсы RS-232 и RS-485 и позволяет передавать сигнал в условиях прямой видимости с помощью малогабаритной антенны на расстояние до 150 м, а в случае применения направленных антенн - до 10 км.

И, наконец, модуль ADAM – 4500 может использоваться в качестве полноценного контроллера сети. Он имеет в своем составе микропроцессор 80C188, два последовательных порта (RS-232 и RS-422/ RS-485), флэш-память для хранения операционной системы (ROM-DOS) и исполняемой прикладной программы. Кроме того, модуль содержит часы реального времени и сторожевой таймер.

Рассмотренные выше модули серии ADAM могут быть объединены в сетевую структуру. Объединяющим началом такой сети является контроллер сети (КС). Взаимодействие между КС и модулями сети осуществляется путем последовательной передачи КС в адрес каждого модуля запроса, содержащего префикс типа команды, символьное представление сетевого адреса запрашиваемого модуля, число, соответствующее подтипу команды, и символ окончания (возврат каретки). При получении команды встроенное программное обеспечение модуля производит проверку ее корректности и идентификацию, после чего исполняет команду и посылает в адрес КС запрашиваемую информацию в виде строки символов.

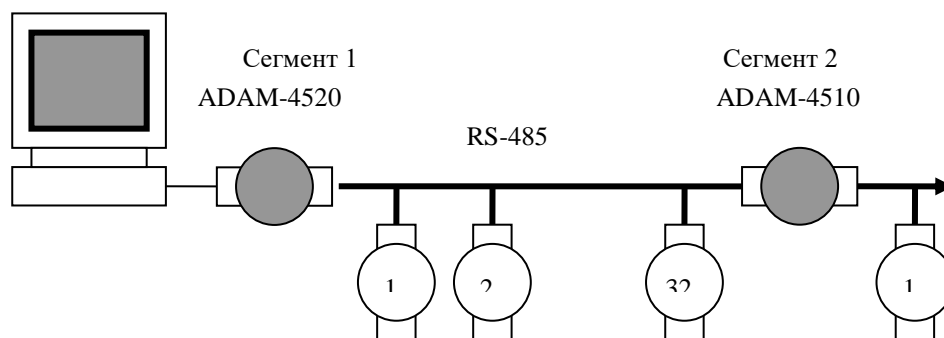


Рис.3.18. Внутренняя сеть типа линейная шина

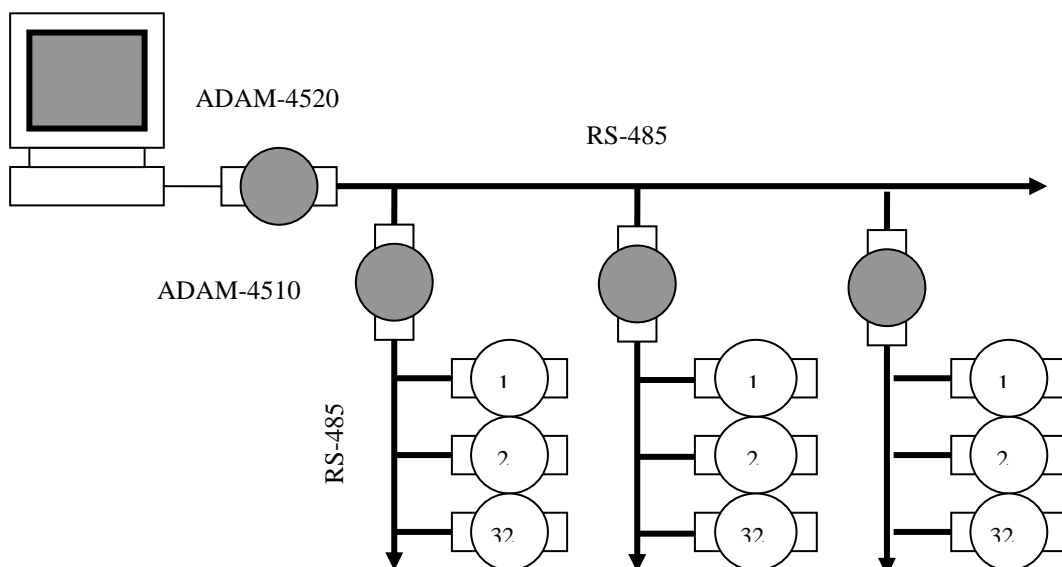


Рис.3.19. Внутренняя сеть типа звезда

Возможные варианты объединения модулей ADAM - 4000 в сеть показаны на рис. 3.18-3.21. На первом из них представлено соединение типа линейная шина с наращиванием сегментов и наличием в сегменте до 32 модулей ввода-вывода. В качестве контроллера сети использован промышленный компьютер.

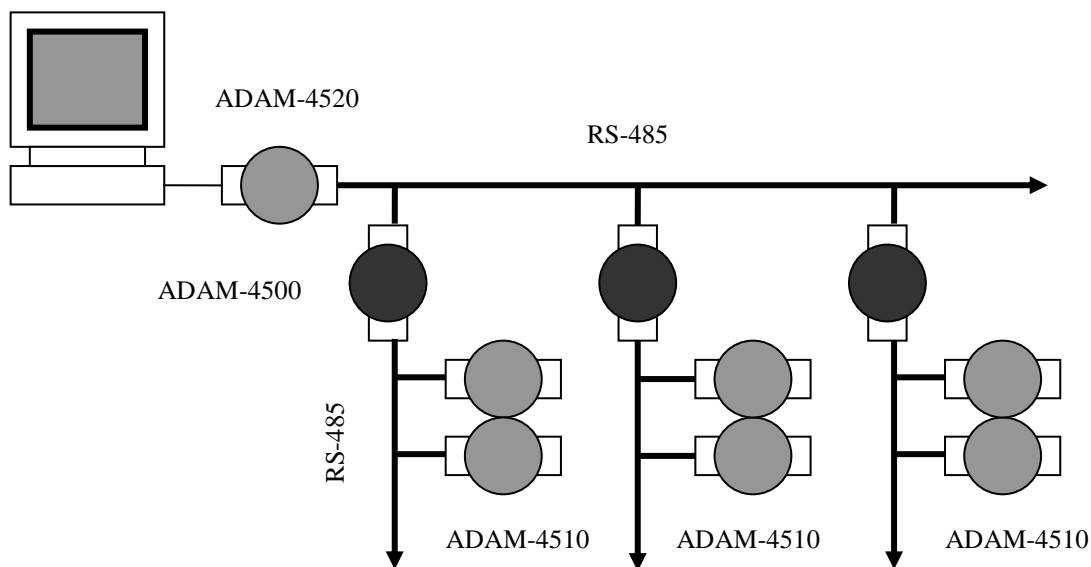


Рис.3.21. Внутренняя сеть на основе сетевых контроллеров

Следующий рисунок изображает соединение линейных сегментов в сеть типа звезда. Практически реализуется иерархическая структура на основе двух разделенных сред типа RS-485.

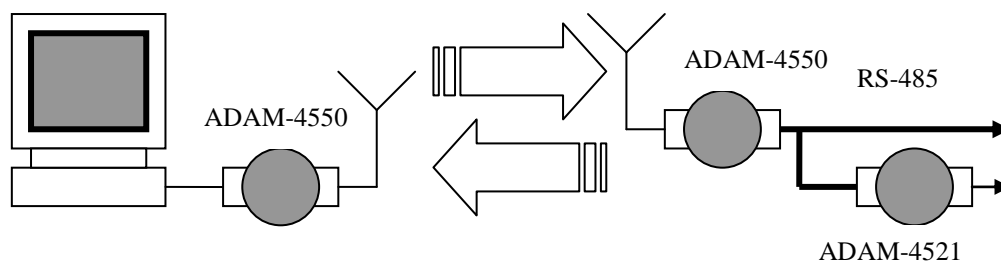


Рис.3.20. Внутренняя сеть на основе радиоканала

На каждом из уровней пространственное разнесение между модулями в целом до 1200 м. Допустимо использование комбинации из представленных типов соединений линейной шины и звезды.

На рис. 3.20 приведен пример использования модулей ADAM - 4550 для организации внутренней сети с радиоканалом. В заключение приведена схема внутренней сети на базе сетевых контроллеров ADAM-4500. Схема на верхнем уровне также как и на рис. 3.19 изображает соединение модулей ADAM-4500 в звезду, но не сегментов, а автономных сетей, управляемых соответствующим сетевым контроллером ADAM-4500 (1, 2,...).

Итак, предлагаемые варианты построения внутренних сетей на базе модулей серии ADAM предоставляют возможность пространственного разделения модулей входов-выходов для реализации управления технологическими объектами большой протяженности.

Лекция 14. Графический язык "Диаграмма функциональных блоков"

Как известно международный стандарт на языки программирования ПК - ИЕС 1131-3 (МЭК 1131-3) включает структурное программирование, абстрактные типы данных, выделение данных и процедур в блок (инкапсуляция) в сочетании с сохранением тесной связи с классическими языками для ПК. В стандарт включены текстовые языки: язык структурированного текста (из семейства языков Паскаль), позволяющий гибкие процедуры обработки данных; язык инструкций, относящийся к классу ассемблеров, позволяющий создавать эффективные оптимизированные процедуры. Но основу стандарта составляют графические языки: графический язык "Диаграмма цепей", распространенный в основном в США (язык релейных диаграмм для описания логических выражений любой сложности); графический язык "Диаграмма функциональных блоков", повсеместно используемый в Европе, позволяющий строить комплексную процедуру, состоящую из различных функциональных библиотечных блоков. Этот язык удобен для множества прикладных программ, управляющих передачей информации между различными компонентами системы. Функциональные блоки языка, таким образом, представляют специальные функции управления, используемые в управляющих системах.

В дополнение к этим графическим языкам стандарт ИЕС 1131-3 определяет элементы графического языка "Последовательные функциональные схемы", описывающий логику программы на уровне чередующихся процедурных шагов, переходов и блоков операций, которые могут быть использованы для организации "операций", написанных на любом языке, для получения алгоритмов последовательного управления. Основой элементов графического языка "Последовательные функциональные схемы" служит стандарт ИЕС 848, который является международной версией французского стандарта "Grafcet" (Graphical Function Charts).

Рассмотрим основные понятия и особенности программирования универсальных ОПК в системе визуального программирования UltraLogic [52], с использованием графического языка "Диаграмма функциональных блоков".

Назначение системы программирования UltraLogic

Система UltraLogic представляет комплекс программ, исполняемых в рамках единой оболочки под управлением операционной системы WINDOWS 95 (WINDOWS 3.11). Система реализована в соответствии с требованиями стандарта МЭК 1131-3 и предназначена для разработки прикладного программного обеспечения сбора данных и управления технологическими процессами, выполняемыми на программируемых контроллерах с открытой архитектурой.

В качестве языка программирования в системе реализован язык функциональных блокковых диаграмм, предоставляющий пользователю механизм объектного визуального программирования.

Архитектура системы

Архитектура системы UltraLogic представлена на рис. 4.12. UltraLogic состоит из двух частей:

- системы программирования;
- системы исполнения.

Система программирования содержит собственно средства подготовки программ и средства их отладки. Менеджер проекта объединяет средства подготовки программ и содержит: редактор переменных; конфигуратор контроллера; редакторы программ; компиляторы.

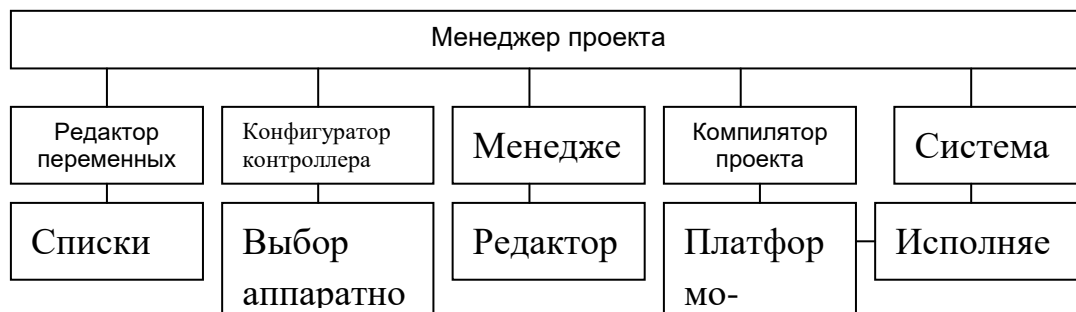


Рис. 4.12. Система программирования UltraLogic

Подсистема отладки содержит загрузчик программ, сетевой драйвер, средства осциллографирования, удаленной и пошаговой отладки.

Система исполнения функционирует в контроллере, который может базироваться как на платформе INTEL, так и другой аппаратной платформе. UltraLogic использует метод сетевого взаимодействия между контроллером (системой исполнения) или группой контроллеров, управляющими технологическим процессом, и системой визуализации данных. Сетевые ресурсы по запросу пользователя автоматически включаются в исполняемый код. Количество участников сети может достигать 256.

Технология программирования контроллеров в среде UltraLogic

Для создания новой программы необходимо создать новый проект. В новом проекте выполняются настройки конфигурации программируемого контроллера, переменных программы, сетевой поддержки и режимов отладки. Для создания проекта рекомендуется придерживаться следующего алгоритма работы:

1. Заполнение таблиц глобальных переменных.
2. Конфигурирование контроллера.
3. Привязка переменных к входам и выходам контроллера.
4. Распечатка листинга переменных.
5. Разработка программ.
6. Компиляция.
7. Загрузка исполняемого кода в контроллер.
8. Отладка программы.

Автоматизируемый технологический процесс разбивается на отдельные формальные задачи, выполняемые последовательно. Программа может состоять из множества программ, находящихся друг с другом в определенных отношениях, образующих иерархическое дерево. Все подпрограммы проекта, работа которых разрешена, выполняются один раз в цикле работы контроллера. Цикл контроллера имеет период 10 миллисекунд.

Собственно программирование осуществляется с помощью специального графического редактора. Разработка собственных функциональных блоков также происходит с помощью графического редактора. Программист с помощью мыши устанавливает функциональные блоки на листе программы, соединяет их связями и присваивает связям имена переменных. Комментарии могут помещаться в любом месте листа программы.

Далее в секции «Переменные» проекта описываются глобальные переменные, сформированные в списки по тем или иным признакам. Разбиение переменных на списки носит условный характер и служит для удобства поиска, редактирования и просмотра. Переменные сформированы в 5 списков: «Константы», «Глобальные», «Входные»,

«Выходные», «Сетевые». Переменные, введенные на этапе конфигурирования, назначенные на входы и выходы контроллера, попадают в эти списки автоматически. Переменные, имена которых введены как имена связей функциональных блоков в программе, являются локальными и в эти списки не попадают.

В секции «Конфигурация» выполняются следующие действия:

- выбирается аппаратная платформа контроллера. UltraLogic поддерживает контроллеры фирм Octagon Systems и Advantech,
- выбирается тип процессорного модуля. Например, процессорная плата MicroPC 6012 фирмы Octagon Systems или контроллер ADAM-5510 фирмы,
- указывается наличие и тип сети, сторожевого таймера и выбирается тип компилятора,
- выбираются типы используемых модулей ввода-вывода контроллера,
- осуществляется привязка переменных к входам и выходам соответствующих модулей.

Все действия происходят в режиме интерактивного диалога путем выбора соответствующих опций из последовательно открываемых окон.

Готовый проект компилируется, после чего полученный код системы исполнения загружается в контроллер. В код системы исполнения встраивается драйвер сетевого обмена, который обеспечивает мониторинг и удаленную отладку. Любая переменная программы, помещенная в список «Сетевые», доступна для чтения и записи другими участниками сети.

Средства отладки программ

Система UltraLogic может работать с реальным контроллером или эмулировать его работу. Драйвер эмулятора позволяет эмулировать работу контроллера, а если в качестве сетевого драйвера выбрать эмулятор, то отладка программы выполняется на компьютере системы программирования без контроллера и других участников сети.

Меню «Отладчик» системы UltraLogic имеет подменю «Переменные», «Осциллограф» и «Терминал». Оно предназначено для загрузки программ и их отладки. Меню «Переменные» служит для просмотра и изменения значений переменных в программе. Меню «Осциллограф» — для включения режима осциллографирования переменных. Меню «Терминал» — для загрузки программ в контроллер и запуска их на исполнение.

В режиме «Переменные» сканируется сеть и выводится список участников. Выбрав участника сети, вы получаете доступ к его сетевым переменным. Список переменных является динамическим, т.е. отражает реальное состояние переменных с частотой сетевого обмена. Изменяя значения переменных удаленного или эмулируемого контроллера, можно поменять, например, уставки регуляторов, изменить масштабные коэффициенты и т.д.

В режиме «Осциллограф» значения переменных представляются в виде графиков. Этот режим удобно использовать для анализа графиков переходных процессов и подбора параметров функций регулирования. Выбор контроллера для осциллографирования производится так же, как и в режиме «Переменные». Во время осциллографирования можно изменять масштаб изображения по осям X и Y, устанавливать границы и размер изображения осциллограмм. После останова осциллограммы могут быть сохранены в виде файла.

В режиме «Эмулятор» в качестве целевого компилятора выбирается эмулятор. После трансляции программы открывается окно эмулятора. В окне отображается иерархическое дерево исполняемой программы и точки останова, если они заданы. В эмуляторе программа может исполняться непрерывно, с остановкой в точках останова и по шагам. В опциях эмулятора можно включить дополнительное окно сообщений о выполняемых блоках программы и возвращаемых ими значениях переменных.

Язык функциональных блокковых диаграмм (FBD)

Каждый функциональный блок представляет собой прямоугольник, внутри которого имеется обозначение функции, выполняемой блоком. Выход функционального блока может быть соединен связью с другими блоками. Каждая связь является переменной. Один или

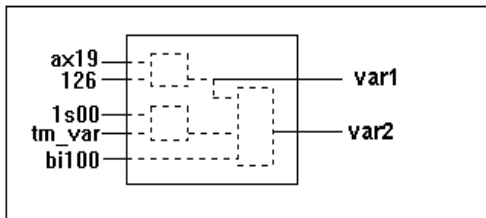
несколько функциональных блоков, соединенных связями между собой, образуют программу на языке FBD. Язык FBD описывает закон преобразования переменных программы. Имеются следующие формальные правила языка FBD:

- функциональные блоки могут располагаться произвольно в поле программы;
- не может быть свободных (несоединенных) входов и выходов функционального блока;
- любая связь может иметь имя и тип;
- входы и выходы функциональных блоков, присоединенные к связям, имеющим одинаковые имена, считаются соединенными;
- очередность выполнения блоков в программе: слева - направо, сверху - вниз.

В том случае, когда одну и ту же переменную формируют несколько функциональных блоков, последнее значение переменной будет присвоено тем блоком, который выполняется последним.

Программа на языке FBD может содержать следующие базовые типы объектов: функциональные блоки и связи; переходы и метки; комментарии.

Функциональные блоки и связи



Переменные и константы FBD программ назначаются на связи, которые присоединяются к входам и выходам функциональных блоков. На входе FBD блока может быть: константное выражение; внутренняя или входная переменная; выходная переменная. На выходе FBD блока может быть любая внутренняя или выходная переменная.

Переходы и метки

Метки и переходы можно использовать для управления процессом выполнения программы. Для этого выход блока соединяют связью с оператором GOTO {имя метки}. Если значение переменной, которая формируется на выходе функционального блока, принимает значение TRUE, то программа продолжается с блока, помеченного меткой. Если оператор GOTO не соединен с переменной типа BOOLEAN, осуществляется безусловный переход к метке.

Для завершения программы существует оператор RETURN. Если на входе оператора RETURN значение переменной принимает значение TRUE, то осуществляется условный переход к следующей программе. Если оператор RETURN не соединен с переменной типа BOOLEAN, осуществляется безусловный переход к следующей программе. В случае, когда оператор RETURN встретился внутри библиотечного блока, управление передается следующему блоку, но не следующей программе.

Базовые функции языка FBD

Базовые функции языка FBD разделены на следующие группы:

- логические функции;
- функции сравнения;
- арифметические функции;
- функции управления программой и таймерами.

Логические функции

Логические функции в качестве операндов используют переменные типа Boolean. Любой вход и выход логических функций может быть проинвертирован. После инициализации программы, переменные на входах и выходах всех функциональных блоков логических функций принимают значение FALSE, кроме тех глобальных переменных, значение которых при инициализации явно объявлены как TRUE. В перечень логических функций входят: инвертирование, логическое "И", логическое "ИЛИ", логическое исключаящее "ИЛИ", детектор фронта. Подробное описание логических функций приведено в прил.8.

Функции сравнения

Все функции сравнения в качестве операндов могут использовать переменные типа INTEGER, FLOAT, TIMER. Входные переменные каждой отдельной функции должны иметь одинаковый тип. На выходах всех функциональных блоков сравнения по результату операции формируются переменные типа BOOLEAN. После инициализации программы эти переменные принимают значение FALSE, кроме тех глобальных переменных, значение которых при инициализации явно объявлены как TRUE. В перечень функций сравнения включены: меньше чем, меньше или равно, больше чем, больше или равно, равно, не равно. Подробное описание функций сравнения приведено в прил.8.

Арифметические функции

В перечень арифметических функций входят: сложение, вычитание, умножение, деление, присваивание.

Все арифметические функции в качестве операндов могут использовать переменные типа INTEGER, FLOAT, TIMER. Входные переменные каждой отдельной функции должны иметь одинаковый тип. После инициализации программы, переменные на выходах всех арифметических функциональных блоков принимают значение 0, кроме тех глобальных переменных, значение которых при инициализации явно объявлены иначе.

Подробное описание арифметических функций приведено в прил.8. На рис.4.13 представлены программы, построенные на основе арифметических функций, соответственно, на рис.4.13 слева - программа вычисления среднего арифметического из 3-х чисел, а на рис.4.13 справа - программа, реализующая алгоритм пропорционального регулятора. В последнем разность (величина рассогласования) между истинным значением регулируемого параметра Val и его заданным значением Ref усиливается в Km раз и используется в качестве регулирующего воздействия.

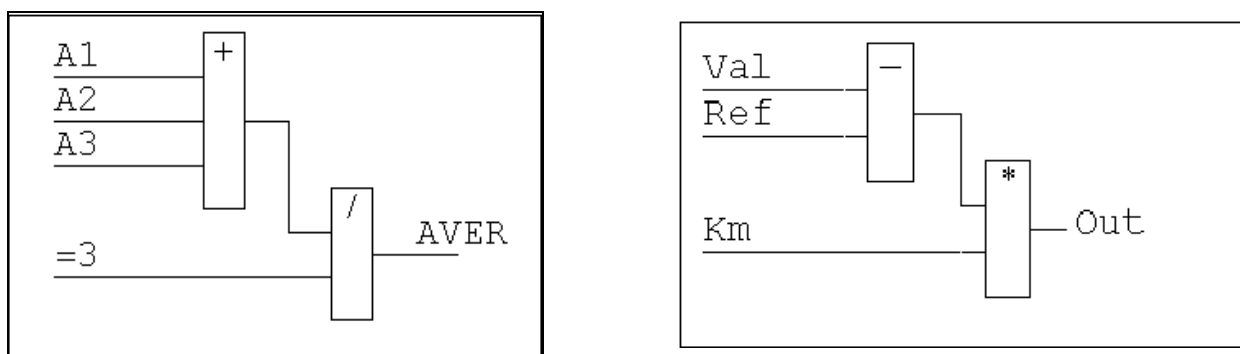


Рис.4.13. Примеры построения программ с использованием арифметических функций

Функции управления программой и таймерами

Функции управления программой в качестве входных операндов могут использовать переменные типа BOOLEAN.

Переход к метке GOTO



Где **Prog1** – метка. Оператор GOTO может быть соединен связью с переменной типа BOOLEAN. Если данная переменная примет логическое значение TRUE, будет осуществлен переход к указанной метке. В других случаях оператор GOTO будет проигнорирован. Если оператор GOTO не соединен с переменной типа BOOLEAN, осуществляется безусловный переход к метке.

Выход из программы RETURN



Завершает текущую программу и передает управление следующей за ней программой. Для условного завершения программы оператор RETURN может быть соединен связью с переменной типа BOOLEAN. Если данная переменная примет логическое значение TRUE, программа завершится. В случае, если оператор RETURN встретился внутри библиотечного блока, то управление перейдет к следующему блоку, но не к следующей программе.

Пуск таймера TSTART

→ **Tstart tm_var**

Где **tm_var** - переменная таймерного типа. Осуществляется

инициализация (сброс) и начинается инкрементирование указанной таймерной переменной с периодом 0.01 сек. до момента выполнения команды **TSTOP**.

Останов таймера TSTOP

→ **TSTOP tm_var**

Останавливает инкрементирование указанной таймерной переменной, состояние переменной сохраняется.

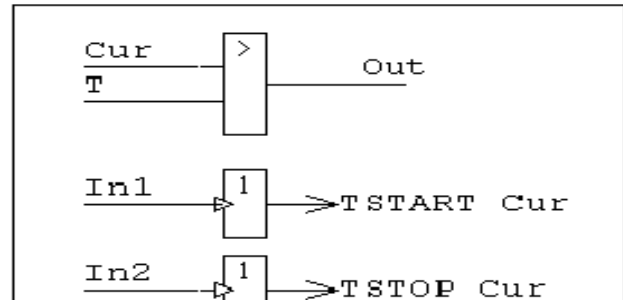
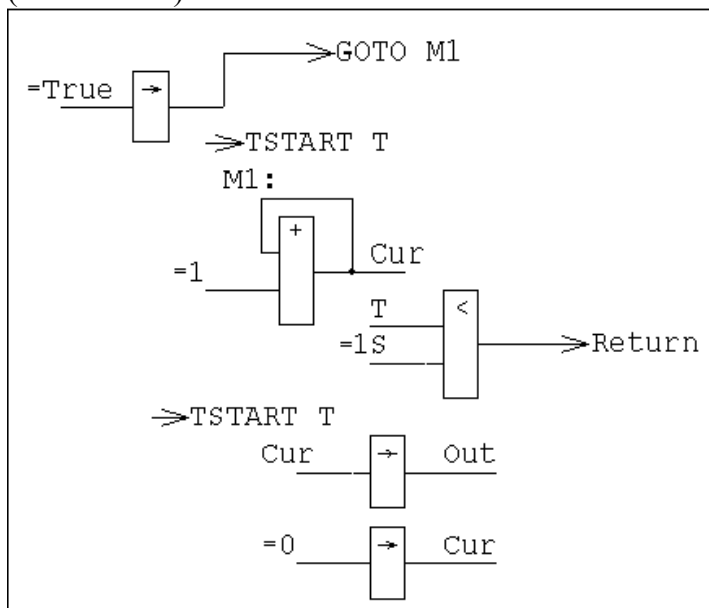


Рис. 4.14. Программа временного компаратора

На рис. 4.14 в качестве иллюстрации использования перечисленных функций представлена программа обнаружения заданного интервала времени. Переменная **Out** принимает значение **TRUE** в том случае, когда интервал времени, замеренный таймером **Cur**, будет больше интервала, заданного переменной (константой) **T**.



На рис. 4.15 представлен пример программы расчет частоты цикла контроллера.

Оператор **GOTO** расположен выше оператора присваивания поэтому выполняется первым. Во время первого цикла на входе оператора **GOTO** установлено состояние **FALSE**, и переход к метке **M1** игнорируется. Управление передается оператору присваивания, который устанавливает на выходе логическое значение **TRUE**. Все последующие входы в программу будут начинаться с метки **M1**. Инициализируется таймерная переменная **T**. Переменная **Cur** увеличивает свое значение на **1** в

каждом цикле контроллера. По истечении 1 сек. таймерная переменная **T** инициализируется вновь, значение переменной **Cur** переписывается в переменную **Out**, переменной **Cur** присваивается значение 0 и счет начинается заново.

Старт программы GSTART

→ **GSTART name_prog**

Где **name_prog** - имя запускаемой (неактивной) программы проекта. Запускает любую неактивную программу сначала, включенную в состав проекта, программе присваивается статус активной.

Стоп программы GSTOP

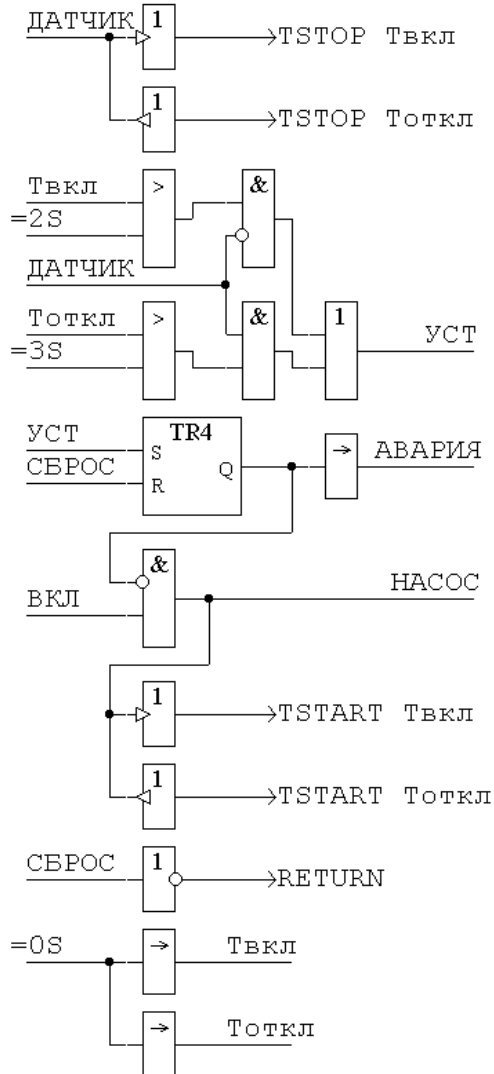
→ **GSTOP name_prog**

Где **name_prog** имя любой запущенной (активной) программы проекта. Останавливает любую активную программу, присваивает программе статус остановленной, все локальные переменные программы теряются.

При написании программы базовые функциональные блоки могут дополняться библиотечными. С помощью редактора FBD программ, используя базовые функциональные блоки, а также другие библиотечные блоки, можно составить программу, реализующую функции создаваемого блока, и графически оформить новый блок, как функционально законченный элемент. При использовании такого блока в программе, он отображается как

единый объект, работая, по сути, как FBD программа. Вы можете создавать собственные функциональные блоки, хранить и накапливать их в библиотеках. Библиотеки лучше всего формировать по тематическим разделам. Например, библиотека регуляторов, библиотека фильтров сигналов, библиотека математических функций и т.п.

Рассмотрим примеры программ управления. На рис.4.16 приведена программа управления насосом. Таймерные переменные (Твкл, Тоткл) используются для определения



контрольных промежутков времени между сигналом на включение насоса (НАСОС) и приходом сигнала подтверждения включения от датчика обратной связи (ДАТЧИК). Этот блок включен в библиотеку UltraLogic "LIB1-Разное" под названием PUMP.

После подачи команды на включение насоса (ВКЛ= TRUE) и если нет аварии (АВАРИЯ=FALSE), устанавливается сигнал включения насоса (НАСОС=TRUE). Одновременно запускается таймерная переменная Твкл, которая измеряет интервал времени между включением насоса и приходом сигнала от датчика обратной связи. За 2S от датчика должен прийти ответ (ДАТЧИК= TRUE), что насос включился и работает нормально. Если за 2S ответ от датчика не пришел, то устанавливается признак аварии (АВАРИЯ=TRUE), и сигнал на управление насосом отключается (НАСОС=FALSE). Сигнал АВАРИЯ является триггерным и сбрасывается сигналом СБРОС.

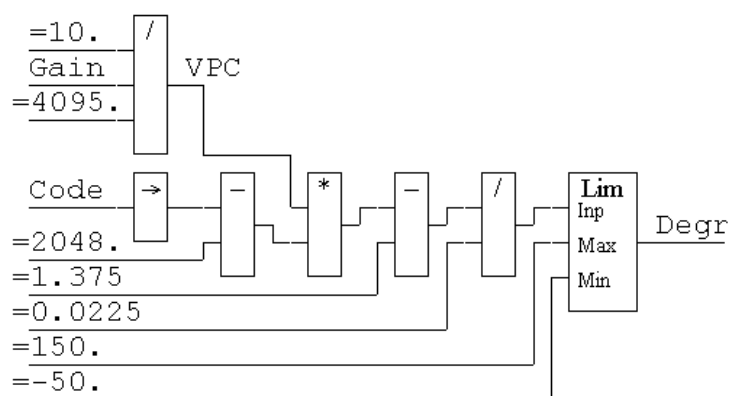
Рис.4.16.Программа управления насосом

Аналогично схема работает и при отключении насоса. После подачи команды на отключение насоса (ВКЛ= FALSE), выключается сигнал управления насосом (НАСОС=FALSE) и одновременно запускается таймерная переменная Тоткл, которая измеряет интервал

времени между выключением насоса и приходом сигнала от датчика обратной связи.

За 3S от датчика должен прийти ответ (ДАТЧИК= FALSE), что насос выключился. Если за 3S ответ от датчика не пришел, то устанавливается признак аварии (АВАРИЯ= TRUE).

На рис.4.17 приведен пример программы преобразования значения сигнала, который приходит в кодах АЦП, в физическую величину, измеряемую датчиком. В примере приведены данные модуля 5710 (фирма Advantech) Analog I/O Card и датчика температуры AD22100. Датчик выдает напряжение пропорционально температуре. При изменении температуры на +1°C напряжение изменяется на +22.5 мВ. Температуре 0°C соответствует напряжение 1.375В. Диапазон датчика от -50°C до +150°C. Модуль 5710 имеет три коэффициента усиления (x1, x10, x100), диапазон входного напряжения 10В и 12-ти разрядный АЦП. Рис.4.17.Программа вычисления температуры



Формула преобразования:

$$Degr = ((Code - 2048) * (MaxV / Gain / MaxC) - 1.375) / dC,$$
 где

Degr — значение измеренной температуры в °C.

Code — код сигнала измеренного АЦП.

2048 — поправка для смещения диапазона АЦП в пределы от -2048... +2047.

MaxV — диапазон входного сигнала АЦП (10 В).

Gain — коэффициент усиления модуля 5710 (1, 10 или 100).

MaxC — максимальный код АЦП (4095).

1.375 — напряжение на датчике при 0°C.

dC — приращение напряжение на датчике на 1°C (0.0225 В/°C).

После расчета температуры по приведенной формуле, значение переменной Degr ограничивается диапазоном от -50 до +150, т.е. областью достоверных показаний датчика температуры AD22100.

Лекция 15. Включение ОПК в локальные сети предприятия

Используется два подхода к организации включения ОПК в локальные сети предприятия: подключение к станции (серверу) сети или непосредственное включение в сеть в качестве станции.

Для подключения к станции (серверу) сети применяют бит - последовательные промышленные шины, в том числе, стандартов RS-232, RS-422 и RS-485 (см. разд. 2.3). В этом случае ОПК располагается от станции на расстоянии, допустимом для указанных интерфейсов, соответственно, 15м, 1200м, 1200м.

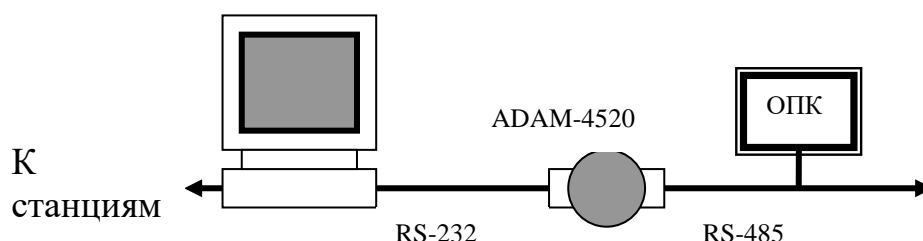


Рис.3.22. Подключение ОПК к станции (серверу) с помощью бит - последовательных промышленных шин, типа RS-232 и RS-485

На рис.3.22 приведен пример подключения ОПК к станции (серверу) с помощью бит - последовательных промышленных шин, типа RS-232 и RS-485. ОПК и станция дополняются модулями (узлами) указанных интерфейсов с прокладкой шины (кабеля) связи. В частности в примере использован для согласования двух интерфейсов RS-232 и RS-485 модуль ADAM-4520.

При использовании для организации связи между контроллерами нестандартных средств, необходимо предусматривать модули согласования интерфейсов. Например, как это сделано в контроллерах серии Ремиконт Р-130. Эти контроллеры объединяются в локальную сеть кольцевой конфигурации "Транзит", поддерживающую интерфейс ИРПС. В сети "Транзит" информация последовательно передается от одного контроллера к другому. Для выхода из кольца или просто для подключения одиночного контроллера Ремиконт Р-130 к станции предусмотрен специализированный контроллер Ремиконт – Шлюз. Он имеет два канала последовательной связи. Назначение этих каналов специализировано. Каналы имеют

следующие наименования: сетевой канал и абонентский канал. С помощью сетевого канала шлюз включается в сеть "Транзит" по тем же правилам, по которым в эту сеть включается Ремиконт Р-130. Через абонентский канал шлюз соединяется со средствами верхнего уровня управления (компьютерами или контроллерами другого типа) либо с абонентским каналом шлюза, входящего в другую сеть "Транзит" (рис. 3.23). Один шлюз, входящий в сеть "Транзит", связывается с одним абонентом. Если с сетью "Транзит" должны взаимодействовать несколько абонентов, то в сеть должны включаться несколько шлюзов.

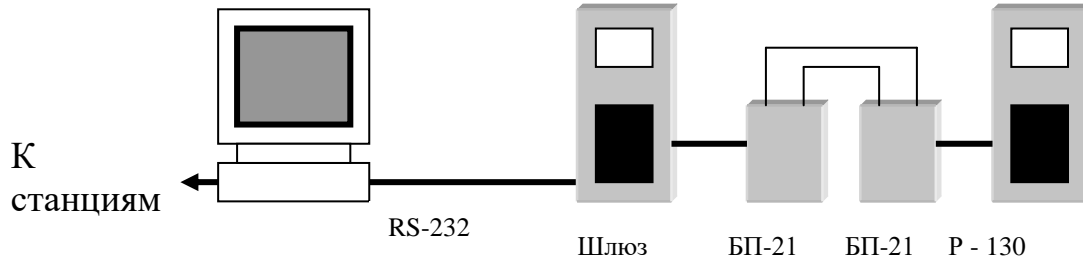


Рис.3.23. Подключение контроллера Р -130 к станции через Шлюз

Однако при большей удаленности ОПК и локальной сети (или при невозможности прокладки кабелей связи) используются радиоканалы или телефонные линии. Пример использования радиоканала для подключения ОПК к станции приведен на рис.3.20 в разделе 3.5. При использовании в качестве передающей среды телефонных линий в составе ОПК и компьютера необходимо устройство, которое может принять сигнал из телефонной сети и преобразовать его в цифровую информацию. На выходе этого устройства информация подвергается МОдуляции, а на входе ДЕМОдуляции, отсюда и название МОДЕМ. Назначение модема заключается в замене сигнала, поступающего из компьютера (сочетание нулей и единиц), электрическим сигналом с частотой, соответствующей рабочему диапазону телефонной линии. Акустический канал этой линии модем разделяет на полосы низкой и высокой частоты. Полоса низкой частоты применяется для передачи данных, а полоса высокой частоты - для приема. Используется много способов кодировки информации, наиболее известными из которых являются метод FSK (Frequency Shift Keying) для скорости передачи до 300 бод (бод - единица скорости передачи информации, равная 1 бит/с) и метод PSK (Phase Shift Keying) для более быстрых модемов, со скоростью передачи до 2400 бод. FSK использует четыре выделенные частоты. При передаче информации сигнал частотой 1070 Гц интерпретируется как логический нуль, а сигнал частотой 1270 Гц - как логическая единица. При приеме нуль соответствует сигналу 2025 Гц, а единица - 2225 Гц. PSK использует две частоты: для передачи данных - 2400 Гц, для приема - 1200 Гц. Данные передаются по два бита, при этом кодировка осуществляется посредством сдвига фазы сигнала. Для кодировки используются следующие сдвиги фазы: 0 градусов для сочетания битов 00, 90 градусов - для 01, 180 градусов - для 10, 270 градусов - для 11. Существуют также и другие виды модуляции (DPSK, QAM, TCM).

Модем выполняется либо в виде внешнего устройства, которое одним выходом подсоединяется к телефонной линии, а другим - к стандартному порту компьютера (разъем RS-232), либо в виде печатной платы, которая устанавливается на системную шину компьютера (внутренние варианты модемов могут быть установлены как в ISA, так и в PCI-шину). На рис.3.24 представлен вариант подключения ОПК к станции с помощью телефонной линии и внешних модемов.

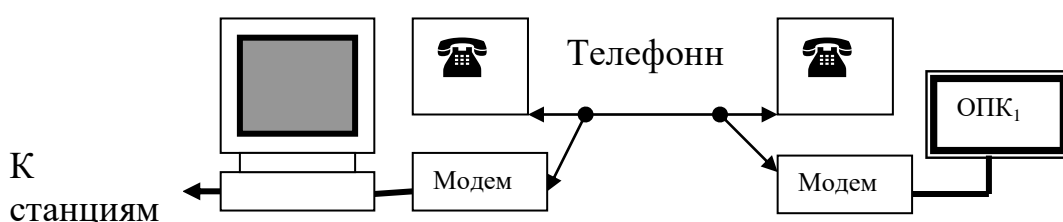


Рис.3.24. Подключение ОПК к станции с помощью телефонной сети

Контроллер модема - это, как правило, специализированный микроконтроллер типа SC1107 или SC1108, содержащий 8-разрядное АЛУ, ПЗУ в 8 Кбайт, ОЗУ 128 байт, таймер, командный регистр, контроллер прерываний, стек, порт ввода/вывода. Если плата модема присоединена к системной шине ПК, то применяется "параллельный" контроллер SC1107. Если же плата работает с компьютером посредством RS232, то используется "последовательный" контроллер SC1108. В конструкциях ОПК роль контроллера выполняет микроконтроллер семейства 8051 с внешним ПЗУ (i2732, 2764) и микросхемой 74LS373.

Непосредственное включение ОПК в сеть в качестве станции осуществляется с помощью 8-, 16- и 32-битных сетевых адаптеров (карт), например, Ethernet [41,42]. Включение адаптеров (карт) производится в системную шину ОПК или компьютера, а выхода карты - в линию связи локальной сети. Линия связи выполняется на витой паре или коаксиальном кабеле. Сетевой адаптер - это плата, которая вставляется в системную шину ОПК. Она имеет два разъема для подключения к сетевому кабелю. Для Ethernet в стандарте ISA используется три вида сетевых адаптеров: 8-, 16- и 32-битовые. Адаптер 8-битовый может вставляться в 8-битовый или 16-битовый слоты материнской платы и используется, главным образом, в компьютерах или ОПК, где нет 16-битовых слотов. Иногда 8-битовые адаптеры используются для ОПК, если требования к скорости передачи данных не высоки. Для 16-битового адаптера необходимо использовать 16-битовый слот. На компьютерах 80386 или 80486 имеет смысл использовать скоростные 32-битовые адаптеры, по крайней мере, для тех станций, на которые приходится максимальная нагрузка. Для ускорения работы на плате сетевого адаптера может находиться буфер. Размер этого буфера различен для адаптеров разных типов и может составлять от 8 кбайт для 8-битовых адаптеров до 16 кбайт и более для 16- и 32-битовых адаптеров. Сетевые адаптеры Ethernet используют порты ввода/вывода и один канал прерывания. Некоторые адаптеры могут работать с каналами прямого доступа к памяти (DMA). На плате адаптера может располагаться микросхема постоянного запоминающего устройства (ПЗУ) для создания так называемых бездисковых рабочих станций. Это, в частности, контроллеры, в которых нет ни винчестера, ни флоппи-дисков. Загрузка операционной системы выполняется из сети, и выполняют ее программа, записанная в микросхеме дистанционной загрузки.

Лекция 16. Технологии визуального программирования АСУ ТП

Как уже отмечалось выше, использование для программирования ОПК SCADA - систем, то есть технологии визуального программирования связано с применением в ПК процессоров высокой производительности (i80386, i80486, и т.п.) и перенесением на нижний уровень автоматизации технологий верхнего уровня автоматизации. Поэтому первоначально предназначенные только для построения человеко-машинного интерфейса (ММИ) верхнего уровня АСУ ТП в настоящее время SCADA - системы интенсивно внедряются в сфере ОПК [53]. Разработка прикладной программы автоматизации в SCADA осуществляется в среде визуального (графического, мультимедийного и видео) представления объекта и средств автоматизации. Понятие язык программирования полностью трансформируется в понятие язык проектирования, а точнее язык взаимодействия оператора с ЭВМ в привычной для него предметной среде [54]. Предметная среда обеспечивается наличием соответствующих библиотек графических примитивов, экранных форм представления информации, формул и алгоритмов математической обработки и управления, драйверов и т.п. На Российском рынке SCADA - систем АСУ ТП следует отметить, прежде всего, TRACE MODE (TM), разрабатываемую и поставляемую фирмой AdAstra Research Group [55,56,57]; Genesis, распространяемую в России фирмой ПРОСОФТ [58,59], Wonderware InToch, распространяемую в России фирмой PLC Systems [60]. Каждая из перечисленных систем работает в среде MS-DOS - Windows и располагает средствами программирования ОПК, построенных на основе архитектуры IBM PC - совместимых промышленных компьютеров.

Рассмотрим SCADA - систему TRACE MODE, версия 4.20 которой ориентирована на проектирование программного обеспечения контроллеров, использующего в качестве технической основы программируемые PC - совместимые контроллеры.

TRACE MODE позволяет структурировать прикладной проект путём выделения объектов, имеющих ясный технологический смысл: цех, агрегат, параметр. В связи с этим, разработка АСУ производится в среде визуального проектирования посредством создания, редактирования и тиражирования объектов. Проектирование осуществляется в графических редакторах без программирования. Созданные в системе ТМ проекты представляют набор файлов, описывающих используемые сигналы и промежуточные переменные, структуру математической обработки данных, документирования и архивирования, а также файлы содержащие графические формы представления информации и управления, шаблоны генерируемых отчетов, проигрываемые технологические и аварийные сообщения и пр. Созданные проекты запускаются под управлением специальных программ мониторов, интерпретирующих рабочие файлы проекта. При использовании в системах управления контроллеров позволяющих запускать программы под управлением MS DOS разработку всего проекта (верхнего и нижнего уровней) можно осуществлять в одной среде разработки. Дело в том, что программный комплекс ТМ включает в себя специальный монитор, ориентированный на выполнение задач непосредственного цифрового управления. При его использовании разработка всего проекта ведется в единой среде, что позволяет значительно упростить согласование данных верхнего и нижнего уровней, упростить обмен данными и сократить количество трудозатрат на разработку. Наличие исполнительных (run-time) модулей - "Мониторов реального времени" (MPB) для разных операционных систем, позволяет создавать рабочие места оператора, работающие под управлением MS-DOS, Windows 3.1 и Windows for Workgroups 3.11, причем ТМ является как бы мостом между MS-DOS и Windows. Все проекты, разработанные в ТМ, могут быть запущены как под DOS, так и под Windows на любом этапе жизненного цикла управляющей программы. TRACE MODE предоставляет возможность разрабатывать неограниченное число прикладных систем автоматизации исключительно в графических редакторах.

TRACE MODE является сетевой системой. В рамках пакета можно создавать сетевые комплексы, включающие до 200 сетевых узлов. При этом в локальную сеть могут объединяться как операторские станции, так и контроллеры, на которых запускаются мониторы ТМ. Кроме обмена данными по локальной сети, в ТМ реализована поддержка обмена операторскими станциями между собой и контроллерами (работающими под управлением ТМ) по последовательным интерфейсам.

Создание АСУТП в TRACE MODE состоит из нескольких этапов:

- Разработка математической основы АСУТП и связи с УСО;
- Создание статичного рисунка (мнемосхем);
- Динамизация проекта;
- Сборка системы, установка параметров реального времени;
- Запуск системы в реальном времени.

Разработка математической основы АСУТП в объектной среде. Объекты ТМ состоят из элементарных объектов - каналов. Каждый канал содержит настройки на источники и приемники данных, функции первичной и математической обработки, функции управления и многие другие. TRACE MODE позволяет выполнить следующие виды первичной обработки:

- аппаратурный контроль (зона нечувствительности);
- фильтрацию;
- трансляцию принятых значений по законам, осуществляющим: масштабирование; компенсацию погрешности; компенсацию смещения; компенсацию чувствительности датчика; линейную и ступенчатую интерполяцию; табличное преобразование; полиномиальное преобразование; логические операции и др. параметры.

Функции автоматического управления позволяют формировать значение управляющего воздействия в зависимости от значений атрибутов любых каналов из базы каналов ТМ. В ТМ реализованы следующие функции автоматического управления:

- PID, PD, PI, P, PDD;
- позиционное регулирование (по зоне нечувствительности, по временному интервалу);
- цифро-импульсное управление (шиотно-импульсная модуляция аналогового сигнала);
- логическое управление;
- групповое управление;
- ручное управление
- управление по произвольной формуле и др.

Математическая обработка позволяет осуществить расчеты значения параметров по косвенным измерениям и включает следующие виды обработки:

- алгебраическую (сумма, разность, умножение, деление, логарифм, тригонометрические функции, квадратные корни, полином);
- статистическую (математическое ожидание, дисперсия, максимум, минимум, сортировка);
- динамическую (скользящее среднее, звено запаздывания, экстраполяция, период колебания и т.п.);
- расчет технико-экономических параметров и др.

В сложных случаях возможно подключение математических моделей пользователя, написанных на языке СИ или СИ++. Сложные алгоритмы логического управления могут быть реализованы с использованием встроенного интерпретатора формул. Для использования интерпретатора не требуется знание профессиональных языков программирования.

Далее в редакторе графических примитивов пользователь создаёт прикладную библиотеку графических примитивов - повторяющихся технических символов, используемых в изображении объекта мониторинга. Созданная библиотека примитивов переносится в редактор рисунка, где пользователь с её помощью, а также, используя встроенные графические элементы создаёт статичные мнемосхемы технологического объекта на графическом масштабируемом поле. Статичные мнемосхемы рисуются в графическом редакторе ТМ.

На третьем этапе в редакторе представления данных задаются динамические формы графической визуализации информации. На созданный рисунок объекта накладываются формы отображения, связанные с базой каналов ТМ. При этом привязка форм отображения к текущему каналу базы производится автоматически. Динамизация измеренных и расчётных параметров осуществляется при помощи более чем 200 различных типов форм отображения, включающих графики-тренды, в т.ч. трёхмерные, гистограммы (уровни), мультипликацию на основе растровых и векторных изображений, мультимедиа и многооконный интерфейс.

TRACE MODE позволяет создавать систему распределённых архивов данных. В архивы записываются значения каналов, отмеченных в редакторе базы для данного вида архива. Монитор архива предназначен для: просмотра архива и уровневого архива в графическом виде (как фильма на видеомagneтoфoне в режиме PLAYBACK). Просмотр осуществляется с произвольной скоростью, начиная с произвольного момента времени; создания распределённых систем отображения информации. Например, при помощи монитора архива создаётся административный контур АСУ, предназначенный для руководящего состава предприятия (главного инженера, службы главного технолога, начальников смены и др.). Руководство предприятия имеет возможность в любой момент подключиться к мониторингу технологического процесса, а также посмотреть (например, в ускоренном темпе) как развивался процесс пока их компьютеры были выключены. Файл архива, с которым работает программа монитора архива, пишется монитором реального

времени. В ТМ существуют следующие типы архивов: уровневый архив временных срезов; отчет тревог (сообщения о технологических событиях); хранилище данных (архив параметров по их изменению); локальный архив изменений параметров; генератор отчетов произвольной формы.

В заключение отметим важный момент в программировании в среде ТМ. Он связан с наличием в ТМ встроенных программ поддержки для наиболее распространенных в России контроллеров (Ремиконт, ТСМ51, Ш711, MicroPC, ADAM, РЕР, КРУИЗ, МФК и др.). Поэтому необходимый протокол обмена с контроллером пользователь выбирает в меню драйверов. Причем допускается обмен одновременно с несколькими разнотипными контроллерами. В ТМ поддерживается обмен по 16 последовательным портам со скоростью до 115 Кбод. Каждый порт может быть настроен на свой протокол обмена. Таким образом, с помощью ТМ можно создавать гетерогенные системы или осуществлять поэтапную модернизацию оборудования. В ТМ встроен протокол обмена по последовательному интерфейсу. Это позволяет создавать сети из мониторов ТМ на основе интерфейса RS-485. В такую сеть могут быть включены: МРВ; Win МРВ; Микро МРВ. Обмен осуществляется в режиме MASTER-SLAVE. Возможно создание до 16 параллельных сетей такого типа.

TRACE MODE поддерживает режимы сетевого обмена.

При использовании IBM-совместимых контроллеров таких как MicroPC, КРУИЗ, МФК (Текон), Круиз, ADAM, MIC2000 и др. применяется специальная run-time система - Микро МРВ, которая конфигурируется средствами ТМ. Таким образом, для такого рода контроллеров ТМ является единым инструментом разработки АСУ ТП как верхнего, так и нижнего уровня. Основные функции и характеристики Микро МРВ ТМ: 120 мкс-55мс – время реакции системы; 4000 каналов ввода/вывода; поддержка плат УСО и контроллеров различных фирм; сетевой обмен; прием и первичная обработка данных; реализация алгоритмов управления и регулирования; групповая обработка событий; контроль исправности УСО; поддержка WATCHDOG-таймера; защита от зависания; управление устройствами; поддержка промышленных сетей, поддержка модемной связи, 4 уровня отладки в пошаговом и непрерывном режимах. При помощи Микро МРВ Модем+ можно создавать глобальные распределенные системы сбора информации, обменивающиеся с HOST-машиной через обыкновенную телефонную сеть.

TRACE MODE автоматизирует наиболее сложные этапы создания системы управления: прием сигналов; обработка сигналов; автоматическое управление исполнительными устройствами; визуализация измеренных величин; ведение архивов; генерирование отчетов.

Микро МРВ ТМ реализует технологии визуального программирования контроллеров указанного типа согласно стандарту МЭК 1131-3. Для разработки используются визуальный и текстовый языки, соответственно, Техно FDB и Техно IL. Языки ориентированы на инженера-технолога, а не на программиста. Библиотека алгоритмов содержит более 150 модулей. Микро МРВ ТМ представляет разработчику функции отладки, включающие отладку алгоритмов первичной обработки, Техно FDB, отдельных объектов базы каналов и эмуляции работы всего проекта в пошаговом и непрерывном режиме. В систему входит профайлер, дающий возможность определения временных параметров, потребляемых ресурсов, диагностики ошибок исполнения и накопления статистики системных характеристик Микро МРВ на этапе отладки. При помощи Микро МРВ можно создавать отказоустойчивые резервированные сетевые комплексы на базе последовательного интерфейса, локальных сетей и т.д.

Таким образом, Микро МРВ ТМ, оставаясь SCADA – системой, является современным инструментарием проектировщика программируемых контроллеров АСУ ТП.