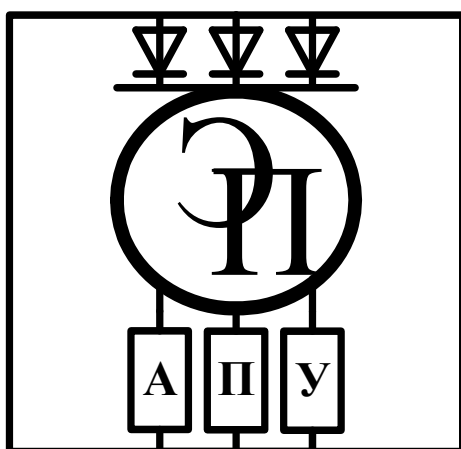


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Электропривод и автоматизация промышленных установок»

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА В АВТОМАТИЗИРОВАННОМ ЭЛЕКТРОПРИВОДЕ

*Методические рекомендации к курсовому проектированию
для студентов специальности 1-53 01 05 «Автоматизированные
электроприводы» очной и заочной форм обучения*



Могилев 2022

УДК 004.3:62-83
ББК 32.973.26-04:31.291
М59

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой ЭП и АПУ «21» декабря 2021 г., протокол № 6

Составитель ст. преподаватель В. Н. Ситников

Рецензент канд. техн. наук, доц. С. В. Болотов

В методических рекомендациях изложены основные сведения, необходимые студентам во время выполнения курсовой работы.

Предназначены для использования студентами специальности 1-53 01 05 «Автоматизированные электроприводы» очной и заочной форм обучения.

Учебно-методическое издание

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА В АВТОМАТИЗИРОВАННОМ ЭЛЕКТРОПРИВОДЕ

Ответственный за выпуск	Г. С. Леневский
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 115 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2022

Содержание

Введение.....	4
1 Цели и задачи курсового проектирования	5
2 Общие вопросы проектирования МПСУ	5
2.1 Этапы проектирования	5
2.2 Функции микропроцессорных устройств управления	7
2.3 Выбор микроконтроллера	9
2.4 Разработка аппаратной части МПСУ	10
2.5 Разработка программного обеспечения МПСУ	12
2.6 Отладка МПСУ	15
3 Технические средства микропроцессорных устройств управления	16
3.1 Общая информация	16
3.2 Архитектурные особенности AVR-микроконтроллеров	19
4 Рекомендации по выполнению и оформлению курсовой работы	33
4.1 Общие методические указания	33
4.2 Содержание курсовой работы.....	33
4.3 Оформление курсовой работы	36
5 Пример микропроцессорной системы.....	37
5.1 Пример на базе микроконтроллера AVR.....	37
5.2 Пример на базе микроконтроллера семейства MCS-51	40
Список литературы	43

Введение

Прогресс в области электронных и информационных технологий существенным образом изменяет как подходы к проектированию систем управления производственными механизмами и установками, так и их элементную базу. За последние десятилетия в практике регулируемого электропривода все шире находят применение средства прямого цифрового управления. Область микропроцессорных систем управления, предназначенная для управления двигателями, получила специальное название – Motor Control. Появление мощных, полностью управляемых полевых транзисторов MOSFET и биполярных транзисторов с изолированным затвором IGBT привело к развитию преобразовательной техники и расширению сферы применения синхронных и асинхронных электроприводов с преобразователями частоты. Другим фактором, обусловившим совершенствование регулируемого электропривода и расширение областей его применения, было создание микропроцессоров и однокристальных микроконтроллеров достаточной вычислительной мощности.

Существующая широкая номенклатура микроконтроллеров предназначена для разнообразных сфер применения и удовлетворяет самым разнообразным требованиям к параметрам локальных цифровых регуляторов. Новые решения в микропроцессорной технике, в том числе в архитектуре микроконтроллеров, исходят из обобщения аппаратных принципов систем управления. Именно совместные усилия специалистов различного профиля привели к созданию семейств микроконтроллеров, предназначенных для управления электродвигателями.

Разработка, производство и эксплуатация микропроцессорных устройств управления требуют подготовки соответствующих специалистов, обладающих достаточно глубокими знаниями в области цифровой и аналоговой схемотехники, знаниями принципов построения, организации и функционирования современных микроконтроллеров, а также программирования.

В ходе выполнения курсовой работы студенты изучат архитектурные особенности конкретного семейства микроконтроллеров, освоят методику проектирования микропроцессорных устройств управления. Процесс проектирования микропроцессорных систем очень трудно формализовать и в нем ярче всего проявляются как способности студента, так и его желания.

1 Цели и задачи курсового проектирования

Выполнение курсовой работы по дисциплине является заключительным этапом подготовки студентов по соответствующей дисциплине и имеет своей целью систематизацию, закрепление и расширение теоретических знаний и навыков, полученных студентом на предыдущих этапах обучения.

Работая над проектом, студент должен овладеть методами и приёмами работы с микропроцессорной техникой, получить знания в области разработки микропроцессорных систем управления (МПСУ), научиться самостоятельно работать с технической литературой.

При выполнении курсовой работы студент должен использовать знания, полученные им при изучении других дисциплин, уметь в короткие сроки осваивать новые разделы в пределах изученных дисциплин, творчески подходить к работе.

Самостоятельное выполнение курсовой работы подготавливает студента к успешному выполнению других курсовых проектов, дипломного проекта и является важным этапом в формировании специалиста в области автоматизированного электропривода.

В курсовой работе рассматриваются конкретные примеры реализации микропроцессорных систем управления, производится выбор элементов аппаратной части устройства, составление алгоритмов работы системы, разработка программного обеспечения.

В курсовой работе необходимо решить следующие задачи:

- разработать функциональную схему устройства управления;
- разработать аппаратную часть микропроцессорного устройства, выполняющего заданные функции на базе определенного микроконтроллера;
- разработать структуру программного обеспечения микропроцессорного устройства, а также составить алгоритмы и программы управления.

2 Общие вопросы проектирования МПСУ

2.1 Этапы проектирования

Программно-аппаратная реализация функций управления в микропроцессорных системах обуславливает особый подход к процессу проектирования таких систем. Прежде всего это связано с необходимостью совместной разработки аппаратных средств и программного обеспечения системы, причём объём затрат на разработку аппаратуры постоянно уменьшается по отношению к суммарным затратам на программирование.

При проектировании МПСУ рекомендуется использовать концепцию нисходящего проектирования, сущность которой заключается в том, что проектируемую систему разбивают на несколько уровней представления. На высшем уровне используется наименее детализированное описание системы, отражаю-

шее только самые общие черты и особенности системы, на следующих уровнях степень подробности рассмотрения возрастает, при этом система рассматривается не в целом, а отдельными блоками. Такой подход позволяет разбить процесс проектирования на несколько этапов, на каждом из которых решаются задачи реализации определённого уровня представления системы. При этом сложные задачи большой размерности разбиваются на группы последовательно решаемых простых задач.

При нисходящем проектировании МПСУ исходят из желаемых свойств системы, описанных на высшем уровне. Постепенно уточняя и разбивая задачи управления на более мелкие, приходят к низшему уровню, на котором система представлена в виде совокупности электронных схем и программ реализации простейших функций управления.

Весь процесс проектирования МПСУ может быть представлен в виде алгоритма, приведенного на рисунке 2.1.



Рисунок 2.1 – Алгоритм проектирования МПСУ

На первом этапе проектирования формируются требования к системе, уточняются цели и задачи управления, определяются особенности и технические характеристики объекта управления.

На втором этапе осуществляется функциональное проектирование системы, в ходе которого определяется последовательность решения задач, необхо-

димые средства сопряжения устройства управления с объектом, устанавливаются потоки информации между задачами, формируются основные требования к архитектуре микроконтроллера. Результатом данного этапа должна явиться разработка функциональной схемы системы управления, которая представляется в виде совокупности функциональных блоков, связанных между собой потоками информации.

Каждый функциональный блок решает определённую задачу управления и может быть реализован как аппаратными, так и программными средствами. Выбор способа реализации в каждом конкретном случае обосновывается техническими или экономическими соображениями.

Один из наиболее ответственных этапов проектирования микропроцессорных систем автоматического управления – выбор микропроцессорных средств, который производится с учётом требований к проектируемой системе и возможностей конкретного микроконтроллера. Правильный выбор микроконтроллера во многом предопределяет успешную реализацию проекта.

Следующие два этапа, в ходе которых осуществляется проектирование аппаратных средств и программного обеспечения, выполняются параллельно. При проектировании аппаратных средств сначала разрабатывается конфигурация системы, определяются способы взаимодействия элементов, производится выбор устройств ввода-вывода и памяти, разрабатывается принципиальная схема и изготавливается прототип. На этапе программирования определяется состав программного обеспечения, строятся алгоритмы решения функциональных задач, выбирается язык программирования, производится трансляция управляющей программы, написанной на языке программирования, в коды команд микроконтроллера.

На заключительном этапе производится объединение аппаратных средств и программного обеспечения в единый комплекс, отладка системы и оценка её эксплуатационных характеристик.

2.2 Функции микропроцессорных устройств управления

Основное содержание процесса управления заключается в последовательном формировании управляющих взаимодействий, обеспечивающих достижение целей управления.

Управляющие сигналы формируются в соответствии с заранее определённым желаемым поведением объекта и на основании информации о состоянии объекта. Характер управляющих сигналов, алгоритмы и последовательность обработки входной информации определяются особенностями объекта и задачами управления.

Каждая конкретная система автоматического управления характеризуется своей совокупностью решаемых задач, однако любая из них выполняет три основные функции:

- 1) ввод и предварительное преобразование входной информации;
- 2) обработка входной информации и решение основных функциональных задач;

3) выработка управляющих взаимодействий и выдача их на объект управления.

Выполнение первой функции в МПСУ заключается в основном в предварительной фильтрации входного сигнала, преобразовании его в форму, удобную для дальнейшей обработки, организации ввода в микроконтроллер. При этом информация, поступающая в виде аналоговых сигналов напряжения или тока, должна быть преобразована в цифровой код.

Обработка входной информации связана с контролем правильности приёма, выделением полезного сигнала и фильтрацией помех, которые обусловлены случайными ошибками измерений, масштабированием и нормализацией входных сигналов, реализацией алгоритмов регулирования и управления состоянием объекта. Большое количество задач в МПСУ решаются с целью организации взаимодействия с оператором, оптимального распределения ресурсов, адаптации системы к изменяющимся условиям. Все эти задачи предназначены для выполнения второй функции системы управления.

Задачи вывода управляющих воздействий включают в себя выполнение следующих основных процедур:

- расчёт параметров управляющих сигналов;
- формирование сигналов по форме и мощности в соответствии с требованиями исполнительных элементов объекта управления.

Во время функционирования объекта некоторые его параметры могут выходить за границы заранее заданного диапазона, могут происходить события, требующие кардинального изменения режима работы. В этой связи реализация защит является еще одной важной функцией, которую должны выполнять микропроцессорные устройства управления.

Кроме вышеперечисленных, особое место в МПСУ занимают задачи тестирования и самодиагностики, решение которых позволяет в значительной степени улучшить эксплуатационные показатели системы, облегчить поиск неисправностей и уменьшить время восстановления.

При разработке функциональной схемы системы следует учитывать особенности, присущие МПСУ:

- работу в реальном масштабе времени, т. е. необходимость обеспечивать минимальное время реакции на изменение внешних условий;
- наличие различных режимов работы;
- число решаемых задач в каждом режиме ограничено;
- использование различных устройств для организации связи с объектом;
- программную и аппаратную реализацию функций управления.

2.3 Выбор микроконтроллера

Выбор оптимального микроконтроллера для конкретного применения является достаточно сложной процедурой. Он связан с постоянным ростом количества различных типов микроконтроллеров, расширением области их применения, а также отсутствием чёткой методики, позволяющей сделать однозначный выбор.

Микроконтроллер является функционально сложным устройством и характеризуется большим количеством параметров, важность которых в конкретном применении неодинакова, иногда низкое значение одного из параметров может быть компенсировано высоким значением другого. Более того, в ряде случаев важнейшими являются характеристики, которые не имеют особого значения в большинстве применений. Поэтому выбор микроконтроллера – это результат компромисса между его параметрами и требованиями конкретного использования.

При выборе микроконтроллера для реализации МПСУ следует учитывать основные требования, предъявляемые к разрабатываемой системе: время решения задач управления, точность проведения расчётов, условия эксплуатации, надёжность.

Выбор микроконтроллера обычно производится с учётом следующих критериев:

- быстродействие микроконтроллера, которое определяется в основном частотой внутреннего или внешнего генератора синхронизации;
- производительность микропроцессорного устройства на базе данного микроконтроллера, которая оценивается временем выполнения типовых задач управления и, кроме быстродействия, определяется такими элементами архитектуры, как система команд, система прерываний и т. д.;
- состав и параметры встроенных периферийных устройств (портов ввода-вывода, таймеров, аналого-цифровых преобразователей и т. д.);
- наличие аппаратных средств для организации стандартных интерфейсов обмена информацией с другими устройствами;
- объем и тип внутренней памяти программ и данных, наличие энергонезависимой перепрограммируемой памяти данных, возможность подключения внешней памяти;
- наличие программного обеспечения для разработки, трансляции, отладки программ для данного микроконтроллера и устройств для его программирования;
- электрическая совместимость с другими интегральными схемами, используемыми в устройстве;
- потребляемая мощность и наличие режимов энергосбережения;
- условия эксплуатации (диапазон рабочих температур, наличие агрессивных сред и т. д.).

Многие производители выпускают микроконтроллеры, ориентированные для управления электрическими двигателями, и это также необходимо учитывать при выборе.

Немаловажное значение для успешной реализации проекта играет и наличие у разработчика опыта работы с выбранным семейством микроконтроллеров.

2.4 Разработка аппаратной части МПСУ

Аппаратные средства, входящие в состав микропроцессорных систем управления, можно разделить на следующие основные части (рисунок 2.2):

- вычислительная часть или микроконтроллер;

- устройства сопряжения с объектом (УСО);
- пульт оператора;
- устройства связи;
- датчики и исполнительные устройства.

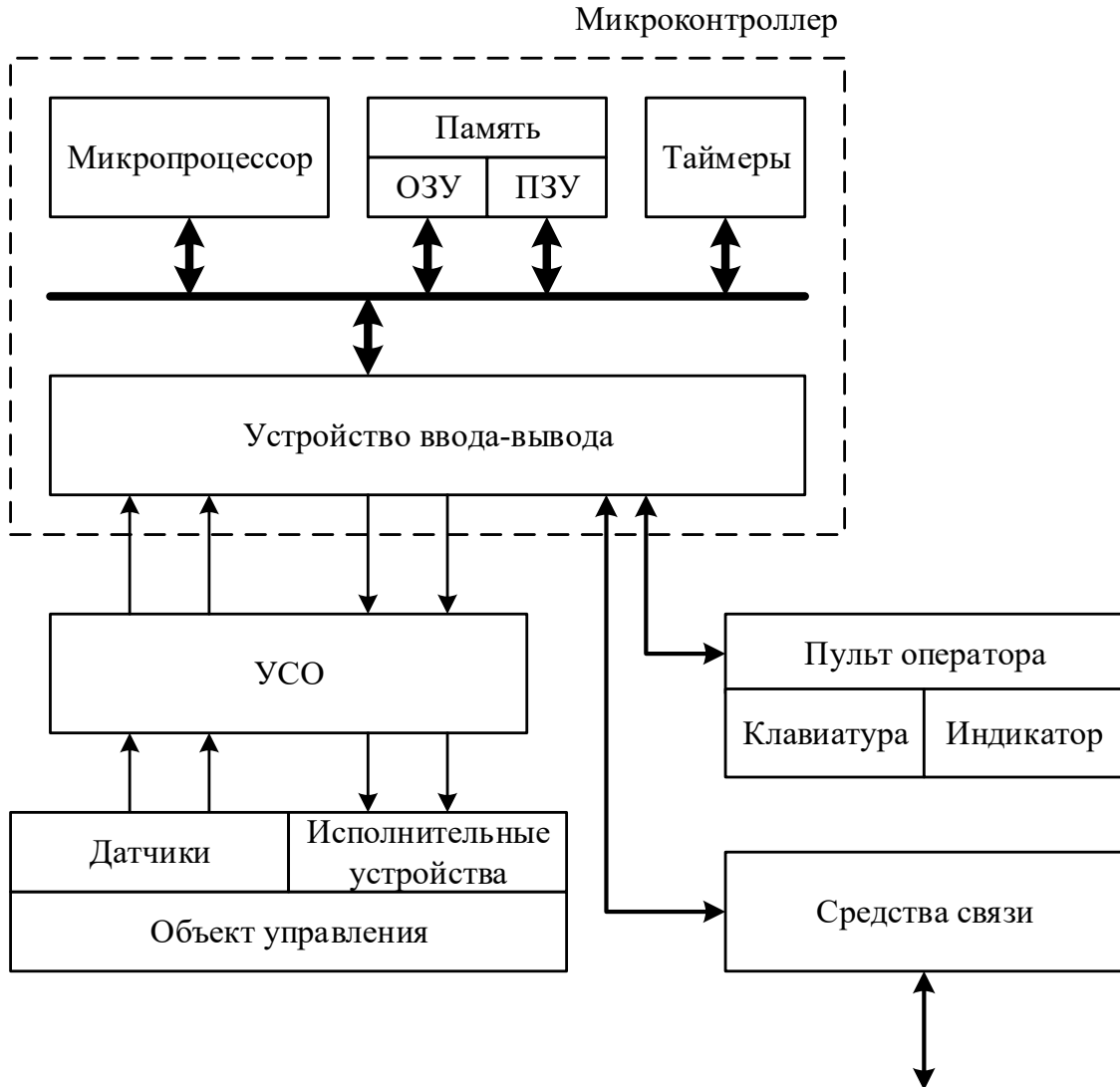


Рисунок 2.2 – Структурная схема микропроцессорной системы управления

Центральным элементом системы является микроконтроллер, состоящий из четырех основных модулей: микропроцессора, памяти, таймеров и устройств ввода-вывода.

Микропроцессор осуществляет обработку информации в соответствии с алгоритмом управления, выполняя все функции микропроцессорного устройства управления, реализованные программно. Принципы обработки информации, назначение и взаимодействие основных элементов микропроцессора такие же, как и во многих других вычислительных устройствах.

Память подразделяется на память данных и память программ. Память данных реализуется в виде оперативного запоминающего устройства (ОЗУ) и предназначена для хранения исходных данных, промежуточных и конечных

результатов расчетов. Память программ содержит управляющую программу и представляет собой постоянное запоминающее устройство (ПЗУ), построенное на элементах масочной, однократно-программируемой или перепрограммируемой памяти. Запись управляющей программы в однократно программируемую или перепрограммируемую память осуществляется с помощью специальных устройств – программаторов в процессе программирования микроконтроллера. Программирование микроконтроллеров с масочным ПЗУ производится при их изготовлении.

Объем памяти может изменяться в широких пределах в зависимости от сложности алгоритмов управления и количества обрабатываемой в процессе управления информации. При выборе микроконтроллера необходимо предусмотреть наличие в нем требуемого объема ОЗУ и ПЗУ или выбирать микроконтроллер, в котором предусмотрена возможность подключения внешней памяти. Для реализации внешней памяти используются микросхемы памяти, выпускаемые промышленностью в достаточной номенклатуре типов и отличающиеся технологическим исполнением, значением параметров, режимами работы и областью применения.

Разработка модуля памяти в этом случае заключается в выполнении работ по определению, выборе микросхем памяти, определению организации модуля и его сопряжения с микроконтроллером.

Кроме ОЗУ и ПЗУ, некоторые микроконтроллеры имеют энергонезависимую перепрограммируемую память, ячейки которой доступны микропроцессору для чтения и записи данных и сохраняют информацию при отключении питания.

Устройство ввода-вывода обеспечивает передачу данных между микроконтроллером и другими элементами МПСУ и состоит из отдельных каналов (портов), которые, в свою очередь, состоят из отдельных линий ввода или вывода. Количество портов и линий в каждом порту зависит от типа микроконтроллера. Некоторые линии могут менять назначение в зависимости от режима работы микроконтроллера и его периферийных устройств, настройка их на конкретную функцию производится с помощью управляющих регистров микроконтроллера.

При проектировании МПСУ необходимо учитывать способы передачи информации между микроконтроллером и определенным источником или приемником, требуемое количество линий ввода-вывода и их функциональные возможности.

Таймеры, входящие в состав микроконтроллера, используются при реализации временных функций и обеспечивают управление в реальном масштабе времени.

Непосредственная физическая связь микроконтроллера с датчиками и исполнительными устройствами, установленными на объекте, осуществляется через УСО. Их наличие в любой МПСУ обусловлено различием представления информации при ее обработке в микроконтроллере и на вводах-выводах объекта управления, многообразием используемых датчиков контроля состояния объекта и исполнительных механизмов. В общем случае УСО содержит в своем

составе схемы ввода и вывода аналоговых и дискретных сигналов, которые обеспечивают сбор информации с датчиков и выдачу управляющих сигналов на исполнительные механизмы.

Выбор конкретных схем для УСО производят с учетом типа используемых датчиков и исполнительных механизмов. При этом необходимо, чтобы устройства обеспечивали нужную скорость и достоверность передачи информации, обладали требуемой помехоустойчивостью, надежностью и были экономически обоснованы.

Пульт оператора применяется для оперативного управления объектом, задания режимов работы, запуска и контроля. Пульты могут быть автономными и встроенными, иметь специальную или универсальную клавиатуру, различные устройства индикации и регистрации параметров объекта.

Устройство связи используется для обмена информацией с другими электронными устройствами при организации многоуровневых иерархических систем управления. Для этих целей в микропроцессорных системах управления электроприводами могут применяться такие интерфейсы, как CAN, RS-485. Кроме того, использование таких интерфейсов, как CAN, Ethernet, позволяет обеспечить возможность удаленного мониторинга и изменения параметров микроконтроллера с персональных компьютеров, что позволяет отказаться от отдельного пульта оператора.

2.5 Разработка программного обеспечения МПСУ

Разработка программного обеспечения МПСУ имеет особенности, которые обусловлены следующими причинами:

- необходимостью реализации программ управления в реальном масштабе времени, т. е. со строго ограниченным временем выполнения;
- необходимостью синхронизации выполнения программ с внешними событиями;
- большой связанностью программ, т. е. наличием большого числа переменных, общих для различных функциональных задач.

Кроме этого, к программам управления в МПСУ предъявляются повышенные требования в отношении устойчивости к различного рода сбоям и ошибкам.

Процесс разработки прикладного программного обеспечения обычно выполняется в следующей последовательности: производится разработка алгоритмов решения задач управления, затем осуществляется кодирование программы на языке программирования, ее трансляция в машинные коды микроконтроллера и отладка.

Разработка алгоритмов управления является наиболее ответственным этапом, от которого существенно зависит качество процесса управления и технические параметры разрабатываемой МПСУ. Даже для относительно простой системы трудно разработать алгоритм, охватывающий все детали и особенности управления объектом, поэтому рекомендуется использовать последовательные уровни детализации алгоритма:

- концептуальную блок-схему программного обеспечения;

- функциональную блок-схему решения отдельных задач;
- машинно-ориентированные блок-схемы.

Концептуальная блок-схема содержит небольшое количество блоков, разделяющих состав программного обеспечения и основные задачи, решаемые программными средствами, которые можно разделить на два класса:

- 1) прикладные, связанные с формированием управления по заданным алгоритмам регулирования;
- 2) системные, связанные с организацией вычислений, управления техническими средствами МПСУ, обмена информацией и обслуживанием пульта.

Две первые схемы могут быть сделаны безотносительно к какому-либо микропроцессорному комплекту, третья же должна учитывать особенности архитектуры конкретного микроконтроллера, на базе которого реализована МПСУ.

Трудоемкость программирования и качество разрабатываемой программы в значительной степени определяется используемым языком программирования. При программировании на машинном языке все элементы программы представляются в двоичной форме (иногда для более компактной записи используются восьмеричные или шестнадцатеричные числа). При этом программист имеет возможность непосредственно управлять всеми программно-доступными элементами МПСУ, полностью контролировать каждый шаг, выполняемый системой. Это позволяет оптимизировать программу с точки зрения времени её выполнения и требуемого для её размещения объёма памяти.

Однако программирование на машинном языке – утомительный и трудоёмкий процесс, требующий от программиста запоминания кодов многочисленных команд, входящих в систему команд микроконтроллера, использование абсолютных адресов ячеек памяти, что неудобно в особенности при протяженных программах, содержащих большое число переходов. Программа, составленная в машинных кодах, трудна для понимания и модификации. Поэтому машинный язык применяется редко и, как правило, при составлении коротких и крайне эффективных по размеру и быстродействию программ.

При программировании на языке ассемблера вместо двоичных кодов операций используются их символические буквенные обозначения – мнемокоды. Кроме этого, ассемблер позволяет записать в символическом виде данные и адреса, используемые в программе, что существенно облегчает процесс программирования и чтения программ. Язык ассемблера, как и машинный язык, требует от программиста свободного владения архитектурой соответствующего МК. В этом смысле оба языка являются машинно-ориентированными.

Программа на языке ассемблера состоит из ряда строк, каждая из которых имеет четыре поля: поле метки, поле кода операции, поле операндов и поле комментариев. Назначение полей следующее:

- поле метки содержит символическое имя, состоящее из букв и цифр, однозначно определяющее абсолютный адрес, по которому хранится оператор. Метка позволяет обращаться к данному оператору из любой точки программы в ходе её выполнения и может отсутствовать, если в программе нет ссылок на оператор. Поле метки отделяется от поля кода операции двоеточием;

- поле кода операции содержит мнемоническое обозначение кода операции, приводимое в системе команд микропроцессора, или символическое обозначение директивы ассемблера (псевдокоманды), которая используется при трансляции программы;

- поле операнда содержит один или два операнда, разделенных запятой, и отражает информационное содержание команд микропроцессора и директив ассемблера, указанных в поле кода операции. В качестве операндов могут использоваться численные или символьные константы, имена, выражения;

- поле комментария служит для записи информации, поясняющей выполняемые действия. Комментарии нужны программисту, они не воспринимаются МПСУ, а лишь повышают удобство чтения. Поле комментариев обычно отделено от предыдущего поля точкой с запятой.

Программирование на языке высокого уровня существенно повышает производительность работы программиста, обеспечивает естественную форму записи вычислительного процесса, но не всегда позволяет использовать технические возможности МПСУ, а полученные после трансляции результирующие машинные программы значительно медленнее и длиннее аналогичной программы на машинном языке. В качестве языка программирования высокого уровня для определенного семейства микроконтроллеров широко используются соответствующие варианты языка СИ.

Разработка управляющих программ для микроконтроллеров производится с применением различных инструментальных средств в виде программных комплексов, реализованных на базе IBM-совместимых компьютеров или других вычислительных устройств. Такой комплекс включает в себя различные программы, позволяющие вводить, редактировать текст управляющей программы, транслировать его в коды команд микроконтроллера, компоновать программу из нескольких программных модулей, производить отладку разработанной программы, используя программную модель микроконтроллера. Кроме этого, часто имеется возможность записать отлаженную программу в память микроконтроллера.

2.6 Отладка МПСУ

Отладка микропроцессорных систем – это процесс обнаружения ошибок, допущенных на предыдущих этапах проектирования, их исправления и обеспечения правильности функционирования системы в реальных условиях. Отладка может быть разделена на несколько этапов, в ходе выполнения которых осуществляется автономная отладка аппаратной части и программного обеспечения, комплексная отладка всей системы в целом.

Отладка аппаратуры предполагает статическую и динамическую проверку. В статике проверяются целостность электрических соединений, работоспособность элементов и величина напряжений в контрольных точках. Динамическая проверка позволяет выявить основной процент неисправностей в аппаратуре. Она проводится при выполнении процессором тестовой программы. При этом проверяется происхождение данных в системе, временные диаграммы сигналов

в контрольных точках и взаимодействие отдельных устройств МПСУ. Для локализации неисправностей широко используются различные приборы: осциллографы, вольтметры, амперметры, частотомеры, генераторы тестовых сигналов, логические и сигнатурные анализаторы.

Этап отладки программ МПСУ производится с использованием специальных средств: программных и аппаратно-программных. Программные средства отладки – комплекс программ, позволяющих выводить на дисплей содержимое ячеек памяти и регистров микропроцессора; изменить содержимое регистров и любой ячейки оперативной памяти; начинать выполнение программы с любой ячейки памяти; останавливать выполнение программы по достижении команды, находящейся в определенной ячейке памяти или при выполнении какого-либо условия. Все это дает возможность реализовать прикладную программу в пошаговом режиме, в режиме трассировки, в режиме остановки программы в контрольных точках. При работе прикладных программ в этих режимах эффективно обнаруживаются ошибки и анализируются их причины.

В пошаговом режиме выполнение отлаживаемой программы прекращается после каждой команды микропроцессора и возобновляется вновь по команде оператора, при этом на дисплее отображается информация о состоянии регистров микропроцессора, ячеек памяти, используемых в последней команде, и несколько следующих команд.

В режиме трассировки автоматически выполняется вся или часть программы до останова извне, при этом на дисплее отображается содержимое регистров после выполнения каждой команды. Программист, анализируя эти данные, может обнаружить ошибку.

В режиме остановки на контрольной точке место останова выполнения программы задается в виде адреса, кода команды, слова состояния микропроцессора, содержимого регистров и комбинации кодовых слов. Это дает возможность контролировать порядок выполнения произвольных ветвей прикладной программы.

Отладка программного обеспечения только программными средствами происходит без связи с соответствующими аппаратными средствами МПСУ, что снижает эффективность этих средств и не позволяет выявить все ошибки программирования.

Аппаратно-программные средства обеспечивают комплексную отладку прикладных программ и аппаратуры, соответствующей МПСУ, путем моделирования работы аппаратных средств микропроцессорной системы. При этом пошаговый метод, трассировка и остановка по контрольным точкам позволяют постепенно вовлекать в работу все большую часть программных средств, а средства имитации аппаратуры МПСУ помогают последовательно охватывать отладкой все большую часть соответствующих аппаратных средств. Этим достигается высокая эффективность комплексной отладки всей МПСУ.

3 Технические средства микропроцессорных устройств управления

3.1 Общая информация

Технические средства разрабатываемой микропроцессорной системы (порты ввода/вывода, УСО) должны обеспечить ввод сигналов от датчиков, установленных на объекте управления, и вывод сигналов управления на исполнительные устройства объекта. Эти сигналы могут быть как дискретными, так и аналоговыми.

При вводе дискретных сигналов может потребоваться решить следующие задачи:

- преобразовать дискретный сигнал в цифровой по уровню, т. е. обеспечить соответствие входного сигнала логическим «0» и «1» микропроцессора;
- произвести гальваническую развязку;
- подавить помеху и устранитьдребезг контактов;
- обеспечить индикацию состояния входа.

Гальваническую развязку при вводе дискретных сигналов можно реализовать электромагнитным, электромеханическим или оптическим способом.

Так, например, канал ввода дискретного сигнала с использованием оптопары для гальванической развязки может иметь вид, приведенный на рисунке 3.1.

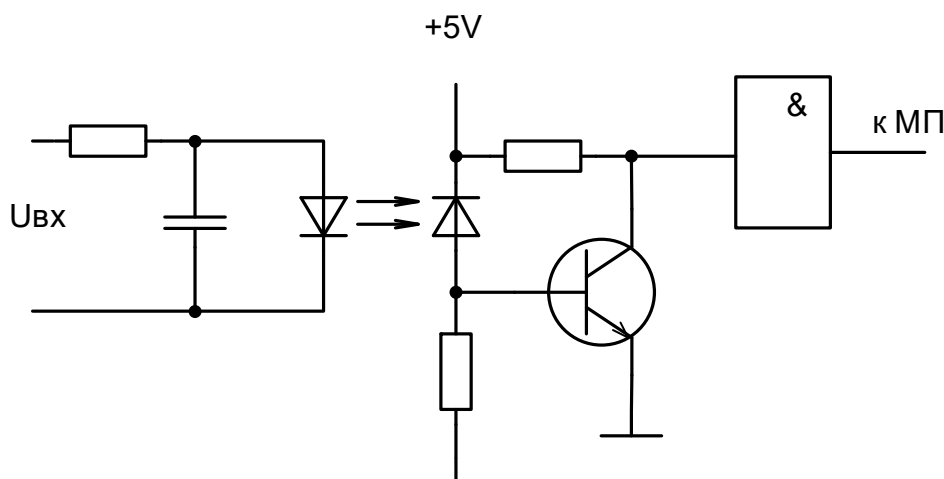


Рисунок 3.1 – Пример схемы ввода дискретного сигнала

При выводе дискретных сигналов может понадобиться решить следующие задачи:

- запомнить значение дискретного сигнала;
- произвести гальваническую развязку;
- усилить сигнал по уровню;
- обеспечить индикацию состояния выхода.

Для сохранения значения выходного сигнала в микропроцессорных системах можно использовать регистры и триггеры. Кроме того, современные мик-

роконтроллеры имеют встроенные порты, в состав которых входят регистры, использование которых позволяет не применять в системе дополнительные микросхемы.

При вводе аналоговых сигналов необходимо:

- отфильтровать сигнал;
- усилить до уровня преобразования АЦП;
- гальванически развязать сигнал;
- преобразовать аналоговый сигнал в цифровой код.

Аналоговые сигналы поступают в микропроцессорную систему от датчиков физических величин. Такие сигналы часто подвержены помехам. Как правило, информация с датчиков используется для расчета выходного сигнала управления, и часто в процессе обработки этот сигнал в регуляторах усиливается, что приводит к усилению помехи и ухудшению точности регулирования. Таким образом, устранение помехи из аналогового сигнала – очень важная задача. Самым простым способом фильтрации является применение пассивных фильтров, например, RC-цепей. Более сложным способом является применение активных фильтров на базе операционных усилителей.

Для преобразования аналогового сигнала в цифровой код в микропроцессорных системах используется аналогово-цифровой преобразователь (АЦП). Выбор АЦП производится по его параметрам, основными из которых являются:

- разрядность, характеризующая точность преобразования;
- диапазон входного напряжения;
- время преобразования;
- тип цифрового интерфейса.

Большинство моделей современных однокристальных микроконтроллеров оснащены встроенными АЦП, использование которых позволяет значительно упростить систему. Однако в некоторых случаях характеристики встроенных АЦП могут не удовлетворять предъявляемым требованиям (например, слишком малая разрядность или большое время преобразования), поэтому в таких системах необходимо использовать внешние преобразователи.

При выводе аналоговых сигналов необходимо:

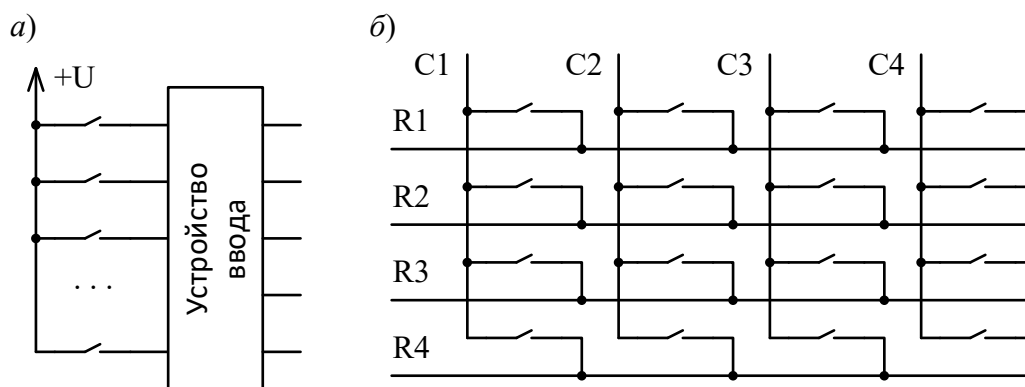
- запомнить значение выходного кода;
- произвести гальваническую развязку;
- преобразовать цифровой код в аналоговый сигнал;
- усилить сигнал по уровню.

Для преобразования цифрового кода в аналоговый необходимо использовать цифро-аналоговый преобразователь (ЦАП), выбор которого осуществляется по следующим параметрам:

- разрядность входного кода;
- время установления выходного напряжения;
- максимальная частота преобразования;
- коэффициент преобразования – отношение приращения выходного сигнала к приращению входного сигнала для линейной функции преобразования.

Многие современные однокристальные микроконтроллеры имеют встроенный цифро-аналоговый преобразователь, что также позволяет упростить принципиальную схему системы.

Пульт оператора используется для обеспечения связи МПСУ с оператором. Органы пульта состоят из элементов, позволяющих в удобном для человека виде вводить и выводить информацию. Так, для ввода информации может использоваться клавиатура, которая может строиться линейным или матричным способом (рисунок 3.2).



a – линейная; *б* – матричная

Рисунок 3.2 – Клавиатура

Устройство индикации пульта может строиться с использованием:

- светодиодов;
- семисегментных индикаторов;
- жидкокристаллических индикаторов (матричных или графических).

Например, на рисунке 3.3 приведена структурная схема построения устройства динамической семисегментной индикации.

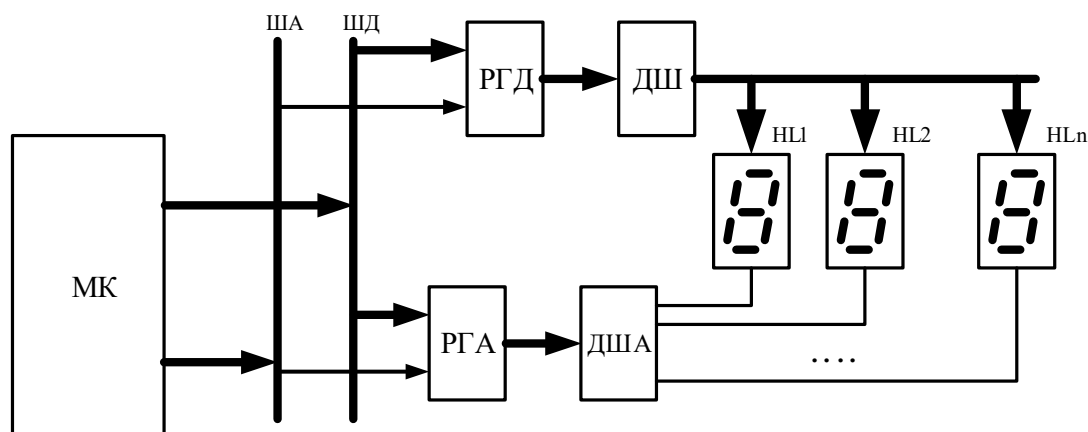


Рисунок 3.3 – Устройство динамической индикации

В качестве индикатора в такой схеме могут использоваться или набор одноразрядных индикаторов, или многоразрядные индикаторы.

Выбор элементной базы (типы микросхем, уровни сигналов), используемой для построения МПСУ, производится на основании параметров применяемого в системе микропроцессорного устройства.

3.2 Архитектурные особенности AVR-микроконтроллеров

В рамках единой базовой архитектуры микроконтроллеры AVR подразделяются на три семейства:

- 1) Classic AVR;
- 2) Mega AVR;
- 3) Tiny AVR.

Микроконтроллеры семейства Classic (впрочем, как и все микроконтроллеры AVR фирмы Atmel) являются 8-разрядными микроконтроллерами, предназначенными для встраиваемых приложений. В семейство Classic входят микроконтроллеры с различным сочетанием периферийных устройств, различными объемами встроенной памяти и различным количеством выводов. Такое разнообразие дает разработчику возможность сделать оптимальный выбор и использовать именно тот микроконтроллер, который наилучшим образом подходит для его нужд.

Микроконтроллеры семейства Tiny имеют небольшие объемы памяти программ (1...2 Кбайта) и весьма ограниченную периферию. Практически все они выпускаются в 8-выводных корпусах и предназначены для так называемых «бюджетных» решений, принимаемых в условиях жестких финансовых ограничений. Область применения этих микроконтроллеров – интеллектуальные датчики различного назначения (контрольные, пожарные, охранные), игрушки, зарядные устройства, различная бытовая техника и другие подобные устройства.

Микроконтроллеры семейства Mega, напротив, имеют наиболее развитую периферию, наибольшие среди всех микроконтроллеров AVR объемы памяти программ и данных. Они предназначены для использования в мобильных телефонах, контроллерах различного периферийного оборудования (принтеры, сканеры, современные дисковые накопители, приводы CD-ROM/DVD-ROM и т. п.), сложной офисной технике и т. д.

При этом в настоящее время семейство Classic считается устаревшим и использование микросхем данного семейства в новых разработках не рекомендуется, хотя приобрести их можно. Выбрать же подходящий микроконтроллер можно из семейств Tiny и Mega.

Говорить об отличительных особенностях микроконтроллеров сложно, поскольку все семейства представлены большим количеством микросхем, значительно отличающихся друг от друга набором периферийных узлов. Выделим основные отличительные особенности:

- усовершенствованная RISC-архитектура, предполагающая наличие набора команд, состоящего из минимума компактных и быстроисполняющихся инструкций; типовая инструкция выполняется за один такт;
- контроллеры построены по гарвардской архитектуре, поэтому имеют отдельные шины памяти команд и данных, что позволяет ускорить выполне-

ние программы за счет одновременной выборки команд и данных;

- наличие трех типов памяти: FLASH-память программ объемом от 1 до 8 Кбайт для контроллеров Tiny, от 1 до 256 Кбайт – для Mega (число циклов стирания/записи не менее 1000); оперативная память (статическое ОЗУ) объемом до 2 Кбайт для Tiny и до 8 Кбайт для Mega; память данных на основе ЭС-ППЗУ (EEPROM) объемом до 512 байт для Tiny и до 4 Кбайт для Mega (число циклов стирания/записи не менее 100000);

- возможность защиты от внешнего чтения и модификации памяти программ и данных (в EEPROM);

- различные способы синхронизации: встроенный генератор с внутренней или внешней времязадающей RC-цепочкой; встроенный генератор с внешним резонатором (пьезокерамическим или кварцевым); внешний сигнал синхронизации;

- полностью статическая архитектура; минимальная тактовая частота равна нулю;

- многоуровневая система прерываний;

- наличие портов ввода/вывода, каждый вывод которых может быть независимо запрограммирован как вход или выход;

- встроенные устройства для обработки аналоговых сигналов: аналоговый компаратор и многоканальный АЦП;

- наличие таймеров-счетчиков с предустановкой и возможностью выбора счетных импульсов;

- наличие сторожевого таймера, позволяющего осуществлять автоматическую перезагрузку контроллера через равные промежутки времени;

- встроенный монитор питания – детектор падения напряжения;

- наличие нескольких режимов пониженного энергопотребления;

- диапазон напряжений питания от 2,7 до 5,5 В (возможна работа при пониженном до 1,8 В напряжении питания).

Ядро микроконтроллеров AVR выполнено по усовершенствованной RISC (enhanced RISC)-архитектуре (рисунок 3.4), в которой используется ряд решений, направленных на повышение быстродействия микроконтроллеров.

Арифметико-логическое устройство (АЛУ), выполняющее все вычисления, подключено непосредственно к 32-м рабочим регистрам, объединенным в регистровый файл. Благодаря этому АЛУ выполняет одну операцию (чтение содержимого регистров, выполнение операции и запись результата обратно в регистровый файл) за один машинный цикл. Кроме того, в микроконтроллерах семейства Tiny каждая из команд занимает только одну ячейку памяти программ.

Периферийные модули позволяют осуществить сопряжение контроллера с внешними устройствами. В состав периферийных узлов всех контроллеров семейства AVR входят АЦП, аналоговый компаратор, порты ввода-вывода, модуль прерываний, как минимум один таймер общего назначения.

В микроконтроллерах AVR реализована гарвардская архитектура, которая характеризуется отдельной памятью программ и данных, каждая из которых имеет собственные шины доступа к ним. Такая организация позволяет одновременно работать как с памятью программ, так и с памятью данных. Разделение шин доступа позволяет использовать для каждого типа памяти шины раз-

личной разрядности, а также реализовать конвейеризацию. Конвейеризация заключается в том, что во время исполнения текущей команды производится выборка из памяти и дешифрация кода следующей команды.

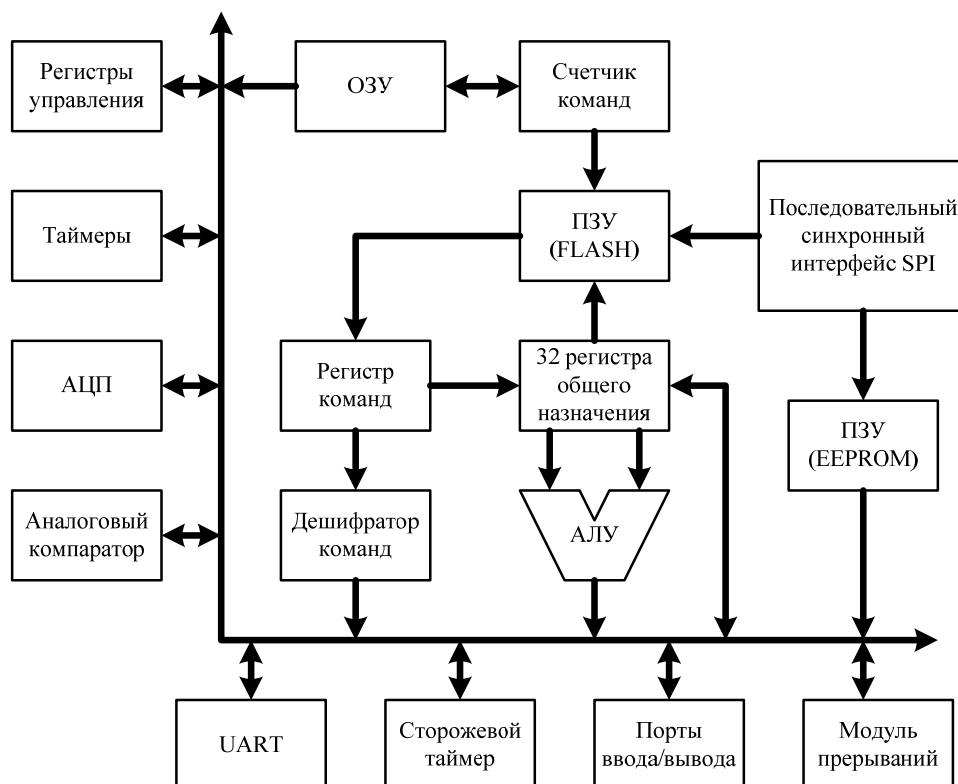


Рисунок 3.4 – Архитектура ядра микроконтроллеров AVR

В отличие от RISC-микроконтроллеров других фирм, в микроконтроллерах AVR используется двухуровневый конвейер, а длительность машинного цикла составляет всего один период кварцевого резонатора. В результате при более низкой тактовой частоте они могут обеспечивать ту же производительность, что и RISC-микроконтроллеры других фирм.

Как было указано, организация памяти микроконтроллеров AVR выполнена по схеме гарвардского типа, в которой разделены не только адресные пространства памяти программ и памяти данных, но также и шины доступа к ним. Память данных состоит из двух областей: регистровая память и память на основе EEPROM. Каждая область расположена в своем адресном пространстве.

Обобщенная карта памяти микроконтроллеров AVR семейства Tiny приведена на рисунке 3.5. При этом в связи с тем, что различные контроллеры отличаются друг от друга объемом памяти, адреса последней ячейки памяти Flash, ОЗУ и EEPROM обозначены символами F_END, S_END и E_END соответственно и для каждой из моделей микроконтроллеров принимают конкретное значение.

Кроме того, необходимо обратить внимание на следующие моменты. Поскольку микроконтроллеры AVR имеют 16-разрядную систему команд, объем памяти программ на рисунке указан не в байтах, а в 16-битных словах.

Символ «\$» перед числом означает, что это число записано в шестнадцатеричной системе счисления.

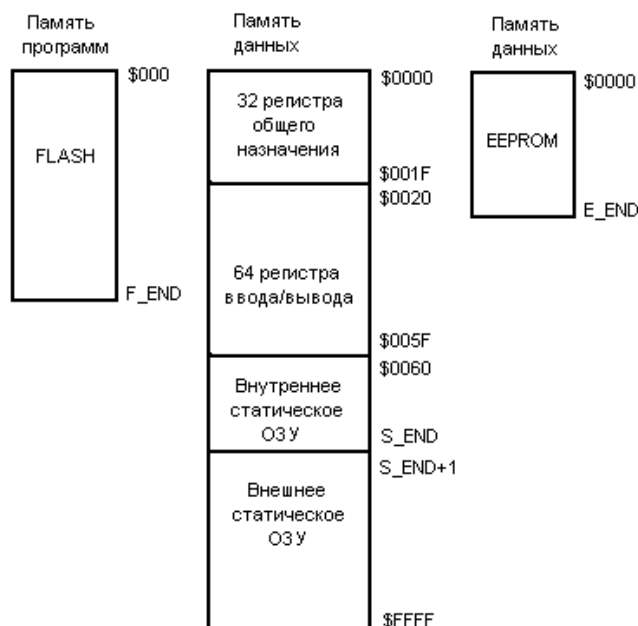


Рисунок 3.5 – Обобщенная карта памяти микроконтроллеров AVR

Память программ предназначена для хранения команд, управляющих функционированием микроконтроллера. В памяти программ хранятся также различные константы, не меняющиеся во время работы программы. Как было отмечено, память программ представляет собой электрически стираемое ППЗУ (FLASH-ПЗУ). Поскольку все команды занимают в памяти по 16 бит, память программ имеет 16-разрядную организацию. Соответственно, объем памяти микроконтроллеров различных семейств составляет от 512 до 128 К 16-битных слов.

Для адресации памяти программ используется счетчик команд (PC – Program Counter). Размер счетчика команд может достигать 16 разрядов в зависимости от объема адресуемой памяти.

Некоторые адреса памяти программ зарезервированы, т. е. используются для неких специальных целей. И первым зарезервированным адресом можно считать нулевой адрес. По адресу \$0000 памяти программ находится вектор сброса. После инициализации (сброса) микроконтроллера выполнение программы начинается с этого адреса (по этому адресу рекомендуется размещать команду относительного перехода к инициализационной части программы). Остальные зарезервированные адреса – это векторы прерываний. Так как любой микроконтроллер AVR имеет несколько источников прерывания, то и векторов прерывания тоже несколько: по одному на каждый вид прерывания. Адреса векторов прерываний находятся сразу за вектором сброса. То есть занимают ячейки с адресами \$001, \$002 и т. д. Количество векторов прерываний для разных микросхем может отличаться. При этом для контроллеров семейства Mega положение таблицы векторов прерываний можно изменить.

При возникновении прерывания после сохранения в стеке текущего значения счетчика команд происходит выполнение команды, расположенной по адресу соответствующего вектора. Поэтому по этим адресам располагаются команды относительного перехода к подпрограммам обработки прерываний.

Если в программе прерывания не используются (запрещены), то основная программа может начинаться непосредственно с адреса \$001.

Следует отметить, что память программ может использоваться не только для хранения кода программы, но также и для хранения различных констант. Данные записываются в виде восьмиразрядных двоичных чисел (байтов). При этом каждая шестнадцатиразрядная ячейка памяти программ используется как две восьмиразрядные. При чтении данных из программной памяти микроконтроллер может обратиться к каждому байту отдельно. Для пересылки байта из памяти программ в память данных имеются специальные команды – LPM и ELPM. FLASH-ПЗУ, используемое в микроконтроллерах AVR, рассчитано как минимум на 1000 циклов стирания/записи.

Память данных микроконтроллеров разделена на три части (см. рисунок 3.5): регистровая память, оперативная память (в некоторых моделях семейства Tiny отсутствует) и энергонезависимое ЭСППЗУ (EEPROM).

Регистровая память включает 32 регистра общего назначения (РОН), объединенных в так называемый «файл», и служебные регистры ввода/вывода (РВВ). Размер регистровой памяти фиксирован и для всех моделей составляет 96 байт, соответственно, под РОН отводится 32 байта, а под РВВ – 64 байта.

Все регистры общего назначения объединены в файл, структура которого показана на рисунке 3.6, и занимают в памяти данных адреса с \$0000 по \$001F.

R0
R1
R2
...
R25
R26 (регистр X, младший байт)
R27 (регистр X, старший байт)
R28 (регистр Y, младший байт)
R29 (регистр Y, старший байт)
R30 (регистр Z, младший байт)
R31 (регистр Z, старший байт)

Рисунок 3.6 – Структура файла регистров общего назначения

Все ячейки этой области памяти одновременно являются регистрами общего назначения, т. е. записывая байт данных в ячейку памяти с адресом \$0000, информация помещается в регистр R0. Как уже было отмечено, в микроконтроллерах AVR все 32 РОН непосредственно доступны АЛУ в отличие от

микроконтроллеров других фирм, в которых имеется только один такой регистр – рабочий регистр-аккумулятор. Благодаря этому любой РОН может использоваться во всех командах и как операнд-источник, и как операнд-приемник. Исключение составляют лишь пять арифметических и логических команд, выполняющих действия между константой и регистром (SBCI, SUBI, CPI, ANDI, ORI), а также команда загрузки константы в регистр (LDI). Эти команды могут обращаться только ко второй половине регистров (R16...R31).

Регистры общего назначения R26...R31 формируют три 16-разрядных индексных регистра X, Y, Z, которые используются в качестве указателя при косвенной адресации памяти программ и памяти данных.

Регистры ввода/вывода располагаются в так называемом пространстве ввода/вывода размером 64 байта, начиная с адреса \$0020 по \$005F. Все PVB можно разделить на две группы: служебные регистры микроконтроллера (регистр управления микроконтроллером, регистр состояния и т. п.) и регистр управления периферийными устройствами (в том числе порты ввода/вывода). Размер каждого регистра – 8 бит. Распределение адресов пространства ввода/вывода зависит от конкретной модели микроконтроллера, т. к. разные модели имеют различный состав периферийных устройств и, соответственно, разное количество регистров. Общее количество PVB зависит от конкретной модели микроконтроллера, их количество и размещение в адресном пространстве ввода/вывода можно найти в документации.

Область памяти, начиная с адреса \$0060 до \$FFFF, – не совмещенная ни с чем область памяти – не выполняет никаких дополнительных функций и предназначена исключительно для оперативного хранения данных. Эту область, в свою очередь, можно разделить на область внутреннего ОЗУ (\$0060... S_END) и область внешнего ОЗУ (S_END+1...\$FFFF). Под S_END понимается адрес последней ячейки внутреннего ОЗУ конкретного микроконтроллера. Некоторые микроконтроллеры Tiny AVR имеют лишь встроенное ОЗУ. Однако в состав серии входят микросхемы, допускающие подключение внешних микросхем ОЗУ. В результате объем ОЗУ микроконтроллера может быть расширен до 64 Кбайт. При этом общий объем оперативной памяти может достигать значения \$FFFF.

Энергонезависимую память данных (EEPROM) удобно использовать для долговременного хранения данных, которые могут изменяться в процессе функционирования микропроцессорной системы (настроечные или калибровочные переменные, серийные номера, счетчик перезапусков устройства, время последнего включения/выключения устройства). В отличие от FLASH-памяти ячейки EEPROM можно стирать и изменять по отдельности. Объем этой области памяти зависит от конкретной модели контроллера.

EEPROM – необычная память. Поэтому к этой памяти процессор обращается не так, как к остальным видам памяти. Для центрального процессора не существует адресного пространства EEPROM. К этому виду памяти микроконтроллер обращается при помощи регистров ввода/вывода. Для микроконтроллеров с объемом EEPROM менее 256 байт таких регистров всего три:

- 1) EEAR – регистр адреса EEPROM;

- 2) EEDR – регистр данных EEPROM;
- 3) EECR – регистр управления EEPROM.

Если объем EEPROM превышает 256 байт, то вместо одного регистра адреса (EEAR) такой микроконтроллер имеет два регистра: EEARH и EEARL.

Регистр адреса EEAR (или EEARH, EEARL) доступен как для записи, так и для чтения. В этот регистр загружается адрес ячейки, к которой будет производиться обращение. Количество задействованных разрядов зависит от объема EEPROM-памяти контроллера, поэтому содержимое незадействованных старших разрядов регистра EEAR (или EEARH) игнорируется.

Регистр данных (EEDR) работает как на запись, так и на чтение. Через этот регистр в EEPROM поступает записываемый байт. Через него же процессор получает байт при чтении из EEPROM.

Регистр управления (EECR) определяет режимы работы. Именно через него подаются команды чтения и записи EEPROM.

Два важных регистра, которые существуют в любом микропроцессоре или микроконтроллере, – это счетчик команд и указатель стека.

Счетчик команд хранит адрес текущей выполняемой команды. Этот регистр не доступен для программиста – не существует команд прямой записи или чтения его содержимого. Размер счетчика команд составляет для разных микроконтроллеров AVR от 9 до 12 разрядов. Количество разрядов счетчика команд зависит от размера адресуемой программной памяти конкретного микроконтроллера. После сброса микроконтроллера в счетчик команд записывается ноль. Затем процессор переходит в режим выполнения программы. В процессе выполнения программы счетчик всегда указывает на текущую выполняемую команду. При считывании кода команды значение счетчика увеличивается на один или два (в зависимости от длины команды). При выполнении команд безусловного и условного переходов содержимое счетчика резко меняется. В него записывается новое значение адреса. При возникновении прерывания в счетчик команд загружается адрес соответствующего вектора прерывания (\$001...\$010). Если прерывания используются в программе, по этим адресам должны размещаться команды относительного перехода к подпрограммам обработки прерываний. В противном случае основная программа может начинаться непосредственно с адреса \$001.

Важной особенностью, обуславливающей высокое быстродействие микроконтроллеров AVR, является использование двухуровневого конвейера при выполнении программы. Работа этого конвейера показана на рисунке 3.7. Во время первого машинного цикла происходит выборка команды из памяти программ и ее декодирование. Во время второго цикла эта команда выполняется, а параллельно происходит выборка и декодирование второй команды и т. д. В результате фактическое время выполнения каждой команды получается равным одному машинному циклу.

Микроконтроллеры серии AVR всегда имеют в своем составе от одного до семи портов ввода/вывода. Каждый разряд такого порта подсоединен к одному из выводов (контактов) микросхемы. Порты ввода/вывода служат для обмена информацией с внешними устройствами. Порты могут быть полные и непол-

ные. Полный порт содержит 8 разрядов. В неполных портах задействованы они не все (порты в некоторых моделях Tiny имеют 6 разрядов). Каждый порт имеет свое имя. Они именуются латинскими буквами от А до G. Упрощенная схема одного разряда порта ввода/вывода приведена на рисунке 3.8.

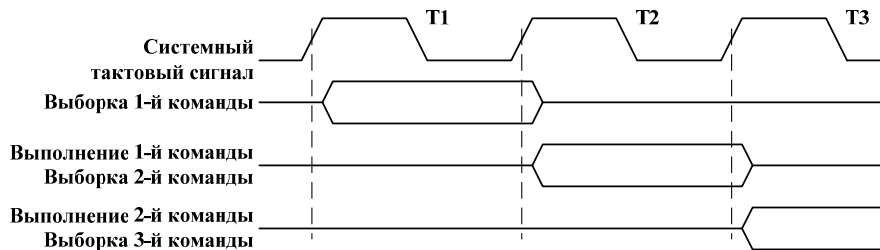


Рисунок 3.7 – Последовательность выполнения команд в конвейере

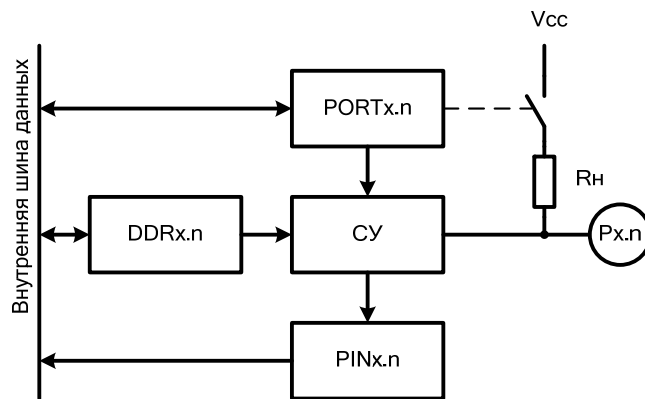


Рисунок 3.8 – Упрощенная схема порта ввода/вывода

Обращение к портам производится через регистры ввода/вывода. Под каждый порт в адресном пространстве ввода/вывода зарезервировано по три адреса. По этим адресам размещаются три регистра: регистр данных порта PORTx, регистр направления данных DDRx и регистр выводов порта PINx. Под «x» подразумевается конкретная буква – имя порта – например, PORTA, DDRA и PINA. Например, формат регистров управления контроллеров ATtiny 11x/12x приведен на рисунке 3.9.

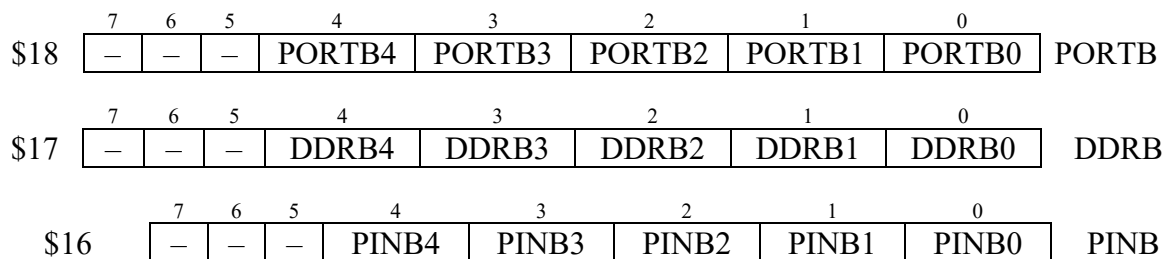


Рисунок 3.9 – Формат регистров PORTB, DDRB и PINB в моделях ATtiny 11x/12x

Любой порт ввода/вывода микроконтроллера серии AVR устроен таким образом, что каждый его разряд может работать как на ввод, так и на вывод. То есть он может быть входом, а может быть выходом. Направление передачи данных определяется содержимым регистра направления данных DDRB. Если разряд DDRB_n этого регистра установлен в «1», соответствующий n-й вывод порта является выходом, если сброшен в «0» – входом.

Для того чтобы выдать информацию на внешний вывод микросхемы, нужно в соответствующий разряд DDRB записать логическую единицу, а затем записать байт данных в регистр PORTB. Для того чтобы прочитать информацию с внешнего вывода контроллера, нужно записать в соответствующий разряд регистра DDRB «0», затем необходимо прочесть байт из регистра PINB.

Порты ввода/вывода микроконтроллеров AVR имеют еще одну полезную функцию. В режиме ввода информации они могут при необходимости подключать к каждому выводу порта внутренний нагрузочный резистор. Внутренний резистор позволяет значительно расширить возможности порта. Такой резистор создает вытекающий ток для внешних устройств, подключенных между выводом порта и общим проводом.

В состав некоторых моделей микроконтроллеров AVR входит 10-разрядный АЦП последовательного приближения. Этот АЦП имеет следующие основные параметры:

- абсолютная погрешность: ± 2 МЗР;
- интегральная нелинейность: $+0,5$ МЗР;
- быстродействие: до 15 тыс. выборок/с.

Структурная схема модуля АЦП представлена на рисунке 3.10.



Рисунок 3.10 – Структурная схема модуля АЦП

В качестве источника опорного напряжения ИОН для АЦП может использоваться внутренний или внешний источник опорного напряжения или напряжение питания микроконтроллера, выбор осуществляется мультиплексором источника опорного напряжения (ИОНМ).

Предделитель (ПД) позволяет изменять частоту тактового сигнала преобразователя.

В процессе работы АЦП может функционировать в двух режимах:

1) одиночного преобразования – запуск каждого преобразования инициируется пользователем;

2) непрерывного преобразования – запуск преобразований выполняется непрерывно через определенные интервалы времени.

Управление модулем АЦП и контроль его состояния осуществляется с помощью регистров ввода/вывода – ADMUX, ADCSR, ADCH, ADCL.

Регистр ADCSR, расположенный по адресу \$06, управляет работой АЦП. Формат этого регистра на примере контроллера ATtiny 15L приведен на рисунке 3.11.

	7	6	5	4	3	2	1	0
\$06	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Рисунок 3.11 – Формат регистра ADCSR

Биты регистра ADCSR имеют следующее назначение:

- ADEN – разрешение АЦП («1» – включено, «0» – выключено);
- ADSC – запуск преобразования («1» – начать преобразование);
- ADFR – выбор режима работы АЦП («0» – одиночное преобразование, «1» – непрерывное преобразование);
- ADIF – флаг прерывания от компаратора;
- ADIE – разрешение прерывания от компаратора;
- ADPS2...ADPS0 – выбор частоты преобразования.

Для разрешения работы АЦП необходимо установить в «1» в разряд ADEN регистра ADCSR, а для выключения сбросить его в «0». Режим работы АЦП определяется состоянием разряда ADFR. Если он установлен в «1», АЦП работает в режиме непрерывного преобразования, если сброшен в «0» – в режиме одиночного преобразования. В режиме непрерывного преобразования запуск каждого следующего преобразования осуществляется автоматически после окончания текущего, в режиме одиночного преобразования запуск каждого преобразования осуществляется по команде пользователя. Запуск каждого преобразования в режиме одиночного преобразования, а также запуск первого преобразования в режиме непрерывного преобразования осуществляется установкой в «1» разряда ADSC регистра ADCSR. После окончания процесса преобразования устанавливается флаг прерывания ADIF регистра ADCSR, одновременно с которым генерируется запрос на прерывание.

Тактовым сигналом модуля АЦП является сигнал с предделителя, на вход которого, в свою очередь, поступает тактовый сигнал микроконтроллера. Коэффициент деления предделителя и, соответственно, длительность преобразования определяются состоянием разрядов ADPS2...ADPS0 регистра ADCSR.

Так, например, при $ADPS2...APRS0 = 000$ коэффициент деления равен 2, при $ADPS2...APRS0 = 111$ – 128.

Результат преобразования сохраняется в регистре данных АЦП. Поскольку АЦП 10-разрядное, этот регистр физически размещен в двух регистрах ввода/вывода, доступных только для чтения: ADCH и ADCL. Эти регистры при включении микроконтроллера содержат значение «\$0000» и расположены по адресам \$05 и \$04. Обращение к регистрам (для получения результата преобразования) должно выполняться в определенной последовательности: сначала необходимо прочитать регистр ADCL, а затем – ADCH.

Управление входным мультиплексором модуля АЦП, а также дополнительное управление модулем АЦП осуществляется с помощью регистра ввода/вывода ADMUX, расположенного по адресу \$07. Формат этого регистра для контроллера ATtiny 15L приведен на рисунке 3.12.

	7	6	5	4	3	2	1	0
\$07	REFS1	REFS0	ADLAR	–	–	MUX2	MUX1	MUX0

Рисунок 3.12 – Формат регистра ADMUX

Разряды регистра ADMUX имеют следующее назначение:

- REFS1:REFS0 – выбор источника опорного напряжения;
- ADLAR – выравнивание результата преобразования;
- MUX2...MUX0 – выбор входного канала, определяют номер активного канала (аналоговый вход, подключенный к входу АЦП) согласно таблице 3.1.

Таблица 3.1 – Управление входным мультиплексором

MUX2... MUX0	Несимметричный вход	Дифференциальный вход		Предварительное усиление
		положительный	отрицательный	
000	ADCO (PB5)	Не применимо		
001	ADC1 (PB2)			
010	ADC2 (PB3)			
011	ADC3 (PB4)			
100	Не применимо	ADC2 (PB3)	ADC2 (PB3)	1x
101		ADC2 (PB3)	ADC2 (PB3)	20x
110		ADC2 (PB3)	ADC3 (PB4)	1x
111		ADC2 (PB3)	ADC3 (PB4)	20x

При выборе канала с дифференциальным входом разряды MUX2...MUX0 также определяют коэффициент предварительного усиления входного сигнала.

Все контроллеры AVR имеют в своем составе минимум один 8-разрядный таймер/счетчик, который может быть использован либо для отсчета и измерения временных интервалов, либо для счета внешних импульсов. Кроме того, таймеры могут использоваться для генерации сигнала с ШИМ. Каждый восьмиразрядный таймер представляет собой один восьмиразрядный регистр, кото-

рый для микроконтроллера является регистром ввода/вывода. Этот регистр хранит текущее значение таймера и называется счетным регистром. Шестнадцатиразрядные таймеры имеют шестнадцатиразрядный счетный регистр. Каждый счетный регистр имеет свое имя.

Счетный регистр восьмиразрядного таймера именуется TCNTx, где «x» – номер таймера. Для таймера T0 регистр называется TCNT0, для таймера T2 – TCNT2. Шестнадцатиразрядные регистры именуются похожим образом. Отличие в том, что каждый шестнадцатиразрядный счетный регистр для микроконтроллера представляет собой два регистра ввода/вывода. Один предназначен для хранения старших битов числа, второй – для хранения младших битов. К имени регистра старших разрядов добавляется буква H, а для регистра младших разрядов – буква L. Таким образом, счетный регистр таймера T1 – это два регистра ввода/вывода: TCNT1H и TCNT1L.

Микроконтроллер может записать в любой счетный регистр любое число в любой момент времени, а также в любой момент прочитать содержимое любого счетного регистра. Когда таймер включается в режим счета, то на его вход начинают поступать счетные импульсы. После прихода каждого такого импульса содержимое счетного регистра увеличивается на единицу. Счетными импульсами могут служить как специальные тактовые импульсы, вырабатываемые внутри самого микроконтроллера, так и внешние импульсы, поступающие на специальные входы микросхемы. При переполнении счетного регистра его содержимое обнуляется, и счет начинается сначала.

Любой таймер жестко завязан с системой прерываний. Вызвать прерывание может целый ряд событий, связанных с таймером. Например, существует прерывание по переполнению таймера, по срабатыванию специальной схемы совпадения.

Таймеры микроконтроллеров семейства AVR могут работать в нескольких режимах. Разные микроконтроллеры имеют разные наборы режимов для своих таймеров. Для выбора режимов работы существуют специальные регистры – регистры управления таймерами. Для простых таймеров используется один регистр управления, для более сложных – два регистра. Регистры управления таймером называются TCCRx (где «x» – номер таймера). Например, для таймера T0 используется один регистр с именем TCCR0. Для управления таймером T1 используется два регистра: TCCR1A и TCCR1B. При помощи регистров управления производится не только выбор соответствующего режима, но и более тонкая настройка таймера. Например, для контроллеров ATtiny 11x/12x регистр управления имеет следующий вид (рисунок 3.13).

Управляющие биты данного регистра позволяют выбрать источник сигналов, поступающих на вход таймера (таблица 3.2).

	7	6	5	4	3	2	1	0
\$33	–	–	–	–	–	CS02	CS01	CS00

Рисунок 3.13 – Формат регистра TCCR0

Таблица 3.2 – Выбор источника тактового сигнала для таймера/счетчика T0

CS02	CS01	CS00	Источник тактового сигнала
0	0	0	Таймер/счетчик остановлен
0	0	1	СК (тактовый сигнал микроконтроллера)
0	1	0	СК/8
0	1	1	СК/64
1	0	0	СК/256
1	0	1	СК/1024
1	1	0	Вывод T0, инкремент счетчика производится по спадающему фронту импульсов
1	1	1	Вывод T0, инкремент счетчика производится по нарастающему фронту импульсов

Система команд микроконтроллеров AVR весьма развита и насчитывает в различных моделях от 90 до 133 различных инструкций. Несмотря на то, что микроконтроллеры AVR являются микроконтроллерами с RISC-архитектурой, по количеству реализованных инструкций и их разнообразию они больше похожи на микроконтроллеры с CISC-архитектурой. Практически каждая из команд (за исключением команд, у которых одним из операндов является 16-разрядный адрес) занимает только одну ячейку памяти программ. Причем это достигнуто не за счет сокращения количества команд процессора, а за счет увеличения разрядности памяти программ.

Как уже было упомянуто, в микроконтроллерах семейства Tiny и Mega память программ является 16-разрядной. Соответственно, большинство команд описываются 16-разрядным словом – кодом операции. Некоторые команды, у которых один из операндов является 16-разрядным адресом, занимают две ячейки памяти программ. Соответственно, код операции таких команд является 4-байтным числом. В ряде случаев значение операнда-источника может содержаться непосредственно в коде операции, а не в регистре. Это происходит в том случае, когда операндом-источником является константа.

По функциональному назначению все множество команд микроконтроллеров AVR можно разбить на несколько групп:

- команды логических операций;
- команды арифметических операций и команды сдвига;
- команды операций с битами;
- команды пересылки данных;
- команды передачи управления;
- команды управления системой.

Команды логических операций позволяют выполнять стандартные логические операции над байтами, такие как «логическое И», «логическое ИЛИ», операцию «исключающее И», а также вычисление обратного (дополнение до единицы) и дополнительного (дополнение до двух) кодов числа. К этой группе можно отнести также команды очистки/установки регистров и команду перестановки тетрад. Операции производятся между регистрами общего назначения

либо между регистром и константой; результат сохраняется в РОН. Все команды из этой группы выполняются за один машинный цикл.

AND Rd, Rr – «логическое И» двух РОН, $Rd \leftarrow Rd \text{ AND } Rr$.

NEG Rd – вычисляет дополнительный код числа, находящегося в регистре Rd, $Rd \leftarrow \$00 - Rd$.

Команды арифметических операций и команды сдвига позволяют выполнять такие базовые операции, как сложение, вычитание, сдвиг (вправо и влево), инкремент и декремент. В микроконтроллерах семейства Mega также имеются команды, позволяющие осуществлять умножение 8-разрядных значений. Все операции производятся только над регистрами общего назначения. При этом микроконтроллеры AVR позволяют легко оперировать как знаковыми, так и беззнаковыми числами, а также работать с числами, представленными в дополнительном коде.

Почти все команды рассматриваемой группы выполняются за один машинный цикл. Команды умножения и команды, оперирующие двухбайтовыми значениями, выполняются за два цикла.

ADD Rd, Rr – складывает содержимое двух РОН Rd и Rr, $Rd \leftarrow Rd + Rr$.

FMULS Rd, Rr – осуществляет умножение дробных чисел, хранящихся в регистрах Rd (число со знаком) и Rr (число без знака), результат перемножения сдвигается на один разряд влево, $R1:R0 \leftarrow (Rd \times Rr) \ll 1$.

Команды операций с битами позволяют выполнять установку или сброс заданного разряда РОН или ПВВ. Причем для изменения разрядов регистра состояния SREG имеются также дополнительные команды (точнее говоря, эквивалентные мнемонические обозначения общих команд), т. к. проверка состояния разрядов именно этого регистра производится чаще всего. Условно к этой группе можно отнести также две команды передачи управления типа «проверка/пропуск», которые пропускают следующую команду в зависимости от состояния разряда РОН или ПВВ.

Всем командам данной группы требуется один машинный цикл для выполнения, за исключением случаев, когда в результате проверки происходит пропуск команды. В этом случае команда выполняется за два или три машинных цикла в зависимости от пропускаемой команды.

CBR Rd, K – сбрасывает разряды в регистре Rd в соответствии с маской, заданной константой K, $Rd \leftarrow Rd \text{ AND } (\$FF - K)$.

SBI A, b – устанавливает разряд b в регистре ввода/вывода, расположенного по адресу A пространства ввода/вывода, $I/O(A).b = 1$.

Команды пересылки данных предназначены для пересылки содержимого ячеек, находящихся в адресном пространстве памяти данных. Разделение адресного пространства на три части (РОН, ПВВ, ОЗУ) предопределило разнообразие команд данной группы. Пересылка данных, выполняемая командами группы, может производиться в следующих направлениях:

- РОН \leftrightarrow РОН;
- РОН \leftrightarrow ПВВ;
- РОН \leftrightarrow память данных.

На выполнение команд данной группы требуется в зависимости от команды от одного до трех машинных циклов.

MOV Rd, Rr – копирует содержимое регистра Rr в регистр Rd, $Rd \leftarrow Rr$.

ST Y, Rr – сохраняет содержимое РОН Rr в памяти данных по адресу, содержащемуся в индексном регистре Y, $(Y) \leftarrow Rr$.

В группу команды передачи управления входят команды перехода, вызова подпрограмм и возврата из них и команды типа «проверка/пропуск», пропускающие следующую за ними команду при выполнении некоторого условия. Также к этой группе относятся команды сравнения, формирующие флаги регистра SREG и предназначенные, как правило, для работы совместно с командами условного перехода.

Команды передачи управления нарушают нормальное (линейное) выполнение основной программы. Каждый раз, когда выполняется команда из этой группы (кроме команд сравнения), нормальное функционирование конвейера нарушается. Перед загрузкой в конвейер нового адреса производится остановка и очистка выполняемой последовательности команд. Соответственно, реинициализация конвейера приводит к необходимости использования нескольких машинных циклов для выполнения таких команд.

JMP k – абсолютный безусловный переход, выполняет переход по адресу, равному сумме содержимого счетчика команд и константы k, $PC \leftarrow k$.

BRTS k – условный относительный переход, проверяет флаг T и выполняет переход, если этот разряд установлен.

В группу команды управления системой входят всего 3 команды:

- 1) NOP – пустая команда;
- 2) SLEEP – перевод микроконтроллера в режим пониженного энергопотребления;
- 3) WDR – сброс сторожевого таймера.

Все команды этой группы выполняются за один машинный цикл.

4 Рекомендации по выполнению и оформлению курсовой работы

4.1 Общие методические указания

В курсовой работе рассматриваются вопросы, связанные с разработкой микропроцессорных устройств управления электроприводом. Проектирование этих устройств требует решения большого комплекса технических задач и обычно выполняется по общей методике, изложенной в разделе 2 данных методических указаний.

Выполненная работа должна соответствовать варианту, указанному преподавателем. К выполнению курсовой работы следует приступить только после изучения методических указаний к курсовому проектированию и теоретического материала учебной литературы.

Рекомендуемая последовательность выполнения курсовой работы:

- после получения задания и определения исходных данных произвести подбор технической литературы;
- изучить принцип работы объекта, выделить управляющие и информационные сигналы и определить последовательность их выдачи;
- определить особенности построения систем с заданным объектом, составить функциональную схему системы;
- разработать общий алгоритм функционирования микропроцессорной системы, определить режим работы, последовательность решения задач;
- изучить особенности архитектуры заданного семейства микроконтроллеров, при этом особое внимание обратить на структуру микроконтроллера, состав и организацию памяти, назначение управляющих сигналов, состав и порядок настройки встроенных периферийных устройств;
- произвести выбор микроконтроллера и других технических средств, необходимых для реализации аппаратной части устройства управления, разработать структурную и принципиальную схемы микропроцессорной системы;
- составить алгоритм управления и написать программу на ассемблере, реализующую предложенный алгоритм;
- оформить курсовую работу в соответствии с требованиями ЕСКД.

Успешное выполнение курсовой работы возможно лишь при творческом подходе к процессу проектирования и последовательном решении.

4.2 Содержание курсовой работы

Курсовая работа состоит из графической части и пояснительной записки.

Графическую часть курсовой работы рекомендуется выполнять на двух листах формата А1 или А2. На первом листе изображается схема электрическая принципиальная разрабатываемого устройства, на втором – функциональная и структурная схемы устройства и алгоритмы работы.

Содержание графической части.

Лист 1. Система микропроцессорная. Схема электрическая принципиальная.

Система микропроцессорная. Перечень элементов.

Лист 2. Система микропроцессорная. Документы прочие.

В пояснительной записке приводятся описание элементов устройства управления, необходимые расчеты и пояснения по всем этапам проектирования, выводы по результатам проектирования.

Основная текстовая часть пояснительной записки состоит из введения, трех разделов и заключения.

Введение является важной частью работы, в которой должны быть отражены актуальность проблемы, решаемой в работе, современное состояние вопроса, цели и задачи курсовой работы.

В первом разделе описываются назначение, принцип работы агрегата или системы, которые заданы в качестве объектов управления в разрабатываемой системе, приводятся временные диаграммы, поясняющие принцип работы, регулировочные характеристики, типовые схемы подключения. В результате

должны быть разработаны функциональная и структурная схемы системы. Функциональная схема представляет собой совокупность функциональных блоков и информационных связей между ними. При описании функциональной схемы необходимо указать количество, назначение и вид (аналоговой/дискретный) всех сигналов управления, формируемых системой управления. Графическое построение схемы должно наглядно отражать последовательность процессов обработки информации при работе системы. Действительный способ реализации функциональных блоков может не учитываться. На структурной схеме изображаются аппаратные блоки, из которых состоит система-микропроцессор, память, интерфейсы, индикация, клавиатура, гальваническая развязка. Функциональная и структурная схемы системы выполняются с учетом всех дополнительных требований, указанных в задании.

Во втором разделе описываются элементы, используемые в схеме. При описании микроконтроллера необходимо указать архитектурные особенности семейства, состав и назначение микросхем, привести структурную схему микроконтроллера, описать назначение элементов структурной схемы и управляющих сигналов. Для всех микросхем, используемых в устройстве, приводятся структурные схемы, описывается назначение управляющих сигналов и способы программирования. При необходимости приводится расчет параметров резисторов и транзисторов. В этом же разделе необходимо описать схему электрическую принципиальную устройства, которая должна соответствовать структурной схеме, привести карту распределения адресов ячеек памяти и портов ввода-вывода.

Третий раздел посвящен разработке программного обеспечения системы. В этом разделе необходимо описать состав, структуру и формат команд микроконтроллера, способы адресации и структуры данных, привести примеры команд каждой группы и указать особенности их выполнения. Общий алгоритм управления разрабатывается в виде блок-схемы и представляет собой описание последовательности решения микроконтроллером функциональных задач. Рекомендуемое общее количество блоков в алгоритме – 15–20. При организации в устройстве прерываний программа обслуживания прерывания представляется в виде отдельного алгоритма. Алгоритм решения заданной функциональной задачи разрабатывается с учетом особенностей системы команд микроконтроллера. Для написания программы используется язык ассемблер или СИ, текст программы обязательно должен содержать комментарии.

Заключение пояснительной записки должно содержать оценку результатов проектирования.

Рекомендуемое содержание пояснительной записки.

Титульный лист.

Задание на проектирование.

Содержание.

Введение.

1 Разработка функциональной схемы системы.

1.1 Описание принципа работы.

1.2 Функциональная схема системы управления.

- 2 Разработка аппаратной части системы.
 - 2.1 Краткое описание микроконтроллера.
 - 2.2 Описание подсистемы памяти.
 - 2.3 Расчет и выбор элементов сопряжения.
 - 2.4 Описание схемы электрической принципиальной.
 - 3 Разработка программного обеспечения системы.
 - 3.1 Краткое описание системы команд микроконтроллера.
 - 3.2 Описание общего алгоритма работы системы.
 - 3.3 Описание текста программы.
- Заключение.
- Список используемых источников.
- Приложение А. Текст программы.

4.3 Оформление курсовой работы

Графическая часть курсовой работы должна быть оформлена в соответствии с требованиями действующих стандартов. Чертежи и схемы выполняются на белой бумаге формата А1 или А2 ручным или машинным способом. Основная надпись (угловой штамп) заполняется в соответствии с ГОСТ 2.104–68, графические обозначения элементов на схеме электрической принципиальной должны соответствовать ГОСТ 2.700–2.770.

Пояснительная записка к курсовой работе выполняется на листах формата А4. Содержание разделов должно быть кратким, но полностью раскрывающим суть вопроса. По тексту необходимо указывать источник информации, делая ссылку на перечень литературы, приводимой в конце работы. При необходимости текст может сопровождаться рисунками, схемами и диаграммами. Примерный объем пояснительной записки – 25–30 страниц.

Курсовая работа должна быть аккуратно оформлена в соответствии с общими требованиями. Текстовая часть работы должна быть написана четким почерком, сокращение слов не допускается. Каждый раздел должен начинаться с новой страницы. На листах должны быть оставлены поля для замечаний.

Количество рисунков в курсовой работе должно быть достаточным для пояснения излагаемого текста. Рисунки выполняются карандашом и должны иметь номер и название, которое помещается под рисунком. Рисунки располагают по возможности ближе к соответствующим частям текста и нумеруют арабскими цифрами в пределах ответа на вопрос. Номер рисунка состоит из номера вопроса и порядкового номера, разделенных точкой, например, Рисунок 1.1. Рисунки при необходимости могут иметь наименования и пояснительные данные (подрисуночный текст). Слово «Рисунок» и наименование помещают после пояснительных данных и записывают следующим образом, например, Рисунок 1.3 – Схема электропривода.

Таблицы применяются для лучшей наглядности и удобства сравнения показателей. Размеры таблицы выбираются произвольно в зависимости от изложения материала. Высота строк таблицы должна быть не менее 8 мм. Таблицу помещают под текстом, в котором дана ссылка на нее. Таблицы нумеруются в

пределах раздела. Номер таблицы состоит из номера вопроса и порядкового номера таблицы, разделенных точкой. Слово «Таблица», ее номер помещаются над таблицей слева. Таблицы могут иметь названия, которые должны отражать их содержание, быть точными и краткими. Название следует помещать над таблицей, после номера таблицы, например, Таблица 3.1 – Технические параметры электродвигателей.

Таблицы слева, справа и снизу, как правило, ограничивают линиями. Если строки таблицы выходят за формат страницы, ее переносят на следующую страницу, при этом над перенесенной частью таблицы пишут слова «Продолжение таблицы» с указанием номера таблицы и повторяют головку таблицы.

В конце курсовой работы следует поместить список использованной литературы. Список использованных источников составляется в порядке следования ссылок на них в тексте. Запись источника должна быть заполнена в следующей последовательности:

- для книг – фамилии и инициалы авторов, наименование, издательство, год издания, страницы;
- для статей – фамилии и инициалы авторов, заголовки статьи, название журнала (или сборника), год издания, том, номер страницы.

Ссылки на источник в курсовой работе выполняются без указания страниц, при этом номер источника берется в квадратные скобки, например, [2].

После проверки курсовой работы, если потребуется переработка каких-либо разделов, исправленный ответ вносится на обратную сторону страницы пояснительной записки. Замечания преподавателя не разрешается ни стирать, ни заклеивать.

К защите студент допускается после устранения всех замечаний преподавателя.

5 Пример микропроцессорной системы

5.1 Пример на базе микроконтроллера AVR

Процесс проектирования микропроцессорной системы начинается с выбора типа микроконтроллера, который будет использоваться для решения поставленных задач. Для выбора типа контроллера необходимо учитывать множество факторов, основными из которых являются:

- быстродействие центрального процессора;
- разрядность обрабатываемых данных;
- наличие и объем встроенной памяти данных и программ;
- наличие и количество периферийных устройств (портов ввода/вывода, АЦП, таймеры и т. д.).

Быстродействие контроллеров AVR достигает 1MIPS/1МГц, что достаточно для решения широкого круга задач. Поэтому при выборе конкретной модели

контроллера следует выяснить набор функций, которые должна выполнять микропроцессорная система.

Например, пусть необходимо разработать МПС измерения среднего значения тока. Для построения такой системы необходимо организовать ввод информации с аналогового датчика, например шунта, соответственно, желательно наличие в составе периферийных узлов контроллера аналогово-цифрового преобразователя. Кроме того, системе требуется устройство для индикации измеренного значения. Для этого необходимо предусмотреть возможность подключения устройства индикации, например, на базе матричного жидкокристаллического индикатора (МЖКИ). Современные МЖКИ обладают возможностью подключения по 4-проводному интерфейсу, для организации которого потребуется семь цифровых линий. При необходимости взаимодействия системы с другими устройствами требуется предусмотреть возможность организации одного из типовых интерфейсов обмена информацией. Так, например, для организации связи посредством SPI-интерфейса потребуется наличие в составе контроллера соответствующего периферийного узла и четырех цифровых линий.

Функциональная схема системы, выполняющей такие функции, может иметь следующий вид (рисунок 5.1).

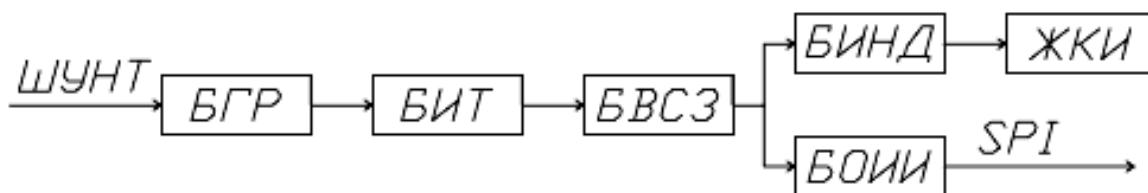


Рисунок 5.1 – Функциональная схема МПС измерения тока

В данной схеме можно выделить следующие блоки:

- блок обмена информацией по интерфейсу (БОИИ) – выдает информацию через интерфейс SPI;
- блок индикации (БИНД) – служит для управления ЖКИ;
- жидкокристаллический индикатор (ЖКИ) – обеспечивает отображение измеряемого значения;
- БВСЗ – блок вычисления среднего значения;
- БИТ – блок измерения тока;
- БГР – блок гальванической развязки.

Требованиям, выдвинутым к микропроцессорному устройству, удовлетворяет микроконтроллер ATtiny 261. К основным отличительным особенностям микроконтроллера (по сравнению с другими контроллерами AVR) можно отнести:

- эффективный набор инструкций: 123 инструкции, большинство из которых выполняются за один машинный цикл;
- наличие памяти программ и данных: 2 кбайт внутрисистемно-программируемой флэш-памяти программ, характеризующейся износостойкостью 10 тыс. циклов чтения/записи; 128 байт внутрисистемно-программируемого ЭСППЗУ, характеризующегося износостойкостью 100000 циклов записи/стирания; 128 байт встроенного статического ОЗУ;

- 8/16-разрядный таймер-счетчик с предделителем и двумя каналами ШИМ;
- 8/10-разрядный высокоскоростной таймер-счетчик с отдельным предделителем;
- три высокочастотных ШИМ-выхода с отдельным генератором паузы неперекрывания;
- универсальный последовательный интерфейс с детектором условия старта;
- 10-разрядный АЦП;
- 16 программируемых линий ввода-вывода;
- рабочее напряжение: – 1,8...5,5 В;
- малая потребляемая мощность;
- активный режим: 1 МГц; 1,8 В; 380 мкА;
- режим снижения потребляемой мощности: 0,1 мкА при 1,8 В.

Для ввода аналогового сигнала с шунтов в микропроцессорную систему, как правило, необходимо предусмотреть возможность его гальванической развязки с сигналами микропроцессорной связи, что позволит повысить надежность системы. Для решения этой задачи можно использовать изолирующие усилители, одним из которых является микросхема HCPL-7840 производства фирмы AvagoTech. Данная микросхема обладает линейной характеристикой при напряжениях –200...+200 мВ. Выходной сигнал дифференциальный.

Основные характеристики микросхемы:

- нелинейность – 0,004 %;
- коэффициент усиления по напряжению – 8;
- напряжение питания 5 В – однополярное;
- полоса пропускания (тип.) – 0,1 МГц;
- напряжение изоляции – 3750 В;
- корпус DIP-8.

В качестве устройства индикации можно выбрать двустрочный дисплей RT162-7 со встроенным контроллером HD44780 (Hitachi). Интерфейс дисплея содержит восемь входов для передачи команд и данных в контроллер дисплея и три линии управления. При этом буфер ввода может работать в двух режимах: 8-битном и 4-битном. При работе в 4-битном режиме двоичное 8-разрядное слово разбивается на две тетрады, которые поочередно заносятся в буфер. Соответственно, для сопряжения с микроконтроллером может потребоваться всего семь линий.

Пример микропроцессорной системы, построенной в соответствии с указанными техническими требованиями, приведен на рисунке 5.2.

Центральным узлом системы является микросхема микроконтроллера ATtiny261 (DD1), к порту А которой подключен ЖК-индикатор HG1. Разряды PA4...PA7 порта используются для передачи данных в индикатор, разряды PA0 и PA2 – для выдачи сигналов управления RS и E соответственно.

Для ввода аналогового сигнала в микроконтроллер используется шунт RS1, параллельно которому включен помехоподавляющий конденсатор. Выводы шунта подключены ко входам микросхемы изолирующего усилителя DA1, дифференциальный выход которой (выводы V_{o+} и V_{o-}) подключен

к входу АЦП. Интерфейс SPI организуется аппаратно, выведен на штыревое соединение.

Переменный резистор RP1 необходим для регулировки контраста дисплея.

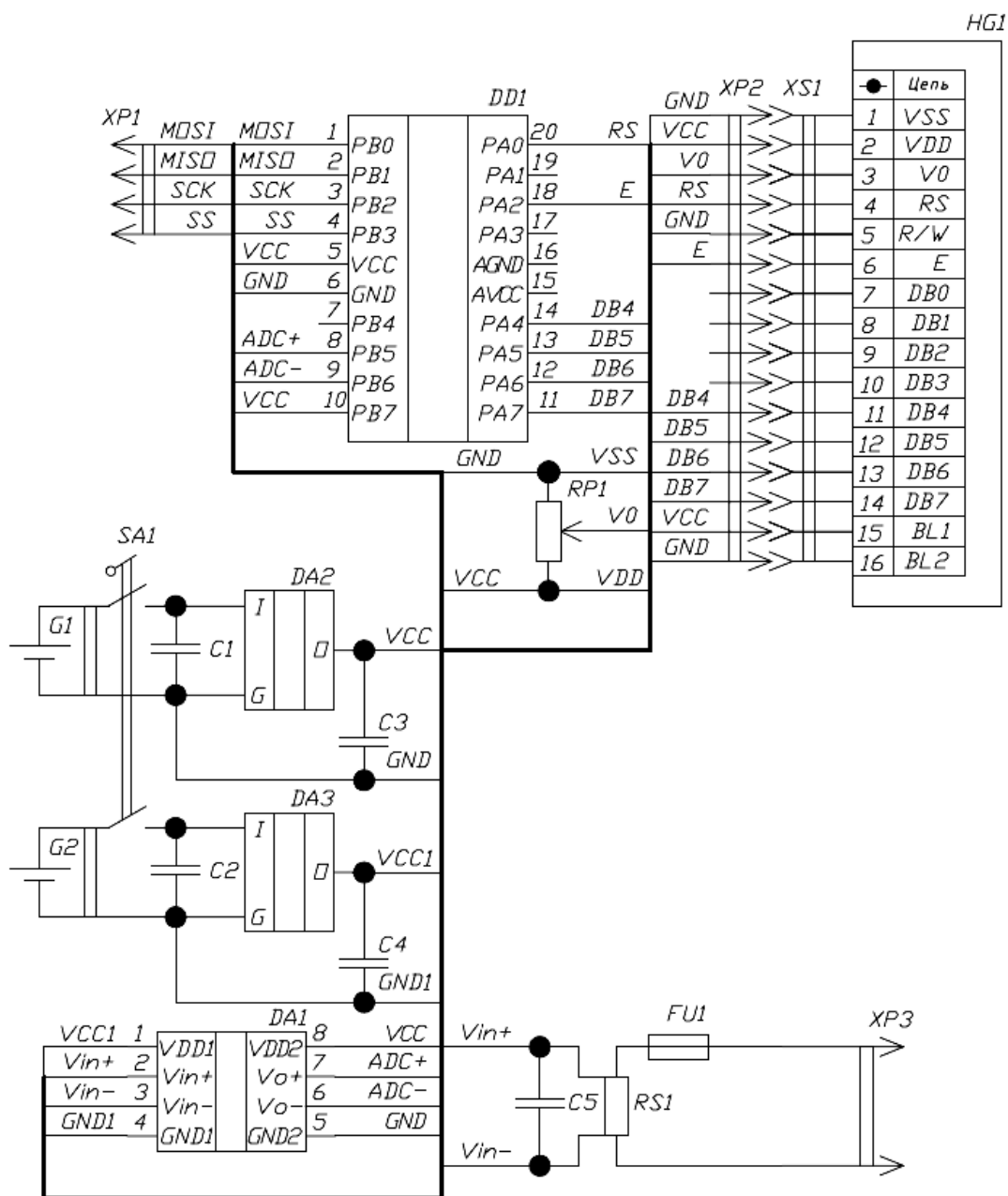


Рисунок 5.2 – МПС измерения значения тока

5.2 Пример на базе микроконтроллера семейства MCS-51

Рассмотрим пример построения функциональной и структурной схем микропроцессорной системы управления электроприводом механизма подъема крана.

С помощью крана могут быть осуществлены операции подъема и опускания груза, а также перемещение его вдоль и поперек рабочего пролета. Управление движением и подъемом, как правило, осуществляется крановщиком (оператором) из кабины.

В большинстве случаев каждый механизм имеет индивидуальный привод и индивидуальную схему управления этим приводом. Питание схем управления отдельными приводами осуществляют от главных троллеев через токосъемники и защитную панель.

В зависимости от назначения и режима работы для привода кранов применяют электродвигатели переменного тока единой серии с короткозамкнутым ротором, специальные краново-металлургические двигатели, а также электродвигатели постоянного тока. Для управления механизмом подъема тяжелых кранов обычно используют магнитные контроллеры. Контроллеры, применяемые для управления электродвигателями, могут быть выполнены с различным числом ступеней и, в свою очередь, могут быть барабанными, кулачковыми или магнитными. Управление двигателем при этом осуществляется параметрическим способом, например, путем введения/выведения реостатов в цепь асинхронного двигателя с фазным ротором (АДФР).

Применение частотного преобразователя в таком приводе позволяет улучшить энергетические показатели привода и отказаться от сложных в эксплуатации АДФР. Для управления приводом вместо релейно-контакторной системы управления можно использовать микропроцессорную систему.

На механизмах подъема и передвижения устанавливаются ограничители хода, воздействующие на электрическую цепь управления. Конечные выключатели механизма подъема ограничивают ход захватывающего устройства вверх, движение вниз не ограничивается.

К конструкции как механической, так и электрической частей правилами «Устройства и безопасной эксплуатации грузоподъемных кранов» предъявляются требования, направленные на обеспечение безопасности.

Первым из них является обязательное наличие в конструкции механической части тормоза так называемого «нормально замкнутого типа», т. е. при отключенном приводе механизм должен быть всегда заторможен. Тормоз должен обладать достаточным запасом тормозного момента, называемым «коэффициент запаса торможения».

Вторым требованием является обязательное обеспечение опускания груза только работающим электродвигателем.

Указанные требования определяют схемные решения и построение силового канала электропривода механизма подъема с преобразователем частоты:

- электропривод должен обеспечивать достаточный запас пускового момента при максимальной допустимой нагрузке и максимальном допустимом снижении питающего напряжения;
- растормаживание механического тормоза должно производиться только после того, как электропривод разовьет пусковой момент, достаточный для подъема груза;
- срабатывание любой защиты электропривода должно приводить к отключению электропривода с затормаживанием механическим тормозом;
- преобразователь частоты должен быть оборудован устройством рекуперации или рассеяния энергии торможения (тормозной резистор) с мощностью,

достаточной для обеспечения опускания груза максимальной допустимой массы на номинальной скорости.

Исходя из приведенных требований функциональная схема системы управления электроприводом механизма подъема может выглядеть следующим образом (рисунок 5.3).

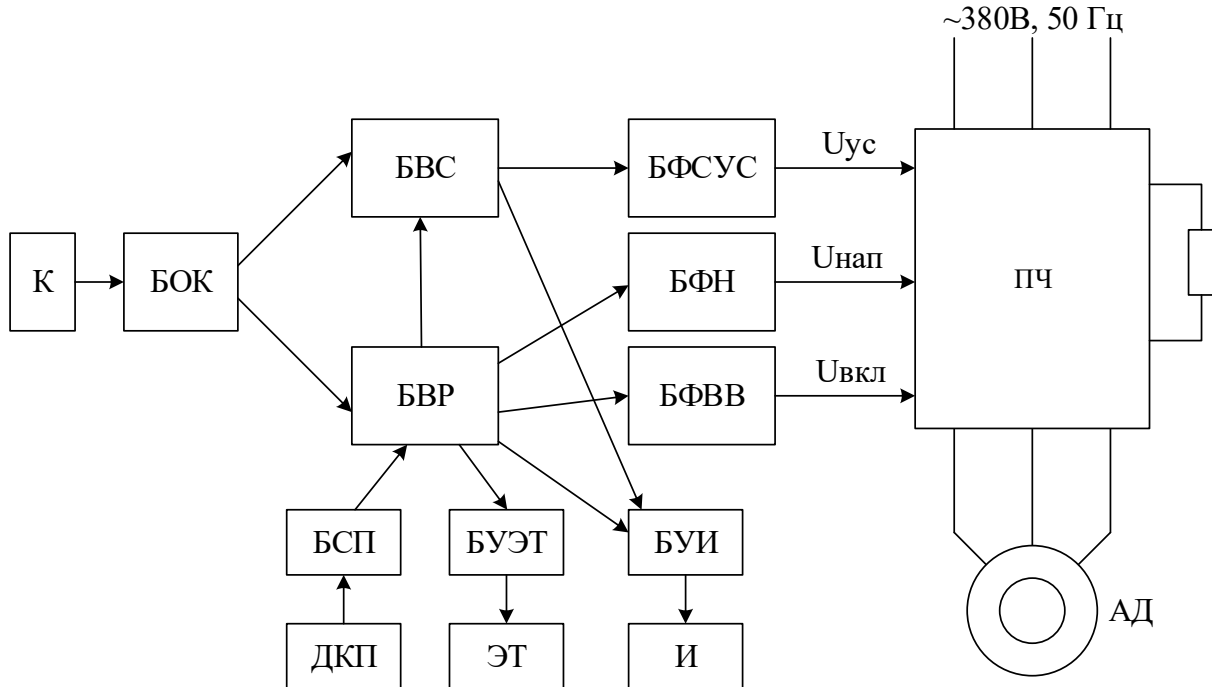


Рисунок 5.3 – Функциональная схема МПС ЭП механизма подъема крана

На приведенной схеме блоки имеют следующее назначение:

- ПЧ – преобразователь частоты, вырабатывающий напряжение для асинхронного двигателя АД, приводящего механизм в движение;
- К – клавиатура – используется для организации связи с оператором, обеспечивает ввод команд;
- БОК – блок опроса клавиатуры – для ввода информации с клавиатуры, используется для задания скорости вращения и выбора режима работы привода;
- ДКП – датчик крайнего положения – позволяет контролировать положение механизма подъема;
- БСП – блок сравнения положения – при достижении сигнала от ДКП заданного значения, вырабатывает сигнал о необходимости останова;
- БВР – в зависимости от состояния кнопок выбирает режим работы (спуск, подъем, разгон, торможение);
- БВС – в зависимости от режима работы выбирает скорость, на которой должен работать ЭП;
- БФСУС – формирует аналоговый сигнал управления скоростью ЭП $U_{ус}$ и подает его на соответствующий вход ПЧ;
- БФН – формирует дискретный сигнал $U_{нап}$ на соответствующий вход ПЧ для выбора направления;

– БФВВ – формирует дискретные сигналы включения/выключения Увкл ЭП и подает их на соответствующие входы ПЧ;

– ЭТ и БУЭТ – электромагнитный тормоз и блок управления ЭТ, обеспечивают фиксацию механизма для предотвращения движения механизмов под действием сил инерции или веса;

– И и БУИ – индикатор и блок его управления – используются для вывода оператору необходимой информации о состоянии ЭП.

Структурная схема микропроцессорного блока управления на базе микроконтроллера семейства MCS-51, выполняющего функции в соответствии с данной функциональной схемой, может выглядеть, как на рисунке 5.4.

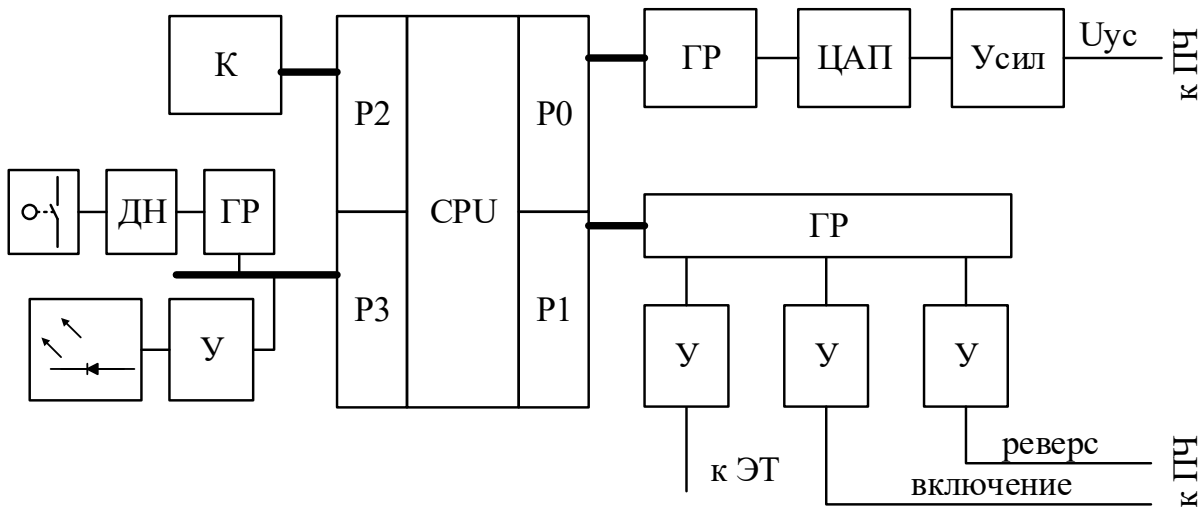


Рисунок 5.4 – Структурная схема микропроцессорного блока

Как видно, входные и выходные сигналы подключаются к выводам портов микроконтроллера. Клавиатура К подключена к порту P2. Выходной сигнал управления скоростью привода вырабатывается через порт P0 с использованием элементов гальванической развязки ГР, ЦАП и усилителя. Выходные сигналы управления включением, выбора направления ПЧ и управления тормозом вырабатываются портом P1 и формируются с использованием элементов гальванической развязки и промежуточного усилителя (например, на базе транзистора). Сигнал с концевого выключателя вводится через одну из линий порта P3 через делитель напряжения ДН и гальванический изолятор. Светодиодный индикатор подключается через усилитель (например, на базе транзисторов) к порту P3.

Список литературы

1 Уилмсхерст, Т. Разработка встроенных систем с помощью микроконтроллеров PIC. Принципы и практические примеры: пер. с англ. / Т. Уилмсхерст. – Киев: МК-Пресс, 2008. – 544 с.

2 Каган, Б. М. Основы проектирования микропроцессорных устройств автоматики / Б. М. Каган, В. В. Сташин. – Москва: Энергоатомиздат, 1987. – 304 с.

- 3 **Хвощ, С. Т.** Микропроцессоры микроЭВМ в системах автоматического управления: справочник / С. Т. Хвощ, Н. Н. Варлинский, Е. А. Попов; под ред. С. Т. Хвоща. – Ленинград: Машиностроение, 1987. – 640 с.
- 4 **Евстифеев, А. В.** Микроконтроллеры AVR семейства Tiny и Mega фирмы ATMEL / А. В. Евстифеев. – 3-е изд. – Москва: Додэка-XXI, 2006. – 288 с.
- 5 **Евстифеев, А. В.** Микроконтроллеры AVR семейства Classic фирмы ATMEL-M / А. В. Евстифеев. – Москва: Додэка-XXI, 2004. – 560 с.
- 6 Микропроцессоры и однокристальные микроЭВМ. Номенклатура и функциональные возможности / А. С. Басманов [и др.]; под ред. В. Г. Дорачева. – Москва: Энергоатомиздат, 1988. – 128 с.
- 7 **Файнштейн, В. Г.** Микропроцессорные системы управления тиристорными электроприводами / В. Г. Файнштейн, Э. Г. Файнштейн. – Москва: Энергоатомиздат, 1986. – 240 с.
- 8 **Сташин, В. В.** Проектирование цифровых устройств на однокристальных микроконтроллерах / В. В. Сташин, А. В. Урусов, О. Ф. Мологонцева. – Москва: Энергоатомиздат, 1990. – 224 с.
- 9 **Ломако, М. В.** Микропроцессорное управление приводами промышленных роботов / М. В. Ломако, И. В. Медведев. – Москва: Машиностроение, 1990. – 96 с.
- 10 Микропроцессорное управление электроприводами станков с ЧПУ / Э. Д. Тихомиров [и др.]. – Москва: Машиностроение, 1990. – 320 с.
- 11 **Голубцов, М. С.** Микроконтроллеры AVR: от простого к сложному / М. С. Голубцов. – Москва: Солон-Пресс, 2003. – 288 с.
- 12 Применение интегральных микросхем памяти: справочник / Под ред. А. Ю. Гордонова. – Москва: Радио и связь, 1994. – 232 с.
- 13 **Шило, В. Л.** Популярные цифровые микросхемы: справочник / В. Л. Шило. – Москва: Радио и связь, 1987. – 352 с.
- 14 Применение интегральных микросхем в электронной вычислительной технике / Под ред. Б. И. Файзулаева, В. В. Тарабрина. – Москва: Радио и связь, 1986. – 384 с.
- 15 **Тавернье, К.** PIC-микроконтроллеры. Практика применения: пер. с фр. / К. Тавернье. – Москва: ДМК Пресс, 2004. – 272 с.
- 16 **Боборыкин, А. В.** Однокристальные микроЭВМ / А. В. Боборыкин. – Москва: МИКАП, 1994. – 400 с.
- 17 **Трамперт, В.** Измерение, управление и регулирование с помощью AVR-микроконтроллеров: пер. с нем. / В. Трамперт. – Киев: М-Пресс, 2006. – 208 с.
- 18 **Гребнев В. В.** Микроконтроллеры семейства AVR фирмы Atmel / В. В. Гребнев. – Москва: ИП Радио Софт, 2002. – 176 с.
- 19 Программирование микропроцессорных систем: учебное пособие / Под ред. В. Ф. Шаньгина. – Москва: Высшая школа, 1990. – 303 с.
- 20 **Шпак, Ю. В.** Программирование на языке С для AVR- и PIC-микроконтроллеров / Ю. В. Шпак. – Киев: МК-Пресс, 2006. – 400 с.
- 21 **Белов, А. В.** Конструирование устройств на микроконтроллерах / А. В. Белов. – Санкт-Петербург: Наука и Техника, 2005. – 256 с.

22 **Предко, М.** Справочник по PIC- микроконтроллерам: пер. с англ. / М. Предко. – Москва: ДМК Пресс, 2002. – 512 с.

23 **Щелкунов, Н. Н.** Микропроцессорные средства и системы / Н. Н. Щелкунов, А. П. Дианов. – Москва: Радио и связь, 1989. – 283 с.

24 **Джозеф, Ю.** Ядро CortexM3 компании ARM. Полное руководство: пер. с англ. / Джозеф Ю. – Москва: Додэка XXI, 2012. – 552 с.: ил.

25 **Редькин, П. П.** 32-битные микроконтроллеры NXP с ядром Cortex-M3 семейства LPC17xx. Полное руководство / П. П. Редькиню – Москва: Додэка XXI, 2015. – 766 с. : ил.

26 **Шишов, О. В.** Технические средства автоматизации и управления: учебное пособие / О. В. Шишов. – Москва: ИНФРА-М, 2021. – 396 с.

27 **Жежера, Н. И.** Микропроцессорные системы автоматизации технологических процессов: учебное пособие / Н. И. Жежера. – 2-е изд. – Москва; Вологда: Инфа-Инженерия, 2020. – 240 с.: ил.

28 **Сеславин, А. И.** Теория автоматического управления. Линейные, непрерывные системы: учебник / А. И. Сеславин. – Москва: ИНФРА-М, 2021. – 314 с.

29 **Шишов, О. В.** Программируемые контроллеры в системах промышленной автоматизации: учебник / О. В. Шишов. – Москва: ИНФРА-М, 2020. – 365 с.