



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Кафедра «Цифровые технологии и платформы в электроэнергетике»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к практическим занятиям
для студентов по дисциплине:
«Автоматизация энергетических систем»
(задание на выполнение контрольной работы)**

Ростов-на-Дону
2023

УДК 681.5 (075.8)

Составитель: Луконин А.В.

Методические указания к практическим занятиям для студентов по дисциплине «Автоматизация энергетических систем». – Ростов-на-Дону: Донской государственный технический университет, 2023. – 50 с.

Настоящие указания содержат индивидуальные задания и рекомендации по выполнению практических занятий, предусмотренных программой дисциплины «Автоматизация энергетических систем». Представлены примеры типовой организации автоматических и автоматизированных схем и алгоритмов их функционирования.

Предназначены для обучающихся по направлению подготовки «Электроэнергетика и электротехника».

Ответственный за выпуск зав. кафедрой «Цифровые технологии и платформы в электроэнергетике», к.т.н. Юров А.А.

В печать _____
Формат 60×84/16. Объем _____ усл. п. л.
Тираж 50 экз. Заказ № _____

Издательский центр ДГТУ
Адрес университета и полиграфического предприятия:
344000, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный
технический университет, 2023
© Луконин А.В.

Практическая работа №1

Тема: «Аппаратура вторичных цепей электроустановок»

Цель работы:

Изучить аппаратуру вторичных цепей схем управления электроустановок. Познакомиться с графическим и позиционным обозначением элементов схем управления.

Задание:

1. Изучить графическое и позиционное обозначение элементов схем управления.
2. Описать аппаратуру вторичных цепей схем управления электроустановок в соответствии с индивидуальным заданием.
3. Ответить на контрольные вопросы.

Индивидуальное задание

Таблица 1.1

Последняя цифра зачетной книжки	Аппаратура вторичных цепей
0	Реле промежуточное
1	Реле напряжения
2	Реле времени
3	Контактор
4	Предохранитель плавкий
5	Выключатель автоматический
6	Кнопка управления
7	Лама сигнальная
8	Выключатель низковольтный
9	Реле тока

Примечание: Описание должно содержать:

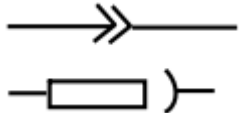
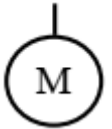





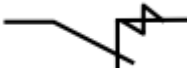

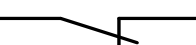

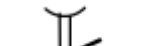



- обозначение типа и наименование, выбранного устройства;
- основные характеристики устройства;
- схему устройства с обозначением характерных точек подключения к электрической схеме;
- описание принципа действия устройства;
- фрагмент схемы использования такого устройства.

Краткая теория

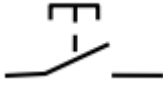
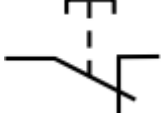
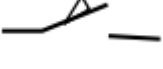

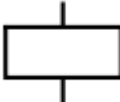


Обозначения в электрических схемах выполняется в соответствии с ГОСТ 2.710-81.

Наиболее часто используемые в схемах автоматики элементы совместно с графическим обозначением представлены в таблице 1.1.

Таблица 1.1

Объект	Графическое изображение	Обозначение на схемах
Соединение контактное разъемное		X
Электродвигатель		M
Предохранитель плавкий		FU
Рубильник, выключатель низковольтный однополюсный		QS или SA (в цепях управления и сигнализации)
Выключатель высоковольтный		Q
Выключатель автоматический		QF SF
Контакт магнитного пускателя замыкающий дугогасительный		KM
Контакт магнитного пускателя размыкающий дугогасительный		KM
Контакт замыкающий или нормально-разомкнутый или NO (normal open)		Обозначение выполняется в соответствии с обозначением устройства к которому относится контакт, например KL1.1, где KL1 – промежуточное реле, а 1- номер контакта
Контакт замыкающий или нормально-замкнутый или NC (normal closed)		
Контакт замыкающий с замедлением при срабатывании		
Контакт замыкающий с замедлением при возврате		
Контакт замыкающий с замедлением при срабатывании и возврате		
Контакт размыкающий с замедлением при срабатывании		
Контакт размыкающий с замедлением при возврате		

Продолжение таблицы 1.1

Объект	Графическое изображение	Обозначение на схемах
Кнопка управления с замыкающим контактом		SB
Кнопка управления с размыкающим контактом		SB
Контакт концевого выключателя замыкающий		SQ
Контакт концевого выключателя размыкающий		SQ
Катушка электромеханического устройства. Общее обозначение		K
Катушка электромеханического устройства: Реле тока Реле напряжения Реле указательное Реле промежуточное Реле мощности Реле времени Реле сопротивления		KA KV KH KL KW KT KZ
Лампа накаливания а) осветительная, б) сигнальная	 или 	а) EL б) HL

Контрольные вопросы

1. Как обозначается (графическое и буквенно-цифровое) катушка электромеханического устройства и его контакты на схемах (на примере промежуточного реле с четырьмя контактами: 1 – NC, 2 – NO, 3 – NO, 4 – NO)?
2. Как обозначается (графическое и буквенно-цифровое) катушка контактора и его контакты (силовые и вспомогательные) на схемах?
3. Что такое концевой выключатель и какое его назначение?
4. Для какого режима работы устройства отображаются контакты на электрических схемах?

Практическая работа №2

Тема: «Схемы автоматического ввода резерва»

Цель работы:

Познакомиться с принципом работы схем автоматического ввода резерва (АВР). Получить практические навыки по сборке цепей АВР.

Задание:

1. Изучить работу схем АВР.
2. Ознакомиться с основными характеристиками элементов схем АВР, в том числе, контакторов, реле контроля напряжения, автоматических выключателей.
3. Разработать схему АВР в соответствии с заданием.
4. Ответить на контрольные вопросы

Индивидуальное задание

Таблица 2.1

		Последняя цифра зачетной книжки									
		0	1	2	3	4	5	6	7	8	9
Предпоследняя цифра зачетной	0-4	С1 О	С1 П1	С1 П2	С1 ЛВ1	С1 ЛВ2	С1 ЛО1	С1 ЛО2	С1 ЛА	С1 ЛП1	С1 ЛП2
	5-9	С2 П2	С2 ЛВ1	С2 ЛВ2	С2 ЛО1	С2 ЛО2	С2 ЛА	С2 ЛП1	С2 ЛП2	С2 О	С2 П1

Примечание:

Выбор схемы

С1- схема на рисунке 2.3.

С2- схема на рисунке 2.4.

Выбор задания

О – Разработать однофазный вариант схемы

П1 – Выполнить изменения в схеме для контактора КМ1: Напряжения срабатывания контактора КМ1 380 В

П2 – Выполнить изменения в схеме для контактора КМ2: Напряжения срабатывания контактора КМ2 220 В

ЛВ1 - Выполнить изменения в схеме: Добавить индикацию для схемы управления основным вводом «Питание осуществляется от основного ввода» - красная лампа

ЛВ2 - Выполнить изменения в схеме: Добавить индикацию для схемы управления основным вводом «Питание осуществляется от резервного ввода» - красная лампа

ЛО1 - Выполнить изменения в схеме: Добавить индикацию для схемы управления основным вводом «Питание от основного ввода отключено» - зеленая лампа

ЛО2 - Выполнить изменения в схеме: Добавить индикацию для схемы управления резервным вводом «Питание от резервного ввода отключено» - зеленая лампа

ЛА - Выполнить изменения в схеме: Добавить индикацию для схемы управления «Наличие питания в цепи нагрузки» - желтая лампа

ЛП1 - Выполнить изменения в схеме: Добавить индикацию для схемы управления основным вводом «Наличие напряжения до основного ввода» - красная лампа

ЛП2 - Выполнить изменения в схеме: Добавить индикацию для схемы управления резервным вводом «Наличие напряжения до резервного ввода» - красная лампа

Краткая теория

Автоматическое включение резерва — включение автоматическим устройством резервного оборудования взамен отключившегося основного. Широко применяется в энергетике, служит для обеспечения бесперебойного электроснабжения потребителей. Автоматический ввод резерва (АВР) является частью сетевой автоматики (релейной защиты и автоматики) объектов энергетики.

Реализацию схем АВР осуществляют с помощью средств релейной защиты и автоматики: реле различного назначения, цифровых блоков защит, переключателей - изделий, включающих в себя механическую коммутационную часть, микропроцессорный блок управления, а также панель индикации и управления.

В качестве измерительного органа для АВР в высоковольтных сетях служат реле минимального напряжения (реле контроля фаз), подключённые к защищаемым участкам через трансформаторы напряжения. В случае снижения напряжения на защищаемом участке электрической сети реле даёт сигнал в схему АВР. Однако, условие отсутствия напряжения не является достаточным условием для того, чтобы устройство АВР начало свою работу. Как правило, должен быть удовлетворён ещё ряд условий:

- на защищаемом участке нет короткого замыкания;
- вводной выключатель включён;
- на соседнем участке, от которого предполагается получать питание после действия АВР имеется напряжение.

После проверки выполнения всех этих условий логическая часть АВР даёт сигнал на отключение вводного выключателя обесточенной части электрической сети и на включение выключателя резервного питания после того, как вводной выключатель отключился.

В низковольтных сетях одновременно в качестве измерительного и пускового органа могут служить реле минимального напряжения, подключённые к вводу или катушки электромагнитного контактора (Рис. 2.1) либо предназначенный для управления схемами АВР микропроцессорный контроллер АВР.

АВР подразделяется также на системы с восстановлением питания от основного ввода и без восстановления. При работе с восстановлением схемы при наличии напряжения на основном вводе с установленной выдержкой схема восстанавливает исходную конфигурацию. Схема может предусматривать режим работы АВР, например, «Вывод АВР», «Ввод АВР». При разработке схемы АВР необходимо учитывать характеристики потребителей и возможности работы вводов «в параллель».

Независимо от того, какая схема АВР выполнена, в проекте обязательно присутствуют две цепи — силовая и сигнальная. Через первую подают напряжение к потребителю, посредством второй управляют работой схемы АВР.

Силовая цепь организуется силовыми контактами контактора или контактами устройств, предназначенных для коммутации нагрузки. При организации силовой цепи с использованием электромагнитного контактора три фазы подают на входы, обозначенные как L1, L2, L3, а трехфазную нагрузку подключают к T1, T2, T3. Для схемы на рисунке 2.2 питание схемы нагрузки выполняется через силовые контакты KM1.1, KM1.2, KM1.3 контактора KM1 при питании от Ввода №1 и через силовые контакты KM2.1, KM2.2, KM2.3 контактора KM2 при питании от Ввода №2.

Сигнальная цепь является цепью управления и организации логики устройства АВР. Сигнальная цепь может иметь разнообразную конфигурацию в зависимости от задания и параметров питаемой сети. Для ее реализации необходимо изучить характеристики оборудования, используемого для ее реализации.

Контактор (лат. *Contāctor* «соприкасатель») — двухпозиционный электромагнитный аппарат, предназначенный для частых дистанционных включений и выключений силовых электрических цепей в нормальном режиме работы. Разновидность электромагнитного реле.

Наиболее широко применяются одно- и двухполюсные контакторы постоянного тока и трёхполюсные контакторы переменного тока. К контакторам из-за частых коммутаций предъявляются повышенные требования по механической и электрической износостойкости. Контактторы как постоянного, так и переменного тока содержат: электромагнитную систему, контактную систему, состоящую из подвижных и неподвижных контактов, дугогасительную систему, систему блок-контактов (вспомогательные контакты, переключающие цепи сигнализации и управления при работе контакторов). В отличие от автоматических выключателей контакторы могут коммутировать только номинальные токи, они не предназначены для отключения токов короткого замыкания.

Общая конструкция контактора включает в себя основную часть, закрепленную стационарно с неподвижными контактами и подвижный якорь,

передвигающийся по направляющим с замыкающимися контактами. В исходном положении якорь удерживается пружиной. При подаче управляющего сигнала электромагнит притягивает якорь, преодолевая её усилие, и замыкает и/или размыкает контакты в зависимости от конструкции реле. После отключения управляющего напряжения пружина возвращает якорь в исходное положение. Управляемая цепь электрически никак не связана с управляющей, то есть они гальванически изолированы друг от друга.

Для обеспечения правильной работы эксплуатационной схемы требуется соответствие параметров контактора эксплуатационным условиям, в том числе, величины. Выражение «величина» является условным термином, обозначающим то, какой ток может пропустить через главные рабочие контакты выбранный магнитный пускатель. При присвоении величины считается, что пускатель работает при напряжении 380 В, а его рабочий режим АС-3 (Пуск электродвигателя с короткозамкнутым ротором, отключение вращающихся двигателей). Всего насчитывается восемь величин магнитных пускателей и каждая из них подразумевает свой параметр нагрузочного тока: 0 – 6,3 А; 1 – 10 А; 2 – 25 А; 3 – 40 А; 4 – 63 А; 5 – 100 А; 6 – 160 А; 7 – 250 А. Допускается небольшое несоответствие (в большую сторону) по допустимому значению тока.

Управление контактором осуществляется посредством вспомогательной цепи, обычно переменного тока, проходящего по катушкам контактора, напряжением 24, 36, 42, 110/127, 220 или 380 Вольт, возможно специализированное исполнение катушки контактора.

В схемах АВР трехфазной сети возможна установка реле контроля фаз, которое обеспечивает постоянный контроль за напряжением ввода. В случае снижения или повышения напряжения на вводе, неисправности или обрыва любой из фаз реле производит переключение потребителя на другой ввод, тем самым, обеспечивая защиту электрооборудования от аварийных режимов электрической сети. Реле также контролирует порядок чередования фаз (фазировка), что позволяет определить корректность питающего напряжения, приходящего к потребителю. К тому же эти реле работают в комплекте с электрооборудованием, для которого неправильное чередование фаз может привести к поломке или неправильной работе.

В число основных требований к системам аварийного восстановления электроснабжения входит:

- схема АВР должна приходить в действие при исчезновении напряжения в линии потребителя по любой причине, в том числе при аварийном, ошибочном или самопроизвольном отключении выключателей рабочего источника питания;
- включение резервного источника питания должно производиться с задержкой по времени, исключающей ложные срабатывания АВР. Либо без задержки, в случаях, когда не допускается даже кратковременная потеря питания: например, больницы, центры обработки данных (ЦОД), аэропорты;
- устройство АВР не должно приходить в действие до отключения выключателя рабочего источника, чтобы избежать одновременного подключения двух источников, которое может привести к аварии в системе.

Какие бывают схемы автоматического ввода резерва?

В простейшем варианте подключения используются 2 источника энергии. Они предназначены для бесперебойного энергоснабжения одного потребителя. Такие схемы принято называть «2 в 1».

Существует два основных типа исполнения, различающиеся приоритетом ввода:

Одностороннее. В таких АВР один ввод играет роль рабочего, то есть используется, пока в линии не возникнут проблемы. Второй – является резервным, и подключается, когда в этом возникает необходимость.

Двухстороннее. В этом случае нет деления на рабочую и резервную секцию, поскольку оба ввода имеют одинаковый приоритет.

Для некоммерческих и промышленных предприятий применяются более сложные схемы и их комбинации. Например, довольно распространённая схема «2 в 2» (два рабочих ввода на две секции шин потребителя) используется в качестве АВР в низковольтных трансформаторных подстанциях на стороне 0,4 кВ и обеспечивает бесперебойное электроснабжение городских объектов инфраструктуры. Для потребителей особой группы, где необходимы три независимых источника, используются схемы «3 в 1» (каскад) и «3 в 2».

Как правило, каждый потребитель подключается к секции шин распределительного устройства низкого напряжения (РУНН) с помощью автоматического выключателя. При переключении такой секции на резервный ввод мощности источника может быть недостаточно для суммарной нагрузки секции шин. В таких случаях часть неприоритетных нагрузок может быть отключена для сохранения энергетического баланса системы.

Если говорить о технической реализации схем, как правило, АВР выполняется на контакторах, автоматических выключателях или реверсивных выключателях нагрузки с моторным приводом и блоками управления. Выбор коммутационного оборудования в схемах АВР в первую очередь зависит от номинальных токов, типа схемы («2 в 1», «2 в 2», «3 в 2»), уровня диспетчеризации и автоматизации потребителя энергоресурсов.

Работа схем АВР

Рассмотрим схему, представленную на рисунке 2.1 с основным и резервным вводами. Наличие напряжения на основном вводе контролируется контактором КМ1. При включении автоматического выключателя QF1 и наличии напряжения на основном вводе контактор КМ1 срабатывает и замыкается контактная группа КМ1.1, КМ1.2 (для схемы на рисунке 2.1 а) и КМ1.1, КМ1.2, КМ1.3 (для схемы на рисунке 2.1 б), обеспечивающая подключение основного ввода к нагрузке. При исчезновении напряжения на основном вводе контактор КМ1 отключается и замыкает цепь питания нагрузки через контактную группу КМ1.3, КМ1.4 (для схемы на рисунке 2.1 а) и КМ1.4, КМ1.5, КМ1.6 (для схемы на рисунке 2.1 б), обеспечивающая подключение резервного ввода к нагрузке. При восстановлении питания основного ввода схема автоматически переходит на питание нагрузки от основного ввода.

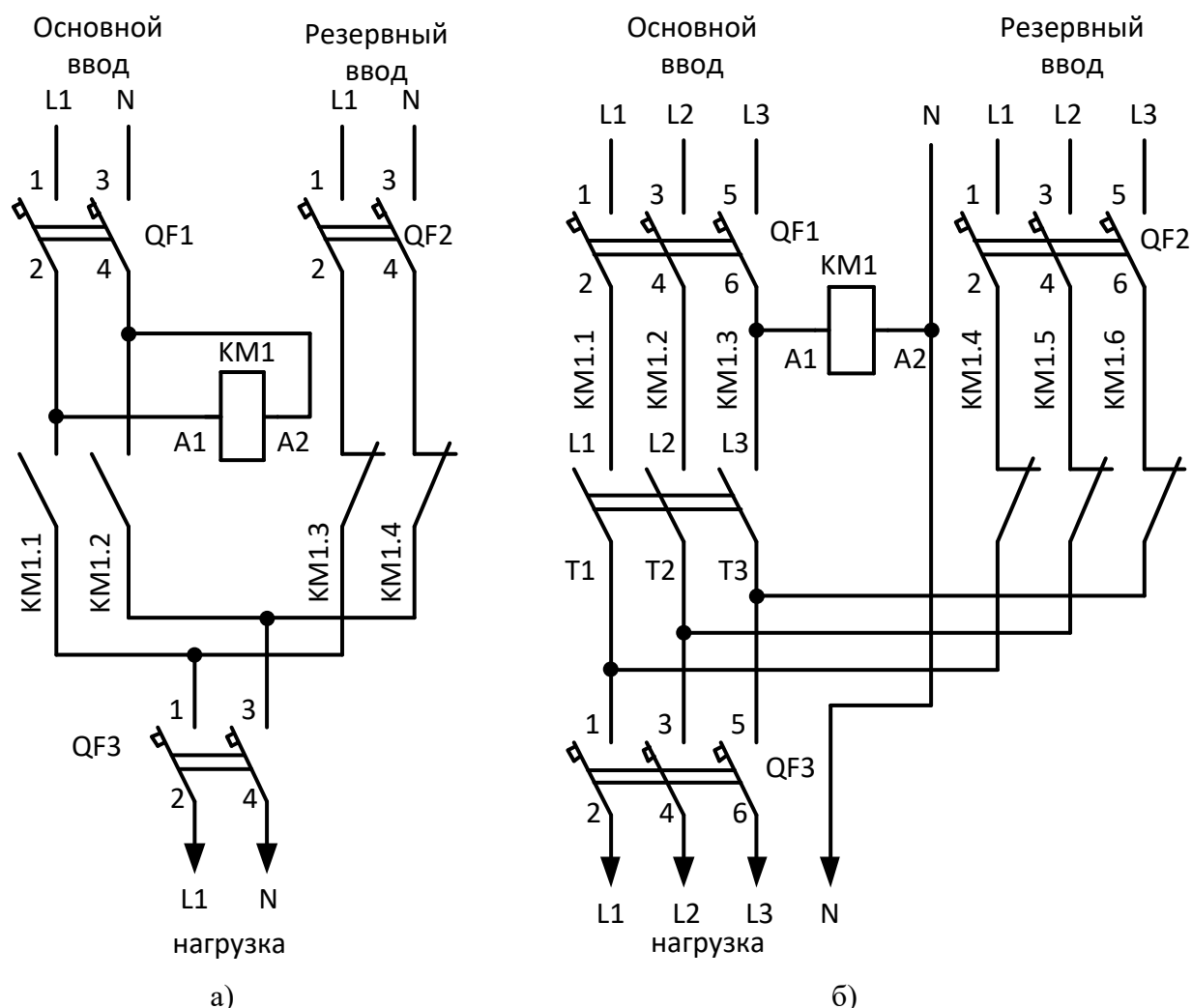


Рис. 2.1. Простая схема АВР с основным и резервным вводом а) однофазного и б) трехфазного исполнения

Рассмотрим схему, представленную на рисунке 2.2 с двумя равноправными вводами. Чтобы не допустить одновременное подключение вводов к схеме нагрузки применяется принцип взаимной блокировки, реализованный на дополнительных контактах KM2.4 и KM1.4 контакторов KM1 и KM2. От какого ввода будет питаться нагрузка, определяется очередностью включения автоматов QF1 и QF2. Если первым включается QF1, то срабатывает контактор KM1, при этом разрывается контакт KM1.4, блокируя поступление напряжения на катушку KM2, а также замыкается контактная группа KM1.1, KM1.2, KM1.3, обеспечивающая подключение Ввода №1 к нагрузке. При включении QF2 конфигурация схемы не изменяется. При отключении QF1 или исчезновении напряжения на Вводе №1 отключается контактор KM1, при этом собирается цепь включения контактора KM2 через контакт KM1.4 и контактор KM2 срабатывает. Разрывается контакт KM2.4 блокируя поступление напряжения на катушку KM1, а также замыкается контактная группа KM2.1, KM2.2, KM2.3, обеспечивающая подключение Ввода №2 к нагрузке.

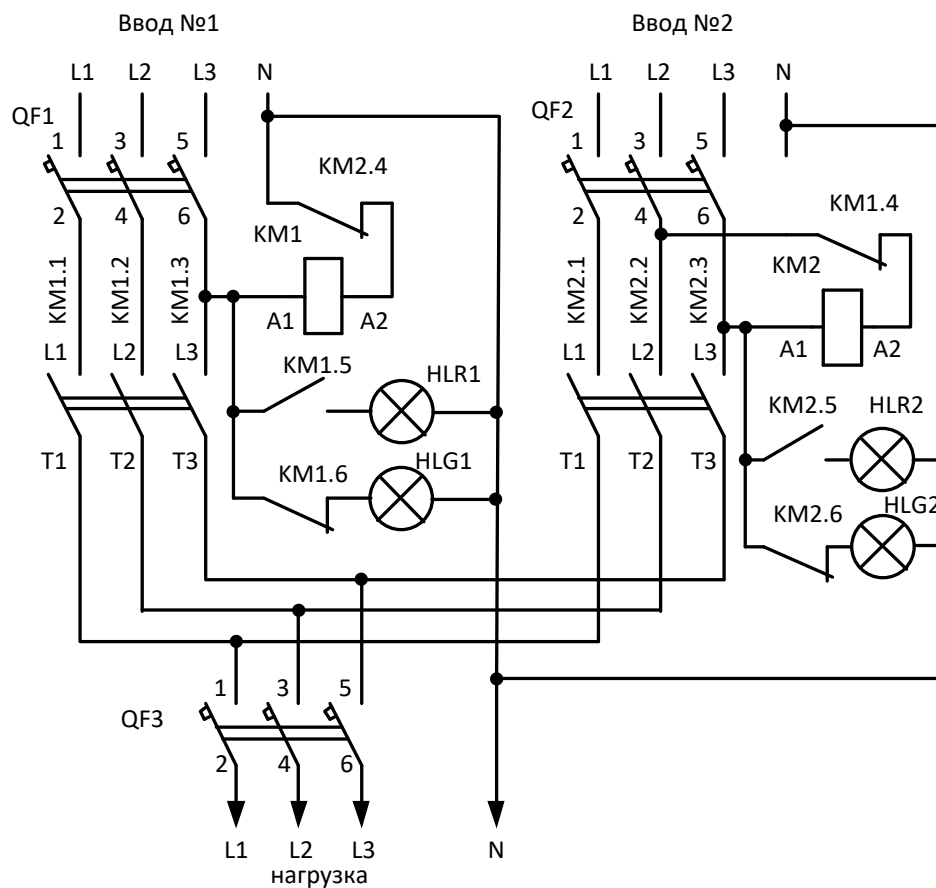


Рис. 2.2. Схема АВР с двумя рабочими вводами трехфазного исполнения

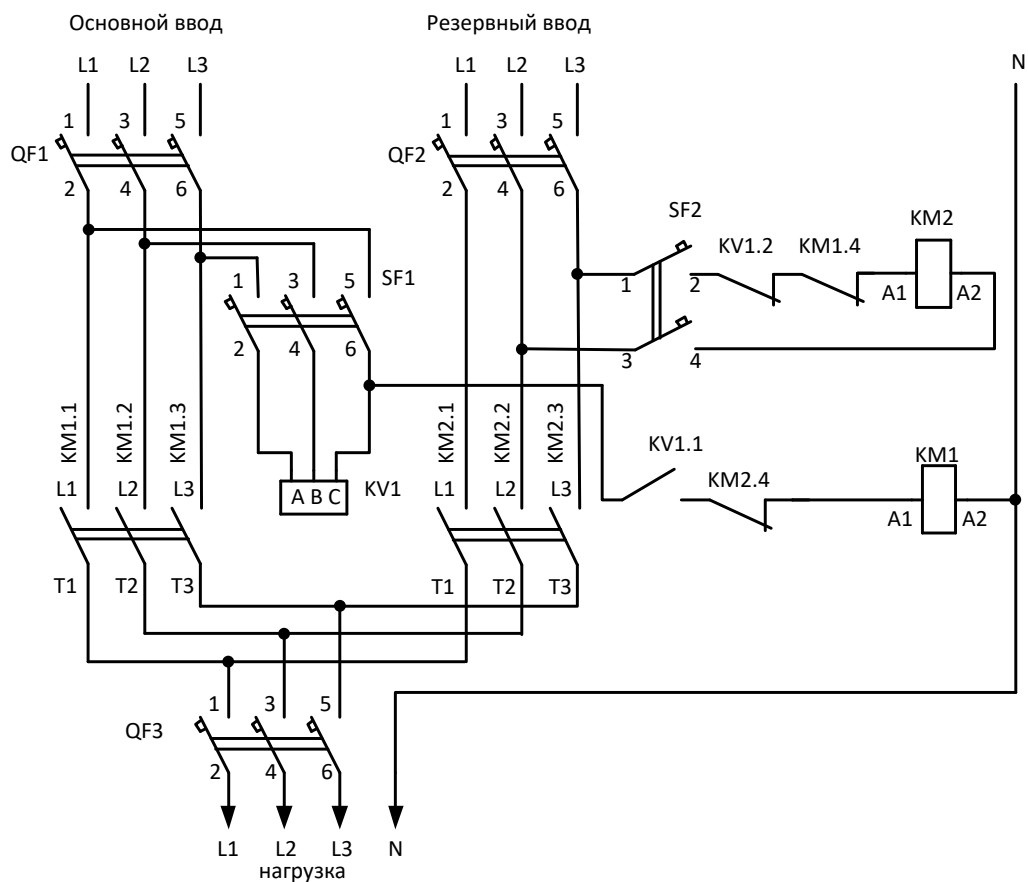


Рис. 2.3. Схема АВР с рабочим и резервным вводами (вариант 1)

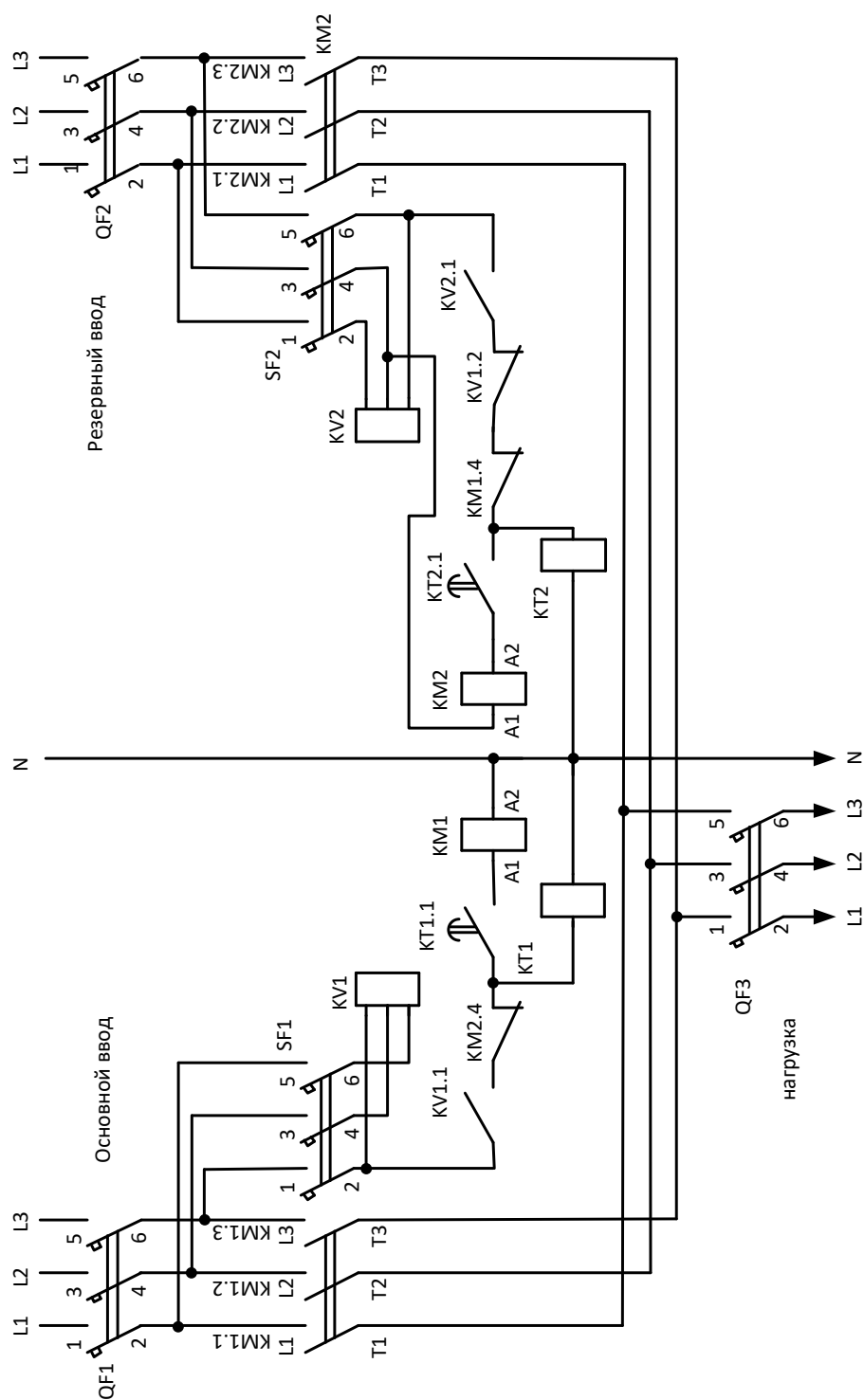


Рис. 2.4. Схема АВР с рабочим и резервным вводами (вариант 2)

Контрольные вопросы

1. Назначение контактора.
2. Как организуется цепь взаимной блокировки контакторов?
3. Какое назначение у силовой части АВР?
4. Какое назначение у сигнальной части АВР?
5. Как определить схему подключения контактора для сети 380 В?
6. Что такое величина контактора?

Практическая работа №3

Тема: «Схемы управления двигателями»

Цель работы:

Познакомиться с принципом работы схем управления. Получить практические навыки по сборке цепей управления.

Задание:

1. Изучить принципы работы схем управления.
2. Разработать схему дистанционного управления двигателем в соответствии с индивидуальным заданием.
3. Ответить на контрольные вопросы

Индивидуальное задание

		Последняя цифра зачетной книжки									
		0	1	2	3	4	5	6	7	8	9
Предпоследняя цифра зачетной книжки	0-2	С1 П1	С1 ПП1	С1 КК	С1 ЛВ1	С3 П2	С3 ПП1	С3 ЛВ3	С3 ЛВ2	С4 П1	С4 ПП1
	3-5	С2 П2	С4 КП	С2 ЛВ3	С2 ЛВ2	С1 П1	С1 ПП1	С1 КК	С1 ЛВ1	С3 ЛВ2	С2 П2
	6-7	С3 П2	С3 ПП1	С3 ЛВ3	С3 ЛВ2	С2 П2	С4 КП	С2 ЛВ3	С2 ЛВ2	С4 КП	С2 ЛВ3
	8-9	С4 П1	С4 ПП1	С4 КК	С4 ЛВ1	С3 П2	С3 ПП1	С3 ЛВ3	С3 ЛВ2	С1 П1	С1 ПП1

Примечание:

Выбор схемы

С1- схема на рисунке 3.1.

С2- схема на рисунке 3.2.

С3- схема на рисунке 3.3.

С4- схема на рисунке 3.4.

Выбор задания

Выполнить изменения в схеме:

П1 –для магнитного пускателя КМ1: Напряжения срабатывания 380 В

П2 –для магнитного пускателя КМ2: Напряжения срабатывания 220 В

ПП1 – Предусмотреть питание сигнальной схемы от автоматического выключателя.

КК - Предусмотреть установку в схеме тепловой защиты.

ЛВ1 - Добавить индикацию «Двигатель включен» - красная лампа
ЛВ2 - Добавить индикацию «Вперед» - красная лампа.
ЛВ3 - Добавить индикацию «Назад» - красная лампа.
КП - Добавить дополнительный кнопочный пост.

Краткая теория

Современное оборудование часто работает в автоматическом и полуавтоматическом режиме. Наиболее распространенный элемент схем управления является асинхронный двигатель с короткозамкнутым ротором. Асинхронные двигатели активно используются во многих отраслях промышленности и сельского хозяйства. Они потребляют примерно 70% всей энергии, предназначенной для преобразования электричества во вращательное или поступательное движение. Асинхронные двигатели зарекомендовали себя наиболее эффективными в качестве электрической тяги, без которой не обходятся многие технологические операции, в том числе, на объектах электроэнергетики.

Схемы управления электродвигателями могут быть различны, так как принципы автоматического и полуавтоматического режима работы могут существенно отличаться друг от друга.

Полуавтоматическое управление предусматривает участие оператора, который инициирует пуск оборудования нажатием соответствующей кнопки или поворачивая рычаг. После этого функция персонала заключается лишь в контроле рабочего процесса. При автоматическом управлении первоначальный пуск оборудования осуществляют реле или датчики, после чего работы выполняется в соответствии с заданными программами. Такое программное устройство часто выполняется с помощью логических схем, вариантов которых может быть довольно много. В промышленности наиболее часто встречаются следующие схемы управления асинхронными электродвигателями:

- нереверсивного управления;
- реверсивного управления с двумя магнитными пускателями.

При использовании схемы управления электродвигателем с нереверсивным управлением (Рис. 3.1) после пуска (нажать SB2) происходит подключение к сети электромагнитной катушки (KM1). С ее сердечником соприкасается подвижный якорь и замыкает силовые контакты (KM1.1, KM1.2, KM1.3). В результате на двигатель подается трехфазное напряжение. Параллельно с силовыми контактами замыкаются блокировочный (KM1.4), что позволяет зашунтировать кнопку пуска (SB2) и отпустить ее. Контакт контактора KM1.4 выполняет функцию «самоподхвата» схемы. Нажимая кнопку стоп (SB1), оператор тем самым разрывает цепь, от которой запитана электромагнитная катушка. Это освобождает якорь, который при падении размыкает силовые контакты, что приводит к остановке оборудования.

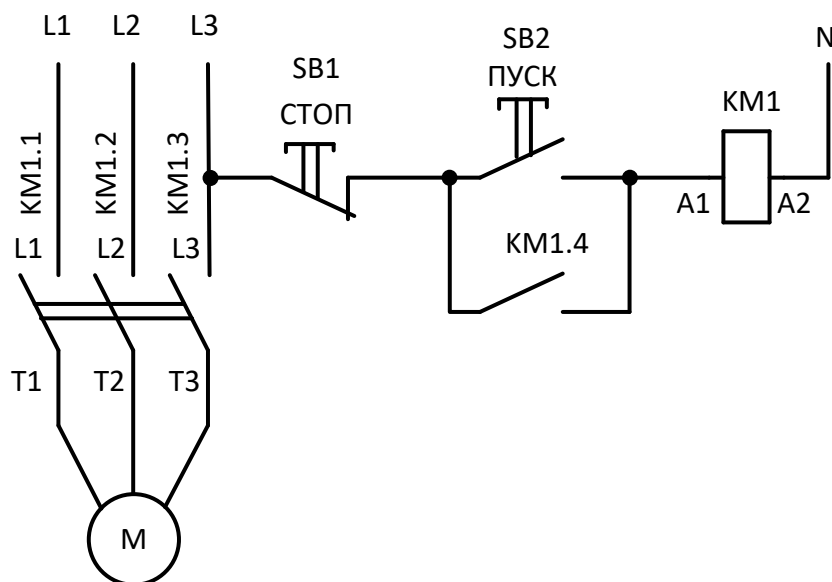


Рис. 3.1. Схема нереверсивного управления двигателем

Схема управления реверсивным двигателем с помощью двух магнитных контакторов (Рис. 3.2) работает аналогично предыдущей. Изменение направления вращения (реверс) ротор двигателя меняет при изменении порядка чередования фаз на его статоре. При включении контактора КМ1 на двигатель приходят фазы - L1, L2, L3, а при включении контактора КМ2 - порядок фаз меняется на L3, L2, L1. В этой схеме управления защита от длительной перегрузки обеспечивается подключением теплового реле КК. Кроме этого, выполнено отдельное питание сигнальной схемы от автоматического выключателя SF1.

Включение двигателя на вращение в одну сторону осуществляется кнопкой SBC1 и электромагнитным контактором КМ1. При необходимости смены направления вращения необходимо нажать на кнопку SBT "Стоп", двигатель остановится и после этого при нажатии на кнопку SBC2 двигатель начинает вращаться в другую сторону. В этой схеме для смены направления вращения ротора необходимо промежуточное нажатие на кнопку "Стоп". В схеме обязательно использование в цепях каждого из контакторов нормально-закрытых (размыкающих) контактов для обеспечения защиты от одновременного нажатия двух кнопок "Пуск" SBC1 и SBC2, что приведет к короткому замыканию в цепях питания двигателя. Дополнительные контакты в цепях контакторов КМ1.4 и КМ2.4 не дают им включиться одновременно, т.к. какой-либо из контакторов при нажатии на обе кнопки "Пуск" включиться на секунду раньше и разомкнет свой контакт в цепи другого контактора.

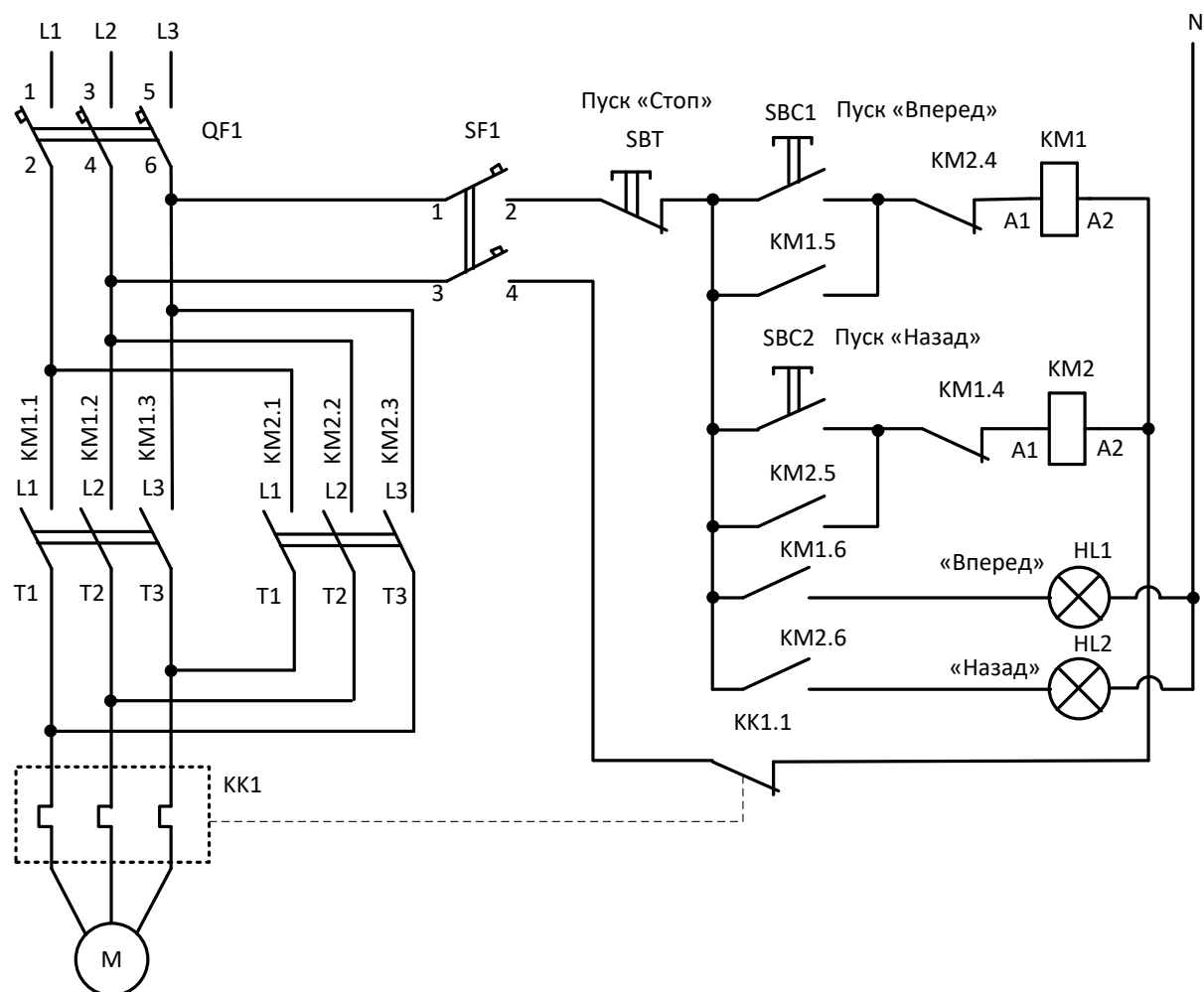


Рис. 3.2. Реверсивная схема подключения асинхронного двигателя (вариант 1)

Необходимость в создании такой блокировки требует использования контакторов с большим количеством контактов, что удорожает и усложняет электрическую схему.

Схема управления реверсивным двигателем с помощью двух магнитных контакторов и трех кнопок (две из которых имеют контакты с механической связью) показана на рисунке 3.3.

Отличие этой схемы от предыдущей в том, что в цепи каждого контактора кроме общей кнопки SBT "Стоп" включены по 2 контакта кнопок SB2 и SB3, причем в цепи KM1 кнопка SB2 имеет нормально-открытый контакт (замыкающий), а SB3 - нормально-закрытый (размыкающий) контакт, в цепи KM2 – кнопка SB2 имеет нормально-закрытый контакт (размыкающий), а SB3 - нормально-открытый. При нажатии каждой из кнопок цепь одного из контакторов замыкается, а цепь другого одновременно при этом размыкается. Такое использование кнопок позволяет отказаться от использования дополнительных контактов для защиты от одновременного включения двух пускателей (такой режим при этой схеме невозможен) и дает возможность выполнять реверс без промежуточного нажатия на кнопку "Стоп", что очень удобно. Кнопка "Стоп" нужна для окончательной остановки двигателя. Еще

одной особенностью схемы является наличие схемы сигнализации о состоянии работы схемы на лампах HL1 «Вперед» и HL2 «Назад».

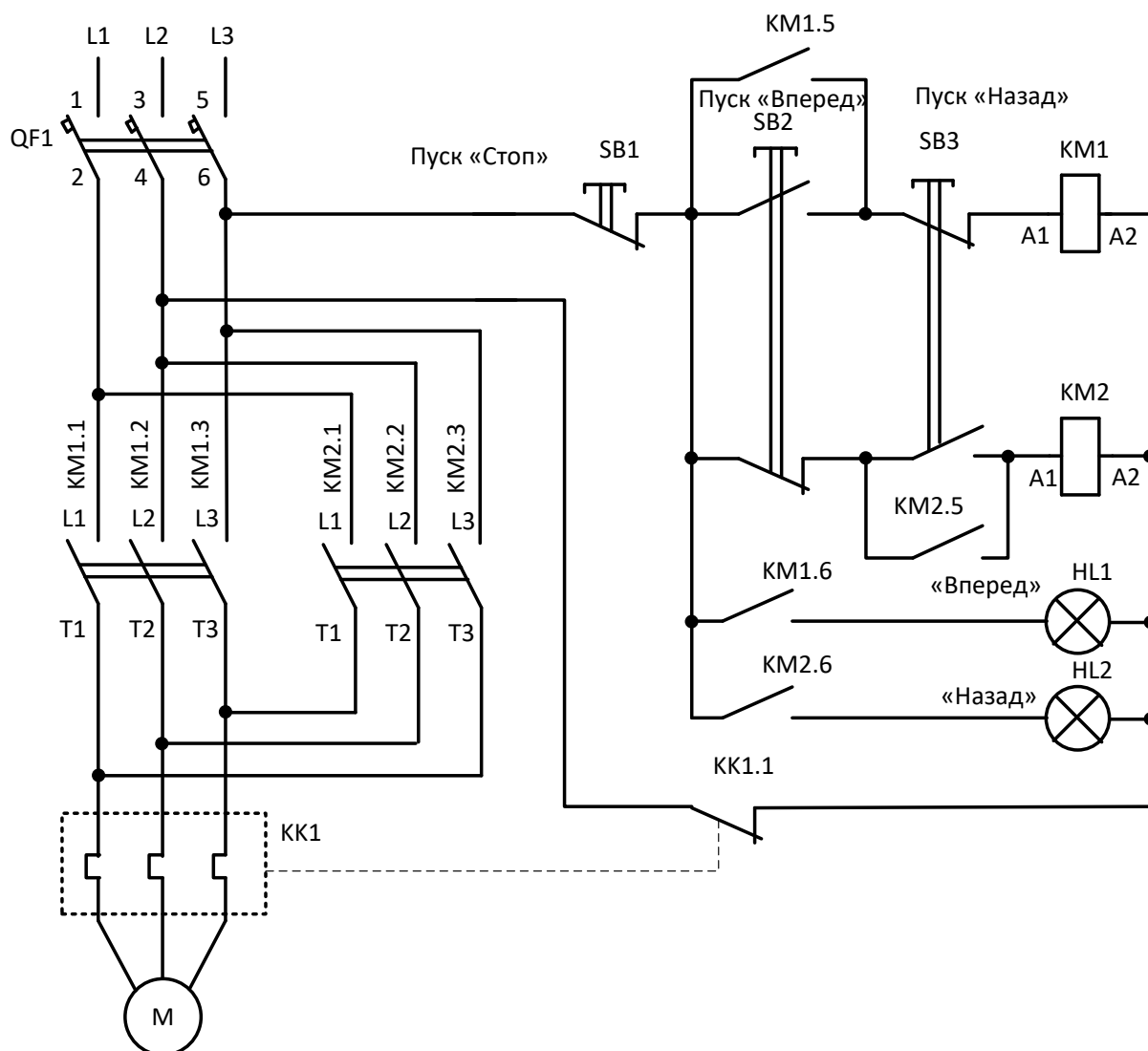


Рис. 3.3. Реверсивная схема подключения асинхронного двигателя (вариант 2)

На рисунке 3.4 представлена схема с управлением с двух рабочих мест. Элементы управления такой схемы находятся в одном кожухе, который называется «кнопочным постом». Пуск и останов схемы осуществляется с любого из кнопочных постов.

Наличие схемы управления электрическими двигателями предполагает наличие пускателя. Пускатель электромагнитный (магнитный пускатель) — электрический аппарат, который предназначен для пуска, останова, реверсирования и защиты электродвигателя. Магнитный пускатель состоит из контактора, кнопочного поста и теплового реле. Наряду с тепловой защитой магнитный пускатель предотвращает самовключение после восстановления исчезнувшего питающего напряжения. Главным образом магнитный пускатель применяется для дистанционного пуска, останова и защиты трехфазных асинхронных электродвигателей с короткозамкнутым ротором. Контакты

магнитных пускателей, коммутирующих токи выше 20..25 А, снабжают дугогасительными устройствами.

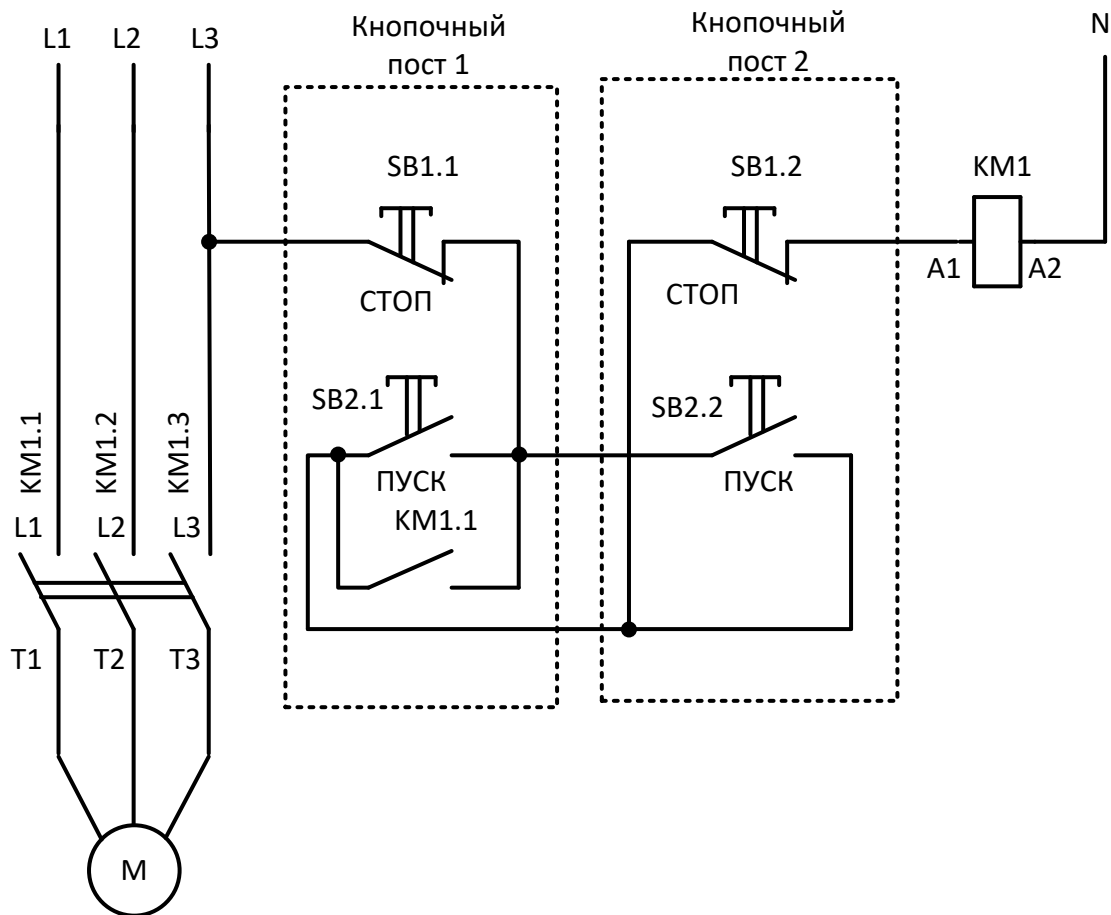


Рис. 3.4. Схема управления двигателем с использованием двух кнопочных постов

В рамках стандартизации к пускателям относят комбинацию всех коммутационных устройств, необходимых для пуска и остановки двигателя, с защитой от перегрузок. Электромагнитным является пускатель, у которого сила, необходимая для замыкания главных контактов, обеспечивается электромагнитом. Наряду с электромагнитными существуют полупроводниковые, реостатные, ручные, с двигателем приводом, пневматические, электропневматические пускатели. Пускатель, включающий расширенные функции с возможностью обмена информацией, называется пускатель управления электродвигателя.

Контрольные вопросы

1. Что такое и какое назначение дугогасительной камеры контактора?
2. Что такое пускатель?
3. Как выполняется полуавтоматическая схема с самоподхватом?
4. Что инициирует пуск автоматической схемы пуска электродвигателя?
5. Что такое кнопочный пост?

Практическая работа №4

Тема: «Логические приемы составления и анализа релейно-контактных схем»

Цель работы: Изучить логические приемы составления и анализа релейно-контактных схем.

Задание:

1. Составить релейно-контактную схему в соответствии с логическим выражением индивидуального задания.
2. Составить таблицу истинности для выражения индивидуального задания.
3. Оптимизировать логическое выражение индивидуального задания по законам алгебры логики.
4. Составить релейно-контактную схему оптимизированного логического выражения индивидуального задания.
5. Составить таблицу истинности для оптимизированного выражения индивидуального задания.
6. Сравнить таблицы истинности для оптимизированного выражения индивидуального задания и исходного.
7. Ответить на контрольные вопросы.

Индивидуальное задание

Таблица 4.1

№ по порядку в списке группы	Логическое выражение
1	$Y = X1 \cdot \overline{X2} + (X3 \cdot X4 + \overline{X1} \cdot X2 \cdot X3) \cdot X4$
2	$Y = (\overline{X1} \cdot \overline{X2} + X3 \cdot \overline{X4} \cdot \overline{X1} + X2 \cdot X3) \cdot X4$
3	$Y = \overline{X1} \cdot X3 \cdot \overline{X2} + X4 \cdot \overline{X1} \cdot (X2 + X3) \cdot \overline{X4}$
4	$Y = \overline{X1} \cdot X3 \cdot (X2 + X4 \cdot \overline{X1} \cdot X2) \cdot (X3 + \overline{X4})$
5	$Y = \overline{X1} \cdot X2 \cdot \overline{X3} \cdot \overline{X4} + X1 \cdot (X2 + X3 + \overline{X4})$
6	$Y = X1 \cdot (X2 + X3 \cdot \overline{X4} + \overline{X1} \cdot (\overline{X2} + X3 + \overline{X4}))$
7	$Y = \overline{X1} \cdot \overline{X2} \cdot \overline{X3} \cdot (X4 + \overline{X1} + X2) \cdot (\overline{X3} + X4)$
8	$Y = \overline{X1} \cdot X2 \cdot X3 \cdot (X4 + \overline{X1} \cdot X2 \cdot X3 \cdot X4)$
9	$Y = X1 \cdot \overline{X2} \cdot X3 \cdot (X4 + \overline{X1} \cdot X2 \cdot X3 + \overline{X4})$
10	$Y = X1 \cdot X2 \cdot X3 \cdot \overline{X4} \cdot (\overline{X1} + X2) \cdot X3 \cdot \overline{X4}$
11	$Y = \overline{X1 + X2 \cdot X3 + X4 + X1 \cdot \overline{X2} \cdot X3 + X4}$
12	$Y = \overline{X1} \cdot \overline{X2} \cdot \overline{X3 + X4} \cdot X1 \cdot (\overline{X2} + X3 \cdot \overline{X4})$
13	$Y = X1 \cdot X2 \cdot (\overline{X3 + X4}) + \overline{X1} \cdot X2 \cdot X3 + X4$
14	$Y = \overline{X1} \cdot \overline{X2} \cdot (X3 + X4 \cdot X1) + X2 \cdot \overline{X3} \cdot X4$
15	$Y = X1 \cdot (X2 + \overline{X3} \cdot X4 \cdot \overline{X1}) + \overline{X2} \cdot X3 + X4$

16	$Y = \overline{X1} \cdot X2 \cdot (X3+X4 \cdot \overline{X1}) \cdot X2 \cdot (\overline{X3}+X4)$
17	$Y = X1+X2+X3 \cdot \overline{X4}+\overline{X1} \cdot X2 \cdot \overline{X3}+\overline{X4}$
18	$Y = (\overline{X1}+X2+\overline{X3}) \cdot \overline{X4} \cdot X1 \cdot X2 \cdot (X3+X4)$
19	$Y = \overline{X1} \cdot \overline{X2} \cdot X3+\overline{X4} \cdot X1 \cdot X2+(\overline{X3}+X4)$
20	$Y = (\overline{X1}+X2) \cdot X3+X4 \cdot \overline{X1}+X2 \cdot X3 \cdot \overline{X4}$
21	$Y = \overline{X1} \cdot (\overline{X2}+X3) \cdot \overline{X4} \cdot X1 \cdot X2 \cdot (X3+\overline{X4})$
22	$Y = (\overline{X1}+X2+\overline{X3}) \cdot X4 \cdot \overline{X1} \cdot (\overline{X2}+X3) \cdot X4$
23	$Y = (\overline{X1}+X2) \cdot X3 \cdot X4 \cdot X1 \cdot (X2+X3+\overline{X4})$
24	$Y = X1+X2 \cdot \overline{\overline{X3}+X4} \cdot X1 \cdot X2+X3+\overline{X4}$
25	$Y = \overline{X1} \cdot X2+X3 \cdot \overline{X4} \cdot X1 \cdot X2+\overline{X3}+X4$
26	$Y = \overline{X1}+X2+X3 \cdot X4 \cdot \overline{\overline{X1} \cdot X2+X3}+X4$
27	$Y = (X1+\overline{X2}+X3) \cdot \overline{X4} \cdot X1 \cdot X2+X3 \cdot X4$
30	$Y = \overline{X1}+(\overline{X2}+X3 \cdot \overline{X4} \cdot X1)+X2+X3 \cdot X4$
31	$Y = (X1+X2+\overline{X3} \cdot X4)+\overline{\overline{X1} \cdot (X2+X3)}+X4$
32	$Y = X1 \cdot \overline{X2}+(X3+\overline{X4} \cdot X1 \cdot X2+X3)+\overline{X4}$

Краткая теория

Высказывание - это повествовательное предложение, про которое можно определенно сказать истинно оно или ложно (истина (логическая 1), ложь (логический 0)).

Логические операции - мыслительные действия, результатом которых является изменение содержания или объема понятий, а также образование новых понятий.

Логическое выражение - устное утверждение или запись, в которое, наряду с постоянными величинами, обязательно входят переменные величины (объекты). В зависимости от значений этих переменных величин (объектов) логическое выражение может принимать одно из двух возможных значений: истина (логическая 1) или ложь (логический 0).

Сложное логическое выражение - логическое выражение, состоящее из одного или нескольких простых логических выражений (или сложных логических выражений), соединенных с помощью логических операций.

Булева алгебра - это формальный способ описания логических операций. Булева алгебра является разделом алгебры со значениями переменных *true* и *false*, обычно обозначаемые 1 и 0, и использует логические операторы, такие как конъюнкция (И, *AND*, \wedge), дизъюнкция (ИЛИ, *OR*, \vee) и отрицание (НЕ, *NOT*, \neg).

К основным функциям алгебры логики относятся:

1. логическое умножение - конъюнкция (функция И, *AND*);

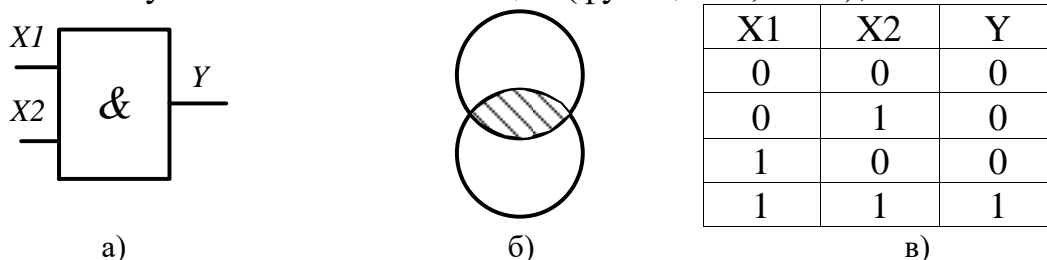


Рис. 4.1. Основные характеристики конъюнкции: а) логическое обозначение функционального знака; б) диаграмма Венна; в) таблица истинности.

2. логическое сложение - дизъюнкция (функция ИЛИ, *OR*);

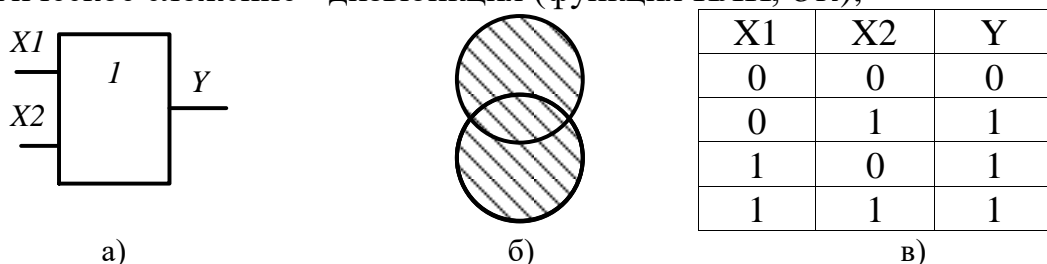


Рис. 4.2. Основные характеристики дизъюнкции: а) логическое обозначение функционального знака; б) диаграмма Венна; в) таблица истинности.

3. логическое отрицание - инверсия (функция НЕ, *NOT*).

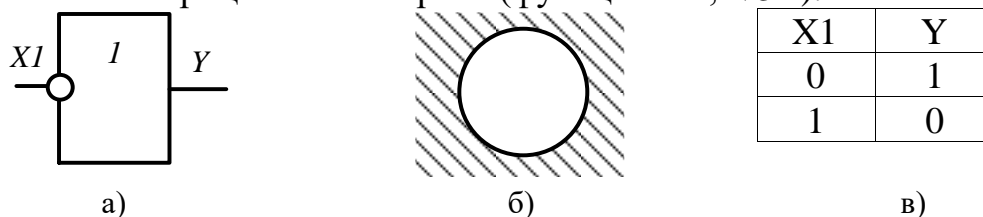


Рис. 4.3. Основные характеристики инверсии: а) логическое обозначение функционального знака; б) диаграмма Венна; в) таблица истинности.

4. логическое сложение по модулю – строгой дизъюнкцией (функция исключающее ИЛИ, *XOR*).

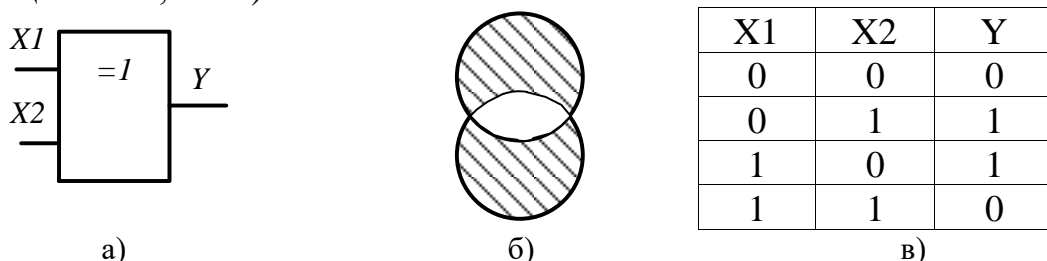


Рис. 4.4. Основные характеристики строгой дизъюнкции: а) логическое обозначение функционального знака; б) диаграмма Венна; в) таблица истинности.

Сложные функции алгебры логики могут быть получены из простых путем объединения их между собой определенными логическими связями. Такими функциями являются Штрих Шеффера (И-НЕ, *NAND*), Стрелка Пирса (ИЛИ-НЕ, *NOR*), Эквивалентность, Импликация, Альтернатива и другие. Использование их для логических высказываний упрощает реализацию логических преобразований.

Схемы автоматического управления разрабатываются для того, чтобы получить определенную последовательность действий исполнительных органов в зависимости от входных команд, подаваемых извне.

В зависимости от задаваемых комбинаций входных команд и определенной последовательности их поступления должны включаться в строго заданной последовательности определенные исполнительные органы системы автоматического релейно-контактного управления. Эта строгая последовательность определяется логически обоснованной структурой схемы.

Результат выполнения конкретной операции схемы может быть описан таблицей истинности - таблицей, которая описывает логическую функцию. Под «логической функцией» в данном случае понимается функция, у которой значения переменных (параметров функции) и значение самой функции выражают логическую истинность. Такие таблицы оказались довольно удобными, и с начала XX века за ними закрепилось это специальное название. Особенно часто таблицы истинности применяются в булевой алгебре.

Сама работа схем релейно-контактной автоматики основана на вполне определенных логических преобразованиях входных команд и перемещениях выходных исполнительных органов.

Математический аппарат алгебры логики имеет дело с двумя состояниями переменных: истинным и ложным (1 или 0), и релейно-контактные схемы также характеризуются двумя состояниями электрических цепей: они могут быть только замкнутыми или только разомкнутыми. Это дает возможность при анализе и синтезе релейно-контактных схем использовать законы алгебры логики.

Исходя из вышеизложенного, видим, что аналогом конъюнкции (а) является последовательное соединение элементов, аналогом дизъюнкции (б) - параллельное, а аналогом инверсии - нормально замкнутый контакт (в), Рис. 4.5.

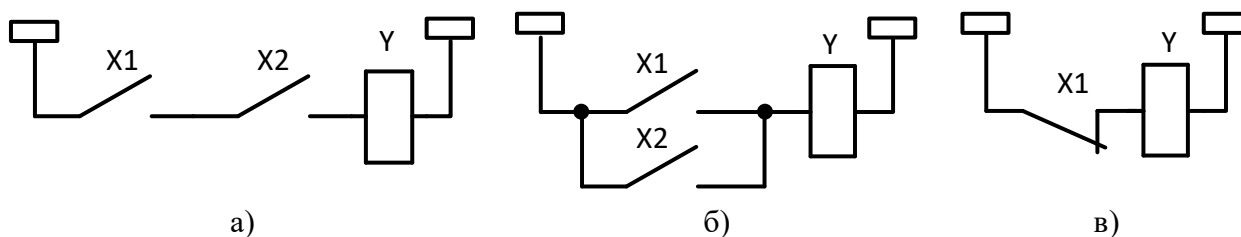
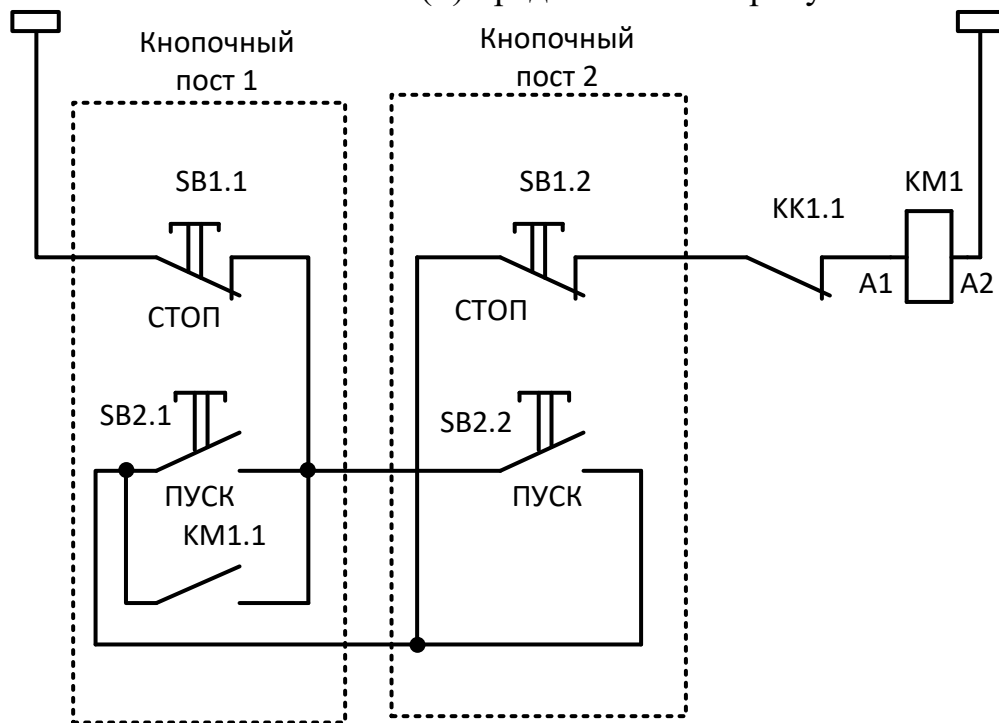


Рис. 4.5. Контактные аналоги логических функций: а) И; б) ИЛИ; в) НЕ.

При анализе и синтезе релейно-контактных схем пользуются аналитической записью работы отдельных элементов схем и работы всей схемы в целом. При этом различные вспомогательные элементы схем (трансформаторы и выпрямители для питания цепей управления, защитные аппараты - плавкие предохранители, тепловые реле, клеммы, разъемы и т.д.), непосредственно не участвующие в создании логики схемы в запись не вводятся. После получения структуры схемы такие вспомогательные элементы вводятся в них с учетом необходимости обеспечения защиты отдельных цепей и механизмов, а также для

создания удобства и безопасности обслуживания. Пример релейно-контактной схемы (а) и ее аналитическая запись (б) представлены на рисунке 4.6.



а)

$$F(KM1) = SB1.1 \wedge SB1.2 \wedge (SB2.1 \vee SB2.2 \vee KM1.1)$$

или

$$F(KM1) = SB1.1 \cdot SB1.2 \cdot (SB2.1 + SB2.2 + KM1.1)$$

б)

Рис. 4.6. Релейно-контактная схема (а) и ее аналитическая запись (б)

Для оптимизации схем необходимо использовать законы преобразования релейно-контактных схем. Алгебра логики и соответственно базирующаяся на ней алгебра релейных схем имеют свои законы во многом напоминающие законы обычной алгебры. Ниже приведены соотношения входных сигналов (контактов) с единицей и нулем:

$$x \cdot 1 = x;$$

$$x + 0 = x;$$

$$x \cdot 0 = 0;$$

$$x + 1 = 1;$$

$$x \cdot \bar{x} = 0;$$

$$x + \bar{x} = 1.$$

Для алгебры релейных схем справедливы переместительный и сочетательный законы, а также распределительный закон умножения относительно сложения.

Переместительный закон:

$$x \cdot y = y \cdot x \text{ - для логического умножения;}$$

$$x + y = y + x \text{ - для логического сложения;}$$

Сочетательный закон:

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \text{ - для логического умножения;}$$

$(x + y) + z = x + (y + z)$ -для логического сложения;

Распределительный закон умножения относительно сложения:

$$x \cdot (y + z) = x \cdot y + x \cdot z.$$

Распределительный закон сложения относительно умножения:

$$x + y \cdot z = (x + y) \cdot (x + z).$$

Закон повторения:

$$x \cdot x \dots = x \quad \text{и} \quad x + x + \dots = x$$

В этом случае цепь, состоящая из последовательного и параллельного соединения одинаковых контактов одного и того же релейного элемента, равносильна по своему действию одному контакту этого элемента.

Законы инверсии (законы де Моргана):

$$\overline{x \cdot y} = \overline{x} + \overline{y} \quad \text{-для логического умножения}$$

$$\overline{x + y} = \overline{x} \cdot \overline{y} \quad \text{-для логического сложения}$$

В соответствии с этим законом исходная и инверсная схемы связаны между собой так, что, когда исходная схема представляет собой замкнутую цепь, инверсная схема является разомкнутой, и наоборот. Следовательно, если исходная схема представляет собой постоянно замкнутую (равную единице) цепь, то инверсная ей схема будет постоянно разомкнутой (равной нулю). При инвертировании меняются на противоположные не только контакты, но и знаки логических действий, т.е. логическое умножение меняется на логическое сложение, а логическое сложение - на логическое умножение. На рис. 4.7 представлена релейно-контактная схема, поясняющая закон инверсии.

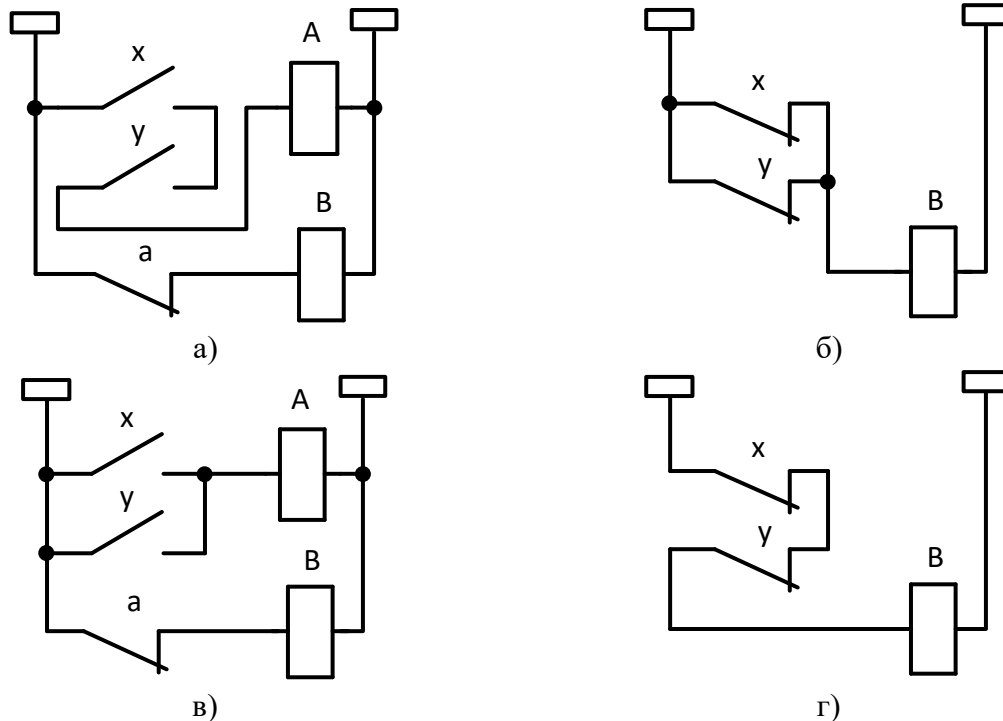


Рис. 4.7. Релейно-контактная схема для пояснения закона инверсии по выражениям (а) $\overline{x \cdot y}$, (б) $\overline{x} + \overline{y}$, (в) $x + y$, (г) $\overline{x} \cdot \overline{y}$

Закон поглощения для сложения и умножения:

$$x + x \cdot y = x;$$

$$x \cdot (x + y) = x.$$

Закон склеивания для сложения и умножения:

$$x + \bar{x} \cdot y = x + y,$$

Пример:

1. Составить релейно-контактную схему в соответствии с логическим выражением: $Y = (X1 \cdot X2 + \bar{X3} + X4) \cdot \overline{X1 + X2 \cdot X3} + X4$,

Решение: Релейно-контактная схема представлена на рис.4.8.

2. Составить таблицу истинности для выражения:

$$Y = (X1 \cdot X2 + \bar{X3} + X4) \cdot \overline{X1 + X2 \cdot X3} + X4,$$

Решение: Таблица истинности представлена в таблице 4.2.

Таблица 4.2

X1	X2	X3	X4	Y	X1	X2	X3	X4	Y
0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	1	0	1	0	1	1	0
0	1	0	0	0	1	1	0	0	0
0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	1	0	0
0	1	1	1	0	1	1	1	1	0

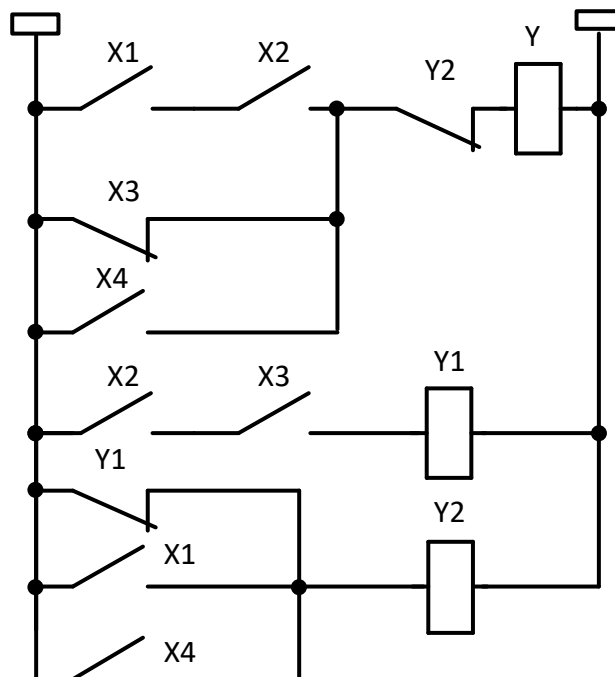


Рис. 4.8. Релейно-контактная схема

3. Оптимизировать логическое выражение по законам алгебры логики.

$$Y = (X1 \cdot X2 + \bar{X3} + X4) \cdot \overline{X1 + X2 \cdot X3} + X4,$$

$$Y = (X1 \cdot X2 + \bar{X3} + X4) \cdot (\overline{X1 + X2 \cdot X3} + X4),$$

$$Y = (X1 \cdot X2 + \overline{X3} + X4) \cdot (\overline{X1} \cdot X2 \cdot X3 \cdot \overline{X4}),$$

$$Y = X1 \cdot \overline{X1} \cdot X2 \cdot X2 \cdot X3 \cdot \overline{X4} + \overline{X3} \cdot X3 \cdot \overline{X1} \cdot X2 \cdot \overline{X4} + X4 \cdot \overline{X4} \cdot \overline{X1} \cdot X2 \cdot X3,$$

$$Y = 0 + 0 + 0,$$

$$Y = 0.$$

4. Составить релейно-контактную схему оптимизированного логического выражения.

Решение: Релейно-контактная схема представлена на рисунке 4.9.

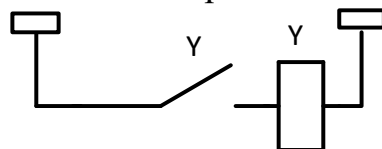


Рис. 4.9. Релейно-контактная схема

5. Составить таблицу истинности для оптимизированного выражения индивидуального задания.

Решение: Таблица истинности представлена в таблице 4.3.

Таблица 4.3

X1	X2	X3	X4	Y	X1	X2	X3	X4	Y
0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	1	0	1	0	1	1	0
0	1	0	0	0	1	1	0	0	0
0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	1	0	0
0	1	1	1	0	1	1	1	1	0

Состояние выходной величины не зависит от состояния входных величин

Контрольные вопросы

1. Какие основные функции алгебры логики?
2. Что такое таблица истинности?
3. Как представлен логический элемент «ИЛИ» на релейно-контактной схеме?
4. Как представлен логический элемент «И» на релейно-контактной схеме?
5. Как представлен логический элемент «НЕ» на релейно-контактной схеме?
6. Какие основные положения закона инверсии?

Практическая работа №5
Тема: «Основы проектирования систем управления»

Цель работы:

Изучить основы проектирования систем управления.

Задание:

1. Составить автоматную таблицу, заданной системы управления.
2. Ответить на контрольные вопросы.

Индивидуальное задание

Необходимо разработать систему управления конвекционным обогревом, состоящую из **A** элементов. При температуре больше **B** °C **H** элементов включены и выдаваться сигнал **F**. При температуре от **D** до **B** °C включены **J** элементов и выдаваться сигнал **N**, при температуре от **E** °C до **D** °C включены **K** элементов и выдаваться сигнал **V**, если температура ниже **E** °C должны быть включены **G** элементов и выдаваться сигнал **L**. Основным возмущающим воздействием является температура помещения. Исходные данные задачи представлены в таблице 5.1.

Таблица 5.1

Последняя цифра зачетной книжки	A	B	D	E	H	J	K	G	F	N	V	L
0	2	16	12	8	0	1	2	2	>T	-	-	-
1	3	22	15	10	1	2	3	3	-	Tнорм	-	-
2	4	25	10	8	2	3	4	4	-	-	Tнорм	
3	3	20	10	5	2	2	3	0	-	-	-	<T
4	2	25	16	5	0	1	2	2	>T	-	-	-
5	2	17	12	9	0	1	2	2	-	Tнорм	-	-
6	3	22	15	10	1	2	3	3	-	-	Tнорм	
7	4	27	10	8	2	3	3	4	-	-	-	<T
8	4	20	11	5	2	2	3	4	>T	-	-	-
9	2	25	16	7	1	1	2	2	-	Tнорм	-	-

Краткая теория

Любой естественный язык, несмотря на свою строгость и точность в его использовании, не обладает возможностью однозначного описания явлений и процессов и не позволяет непосредственно перейти от описания функционирования системы управления к ее программированию. Поэтому требуется формализованное описание функционирования автоматизированной системы управления, обеспечивающее непротиворечивость и полноту задания законов работы системы управления. Многие разработчики и программисты зачастую не используют никаких специальных формальных языков, а

непосредственно от технического задания, составленного на естественном языке, переходят к этапу программирования. Такой путь, в принципе, может привести к созданию качественной системы управления, однако разработчику придется часто вносить в проектируемую систему многочисленные исправления и добавления, обнаруживаемые по ходу проектирования.

Использование автоматного описания существенно облегчает программирование таких систем, состояние которых можно описать в виде совокупности режимов, каждый из которых определяется комбинацией входов и выходов. Если удалось однозначно определить состояние выходов для каждого из режимов, то система управления представляет собой комбинационную схему и для нее может быть применена теория автоматов.

Теория автоматов наиболее тесно связана с теорией алгоритмов: автомат преобразует дискретную информацию по шагам в дискретные моменты времени и формирует результат по шагам заданного алгоритма.

Изменение состояния дискретного автомата вызывается входными сигналами, возникающими вне автомата и передающимися по конечному числу входных каналов. Результатом работы дискретного автомата является выдача выходных сигналов, передаваемых во внешние цепи по конечному числу выходных каналов. Теория автоматов может быть использована для описания алгоритмических процессов, т.е. при проектировании автоматизированных систем управления.

В рамках теории автоматов применяются автоматные таблицы и направленные графы. Для простых централизованных АСУ ТП после составления автоматной таблицы можно переходить к этапу программирования, минуя этап разработки алгоритма. Для составления автоматной таблицы необходимо:

1. Построить модель «объект управления – система управления – оператор».
2. Определить совокупность точек, с которых в систему управления поступает необходимая информация (датчики, сигналы от оператора), и куда подаются управляющие воздействия.
3. Выделить комбинации входов системы управления, которые могут встретиться при функционировании объекта управления.
4. Обозначить режимы.
5. Для каждого режима записать состояние выходов системы управления.
6. Если удалось однозначно определить состояние выходов для каждого из режимов, то составляется автоматная таблица.
7. Если не удалось однозначно заполнить автоматную таблицу (у которой каждому входному набору однозначно соответствует единственный выходной набор), то это означает, что в системе управления недостаточно входов. В этом случае нужно увеличить число входов в систему управления добавлением новых датчиков или введением памяти.
8. При введении дополнительных датчиков повторяются пункты 1–6.

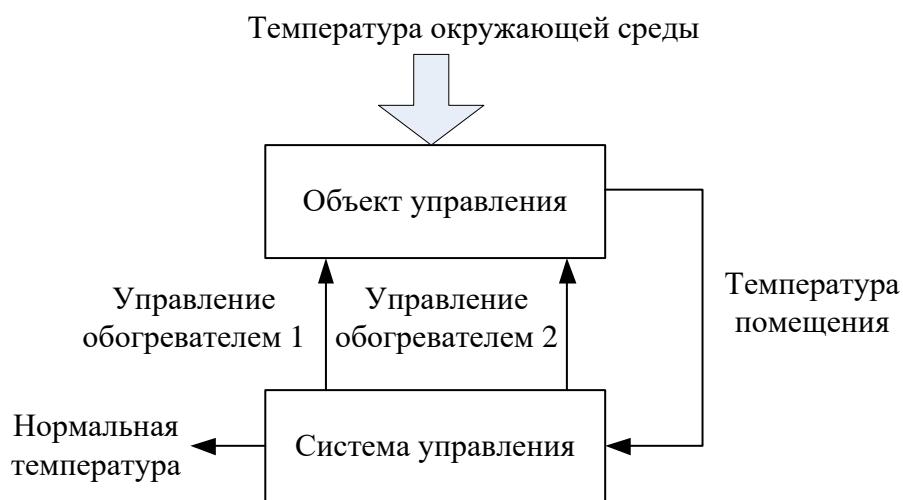
Если в результате получилась комбинационная схема, переходят к этапу программирования. Введение памяти позволяет фиксировать действия, которые система формировала в прошлом. Память в модели системы управления может

быть введена за счет добавления одного входа (состояние памяти в начале цикла), и выхода (со стояние памяти в конце цикла). Этот вход и выход разработчик использует по своему усмотрению для получения комбинационной схемы.

Пример.

Необходимо разработать систему управления конвекционным обогревом, состоящую из 2 обогревателей. При температуре в помещении больше 22 °С обогреватели отключены. При температуре от 16 до 22 °С включен 1 обогреватель и выдаваться сигнал «Нормальная температура», при температуре от 11 °С до 16 °С включены 2 обогревателя, если температура ниже 11 °С должны быть включены 2 обогревателя. Основным возмущающим воздействием является температура помещения.

1. Построить модель «объект управления – система управления – оператор».



2. Определить совокупность точек, с которых в систему управления поступает необходимая информация и куда подаются управляющие воздействия.

У системы управления один аналоговый вход (точка, куда поступает информация от датчика температуры), и три дискретных выхода для подачи управляющего воздействия первому и второму обогревателю, а также выход для сигнала «Нормальная температура». Обозначим вход – X1, а выходы Y1, Y2, Y3, соответственно управление обогревателем 1, управление обогревателем 2 и включение/отключение сигнала «Нормальная температура».

3. Выделить комбинации входов системы управления, которые могут встретиться при функционировании объекта управления.

Так как режим управления при температуре от 11 °С до 16 °С и ниже 11 °С определено одним алгоритмом управления - включены 2 обогревателя, то имеют рассматриваем следующие комбинации входов системы управления:

≥ 22 ; ≥ 16 , < 22 и < 16

4. Обозначить режимы.

Режимы управления, рассматриваемой системой, следующие:

Два обогревателя отключены; Один обогреватель включен и Сигнал «Нормальная температура»; Два обогревателя включены.

Комбинации входных сигналов, которые могут встретиться при функционировании системы управления, приведены в таблице 5.2.

Таблица 5.2

X	Режим
≥ 22	Два обогревателя отключены
$\geq 16, < 22$	Один обогреватель включен Сигнал «Нормальная температура»
< 16	Два обогревателя включены

5. Для каждого режима записать состояние выходов системы управления.

Комбинации выходных сигналов, которые могут встретиться при функционировании системы управления приведены в таблице 5.3.

Таблица 5.3

Y1	Y2	Y3	Режим
0	0	0	Два обогревателя отключены
0	0	1	-
0	1	0	-
0	1	1	-
1	0	0	-
1	0	1	Один обогреватель включен Сигнал «Нормальная температура»
1	1	0	Два обогревателя включены
1	1	1	-

6. Если удалось однозначно определить состояние выходов для каждого из режимов, то составляется автоматная таблица вида:

Состояние выходов однозначно определены. Таким образом на основании таблиц 5.2 и 5.3 строим автоматную таблицу (таблица 5.4).

Таблица 5.4

Входы	Выходы			Режим
X	Y1	Y2	Y3	
≥ 22	0	0	0	Два обогревателя отключены
$\geq 16, < 22$	1	0	1	Один обогреватель включен Сигнал «Нормальная температура»
< 16	1	1	0	Два обогревателя включены

Контрольные вопросы

1. Что такое дискретный автомат?
2. Какие этапы составления автоматной таблицы?
3. Для чего вводится контур памяти в автоматное описание работы автоматизированной системы управления?
4. Что такое датчик информации?
5. Как определяется оптимальное количество датчиков информации, необходимых для корректного описания работы автоматизированной системы управления?

Практическая работа №6

Тема: «Программирование логических контроллеров»

Цель работы:

Изучить основные характеристики программируемых логических контроллеров и получить практические навыки написания программного кода программируемых логических контроллеров.

Задание:

1. Изучить структурную и функциональную схему ПЛК.
2. Изучить основы программирования ПЛК с использованием специализированных программ.
3. Реализовать программу по логическому выражению индивидуального задания (практическое занятие № 4) на языке релейно-контактных схем (Л) с использованием специализированных программ.
4. Реализовать программу логическому выражению индивидуального задания (практическое занятие № 4) на языке диаграмм функциональных блоков (FBD) с использованием специализированных программ.
5. Реализовать программу по логическому выражению индивидуального задания (практическое занятие № 5) на языке релейно-контактных схем (Л) при условии наличия на входах дискретных температурных датчиков с использованием специализированных программ.
6. Реализовать программу логическому выражению индивидуального задания (практическое занятие № 5) на языке диаграмм функциональных блоков (FBD) при условии наличия на входах дискретных температурных датчиков с использованием специализированных программ.
7. Ответить на контрольные вопросы.

Краткая теория

Универсальные логические управляющие модули предназначены для замены традиционных схем управления, выполненных на основе реле, контакторов и подобных им устройств. Алгоритм функционирования модулей задается программой, составленной из набора встроенных функций, который включает в себя самые распространенные на практике логические функции (AND, OR, NOT и др.), а также специализированные функции. Примером может служить серия LOGO! Фирмы SIEMENS. Программирование модулей может осуществляться при помощи встроенной клавиатуры либо с ПЭВМ.

LOGO! Basic имеется для двух классов напряжения:

- Класс 1 < 24 В, т.е. 12 В пост. тока, 24 В пост. тока, 24 В перем. тока
- Класс 2 > 24 В, т.е. 115...240 В пост. и перем. Тока в вариантах:
- без дисплея: 8 входов и 4 выхода.
- с дисплеем: 8 входов и 4 выхода.

Каждый вариант занимает 4 места на профильной шине. Он снабжен интерфейсом для подключения расширения и предоставляет в ваше

распоряжение 30 готовых к использованию и специальных функций для разработки программы.

Каждое базовое устройство LOGO! Basic может быть расширено с помощью модулей расширения того же класса напряжения. Подключению друг к другу устройств, относящихся к различным классам напряжения, препятствует механическое кодирование (штифты в корпусе).

Исключение: левый интерфейс аналогового или коммуникационного модуля имеет потенциальную развязку. Поэтому эти модули расширения могут соединяться с устройствами различных классов напряжения.

Независимо от количества модулей, подключенных к LOGO!, в вашем распоряжении имеются следующие входы, выходы и биты памяти: I1 – I24, AI1 – AI8, Q1 – Q16 и M1 – M8.

Изменение состояния цепи $0 \rightarrow 1 / 1 \rightarrow 0$

Когда состояние цепи меняется с 0 на 1, состояние 1 и, в случае перехода от 1 к 0, состояние 0 должно сохраняться, по крайней мере, в течение одного цикла обработки программы, чтобы LOGO! мог распознать новое состояние цепи. Время цикла обработки программы зависит от величины программы.

Для подключения датчиков к LOGO! необходимо использовать схему, представленную на рис. 6.1.

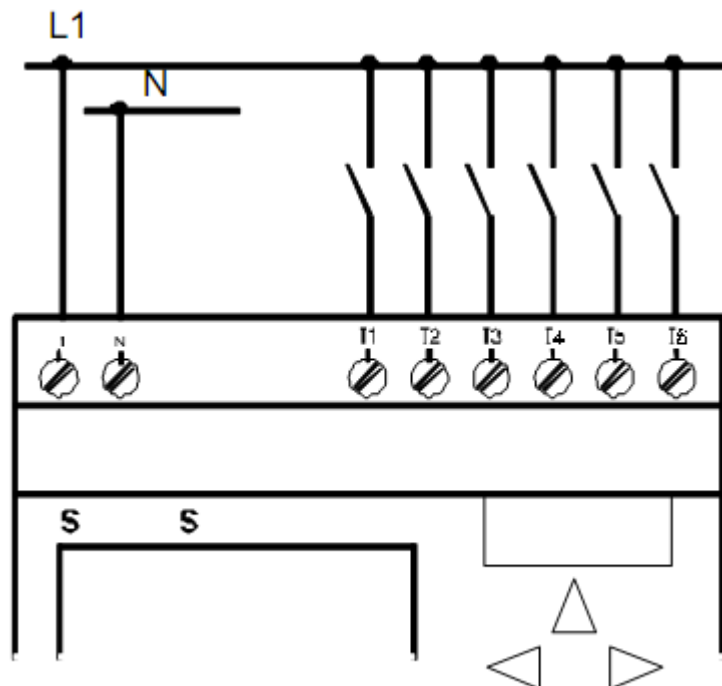


Рис. 6.1. Схема подключение датчиков к LOGO!

Выходами LOGO! являются реле. Контакты реле изолированы от источника питания и входов.

К выходам можно подключать различные нагрузки, например, лампы, в том числе люминесцентные, двигатели, контакторы и т.д.

Подключение

Для подключения нагрузки к вариантам LOGO! необходимо использовать схему, представленную на рис. 6.2.

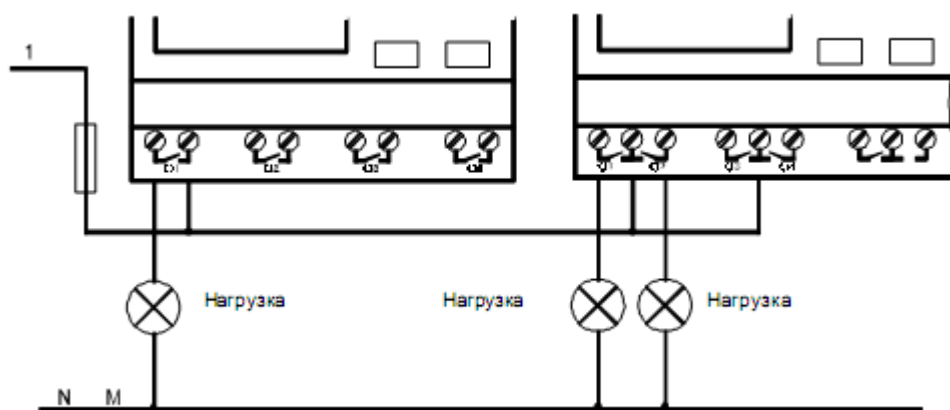


Рис. 6.2. Схема подключение нагрузки к LOGO!

Языки программирования стандарта МЭК 61131-3.

Международный стандарт МЭК 61131-3, описывающий языки программирования для программируемых логических контроллеров (ПЛК). Стандарт МЭК 61131-3 устанавливает пять языков программирования ПЛК, три графических и два текстовых

Первоначально стандарт назывался IEC 1131-3 и был опубликован в 1993 г., но в 1997 г. МЭК (IEC) перешел на новую систему обозначений и в названии стандарта добавилась цифра «6». Продвижением стандарта занимается организация PLCopen (<http://www.plcopen.org>).

Основной целью стандарта было повышение скорости и качества разработки программ для ПЛК, а также создание языков программирования, ориентированных на технологов, обеспечение соответствия ПЛК идеологии открытых систем, исключение этапа дополнительного обучения при смене типа ПЛК.

Системы программирования, основанные на МЭК 61131-3, характеризуются следующими показателями:

- надежностью создаваемого программного обеспечения. Надежность обеспечивается тем, что программы для ПЛК создаются с помощью специально предназначенной для этого среды разработки, которая содержит все необходимые средства для написания, тестирования и отладки программ с помощью эмуляторов и реальных ПЛК, а также множество готовых фрагментов программного кода
- возможностью простой модификации программы и наращивания ее функциональности
- переносимостью проекта с одного ПЛК на другой
- возможностью повторного использования отработанных фрагментов программы
- простотой языка и ограничением количества его элементов

Языки МЭК 61131-3 появились не как теоретическая разработка, а как результат анализа множества языков, уже используемых на практике и предлагаемых рынку производителями ПЛК.

Стандарт устанавливает пять языков программирования со следующими названиями:

1. список инструкций (IL — Instruction List)
2. структурированный текст (ST — Structured Text)
3. релейно-контактные схемы, или релейные диаграммы (LD — Ladder Diagram)
4. диаграммы функциональных блоков (FBD — Function Block Diagram)
5. последовательные функциональные схемы (SFC — «Sequential Function Chart»)

Языки IL и ST являются текстовыми. Графическими языками являются LD, FBD, SFC.

В стандарт были введены несколько языков (а не один) для того, чтобы каждый пользователь мог применить наиболее понятный ему язык.

Программисты чаще выбирают язык IL (похожий на ассемблер) или ST, похожий на язык высокого уровня Паскаль. Специалисты, имеющие опыт работы с релейной логикой, выбирают язык LD. Специалисты по системам автоматического управления (САУ) и схемотехники выбирают привычный для них язык FBD.

Выбор одного из пяти языков определятся не только предпочтениями пользователя, но и смыслом решаемой задачи.

Если исходная задача формулируется в терминах последовательной обработки и передачи сигналов, то для нее проще и нагляднее использовать язык FBD. Если задача описывается как последовательность срабатываний некоторых ключей и реле, то для нее нагляднее всего будет язык LD. Для задач, которые изначально формулируются в виде сложного разветвленного алгоритма, удобнее будет язык ST.

Языки МЭК 61131-3 базируются на следующих принципах:

- вся программа разбивается на множество функциональных элементов — Program Organization Units (POU), каждый из которых может состоять из функций, функциональных блоков и программ. Любой элемент МЭК-программы может быть сконструирован иерархически из более простых элементов

- стандарт требует строгой типизации данных. Указание типов данных позволяет легко обнаруживать большинство ошибок в программе до ее исполнения

- имеются средства для исполнения разных фрагментов программы в разное время, с разной скоростью, а также параллельно. Например, один фрагмент программы может сканировать концевой датчик с частотой 100 раз в секунду, в

то время как второй фрагмент будет сканировать датчик температуры с частотой один раз в 10 сек

- для выполнения операций в определенной последовательности, которая задается моментами времени или событиями, используется специальный язык последовательных функциональных схем (SFC)

- стандарт поддерживает структуры для описания разнородных данных. Например, температуру подшипников насоса, давление и состояние «включено-выключено» можно описать с помощью единой структуры «Pump» и передавать ее внутри программы как единый элемент данных

- стандарт обеспечивает совместное использование всех пяти языков, поэтому для каждого фрагмента задачи может быть выбран любой, наиболее удобный, язык

Список инструкций, IL

Язык IL напоминает ассемблер и используется для реализации функций, функциональных блоков и программ, а также шагов и переходов в языке SFC. Основным достоинством языка является простота его изучения.

Наиболее часто язык IL используется в случаях, когда требуется получить оптимизированный код для реализации критических секций программы, а также для решения небольших задач с малым количеством разветвлений алгоритма.

В основе языка лежит понятие аккумулятора и переходов по меткам.

Пример программы на языке IL с комментариями приведен в листинге 6.1.

Начинается программа с загрузки в аккумулятор значения переменной. Дальнейшие шаги программы состоят в извлечении содержимого аккумулятора и выполнении над ним ограниченного числа допустимых действий (их в языке всего 24).

Листинг 6.1. Пример программы на языке IL

Метки	Операторы	Операнды	Комментарии
	LD	Voltage	(*Загрузить Voltage в аккумулятор*)
	GT	220	(*Если >220*)
	JMPCN	M1	(*Перейти к метке, если ">220" не верно*)
	LD	Current	(*Загрузить Current в аккумулятор*)
	SUB	10	(*Вычесть из аккумулятора 10 *)
	ST	Current	(*Присвоить Current значен. аккумулятора*)
M1:	LD	0	(*Загрузить в аккумулятор значение "0"*)
	ST	Out	(*Присвоить Out значение аккумулятора*)

Структурированный текст, ST

Язык ST является текстовым языком высокого уровня и очень сильно напоминает Паскаль:

Листинг 6.2. Пример программы на языке ST

```
IF Voltage>220 THEN  
Current:=Current - 10; (*Если V>220 В, то уменьшить ток на 10*)  
ELSE  
Current:=50; Speed:= ON;(*Установить ток 50А и включить мотор*)  
END_IF;
```

Язык ST имеет много отличий от языка Паскаль и разработан специально для программирования ПЛК. Он содержит множество конструкций для присвоения значений переменным, для вызова функций и функциональных блоков, для написания выражений условных переходов, выбора операторов, для построения итерационных процессов.

Этот язык предназначен в основном для выполнения сложных математических вычислений, описания сложных функций, функциональных блоков и программ.

Язык релейно-контактных схем, LD

Графический язык релейной логики впервые появился в виде электрических схем, которые состояли из контактов и обмоток электромагнитных реле (рис. 6.3).

Язык релейной логики интуитивно понятен людям, слегка знакомым с электротехникой и поэтому оказался наиболее распространенным в промышленной автоматике. Обслуживающий персонал легко находил отказ в оборудовании, прослеживая путь сигнала по релейной диаграмме.

Однако язык LD проблематично использовать для реализации сложных алгоритмов, поскольку он не поддерживает подпрограммы, функции, инкапсуляцию и другие средства структурирования программ с целью повышения качества программирования. Эти недостатки затрудняют многократное использование программных компонентов, что делает программу длинной и сложной для обслуживания.

Для выполнения арифметических функций в язык LD были добавлены функциональные блоки, которые выполняли операции сложения, умножения, вычисления среднего и т.д.

Сложные вычисления в этом языке невозможны. Недостатком является также то, что только маленькая часть программы умещается на мониторе компьютера или панели оператора при программировании.

Несмотря на указанные недостатки, язык LD относится к наиболее распространенным в мире, хотя используется для программирования только простых задач.

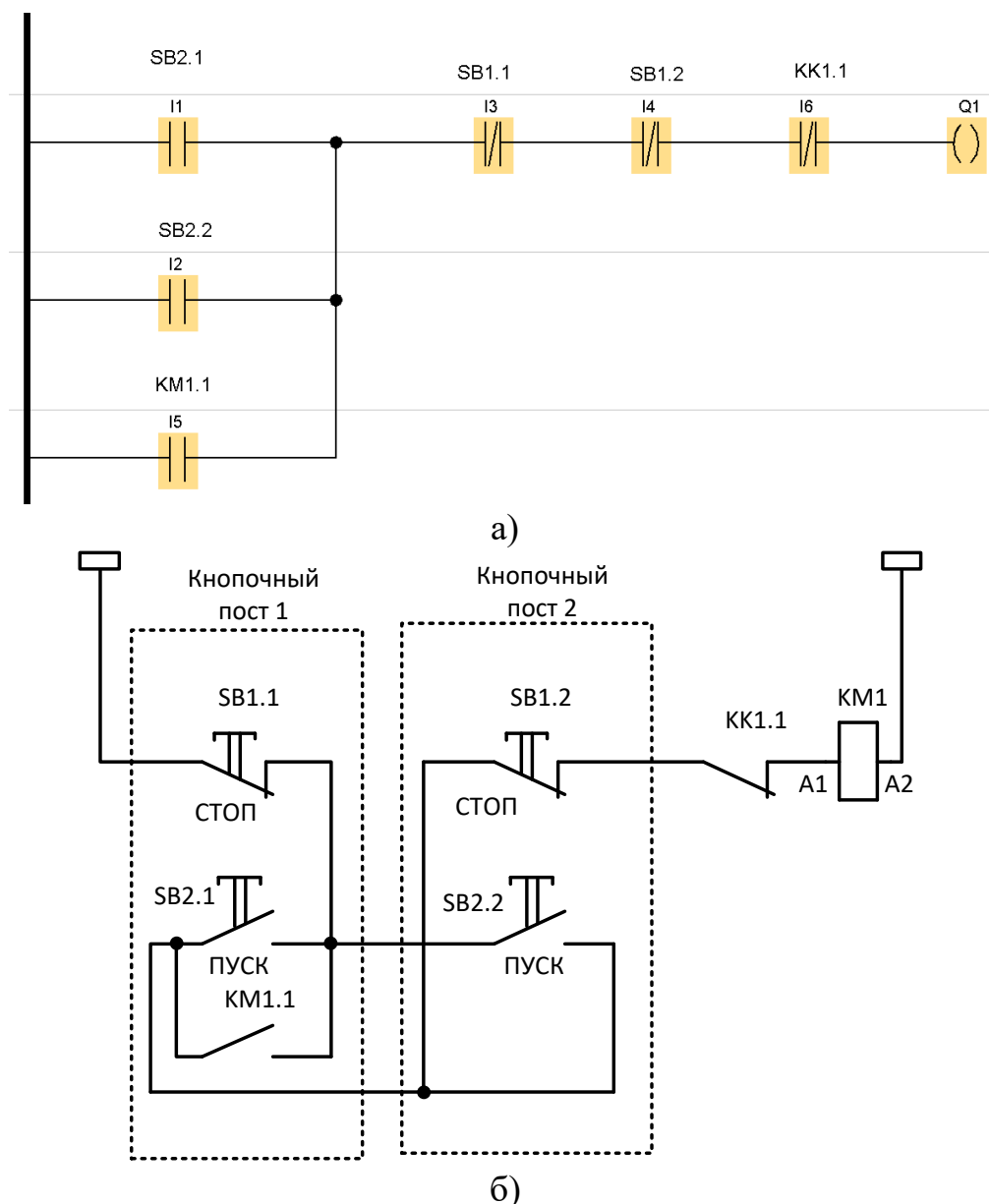


Рис. 6.3. Пример программы на языке LD (а) и ее эквивалент в виде электрической цепи с реле (б)

Диаграммы функциональных блоков, FBD

FBD является графическим языком и наиболее удобен для программирования процессов прохождения сигналов через функциональные блоки.

Язык FBD удобен для схемотехников, которые легко могут составить электрическую схему системы управления на «жесткой логике», но не имеют опыта программирования.

Функциональные блоки представляют собой фрагменты программ, написанных на IL, SFC или других языках, которые могут быть многократно использованы в разных частях программы и которым соответствует графическое изображение, принятое при разработке функциональных схем электронных устройств, см. рис. 6.4.

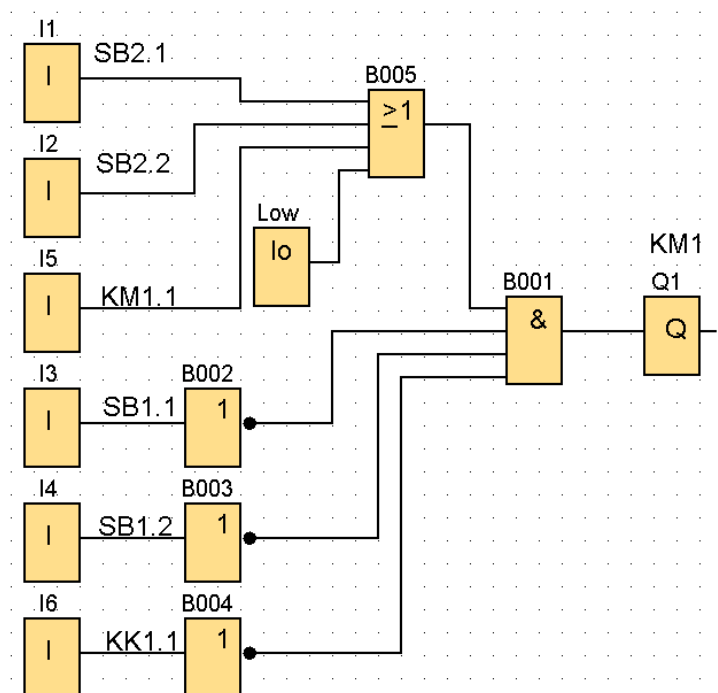


Рис. 6.4. Пример программы на языке FBD для схемы на рисунке 6.3 б

Язык FBD может быть использован для программирования функций, функциональных блоков и программ, а также для описания шагов и переходов в языке SFC. Функциональные блоки инкапсулируют данные и методы, чем напоминают объектно-ориентированные языки программирования, но не поддерживают наследование и полиморфизм.

Все то, что во время компиляции или исполнения программы может содержать или обрабатывать значения различных типов — является полиморфным, например:

- переменные, меняющие свое значение на значение другого типа
- объекты, обладающие свойствами, которые могут менять значение текущего типа на значение другого типа
- функции, принимающие аргументы различных типов

Типичным применением языка FBD является описание «жесткой логики» и замкнутых контуров систем управления.

Язык функциональных блоков является удобным также для создания и пополнения библиотеки типовых функциональных блоков, которую можно многократно использовать при программировании задач промышленной автоматизации.

Последовательные функциональные схемы, SFC

SFC называют языком программирования, хотя по сути это не язык, а вспомогательное средство для структурирования программ.

Он предназначен специально для программирования последовательности выполнения действий системой управления, когда эти действия должны быть выполнены в заданные моменты времени или при наступлении некоторых событий. В его основе лежит представление системы управления с помощью понятий состояний и переходов между ними.

Язык SFC может быть использован также для программирования отдельных функциональных блоков, если алгоритм их работы естественным образом описывается с помощью понятий состояний и переходов.

На рисунке 6.3 показан фрагмент программы на языке SFC.

Программа состоит из шагов и условий переходов.

Шаги показываются на схеме прямоугольниками, условия переходов— жирной перечеркивающей линией.

Программа выполняется сверху вниз.

Начальный шаг на схеме показывается в виде двойного прямоугольника.

Условия переходов записываются рядом с их обозначениями.

Каждый шаг программы может представлять собой реализацию сложного алгоритма, написанного на одном из МЭК-языков.

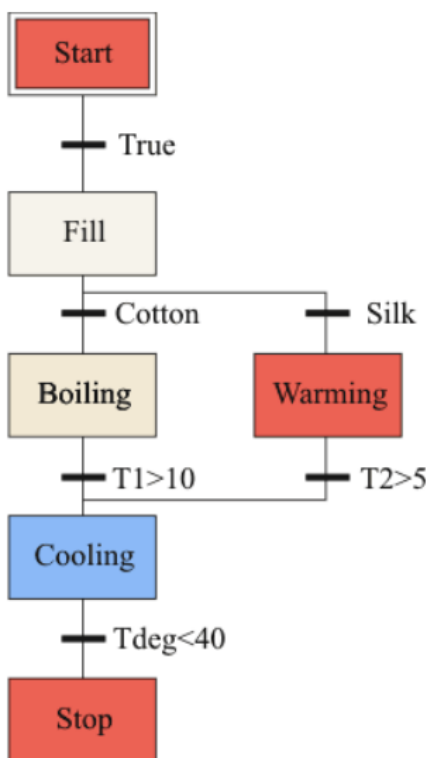


Рис. 6.3.Пример программы на языке SFC

Пример.

1. Реализовать программу по логическому выражению $Y = (X1 \cdot X2 + \overline{X3} + X4) \cdot X1 + \overline{X2} \cdot X3 + X4$, на языке релейно-контактных схем (Л) с использованием специализированных программ.

Реализация программы представлена на рисунке 6.4.

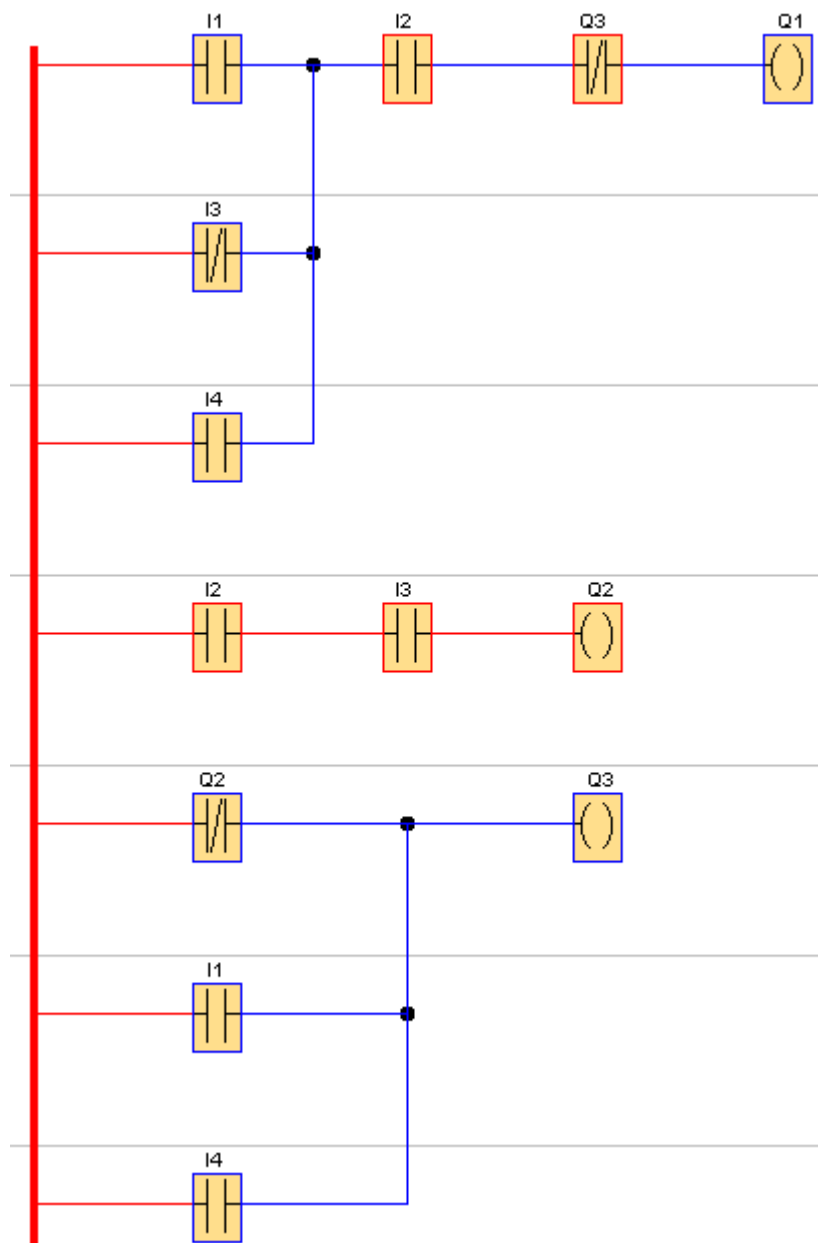


Рис. 6.4. Реализация программы на языке релейно-контактных схем (IL)

2. Реализовать программу логическому выражению $Y = (X1 \cdot X2 + \overline{X3} + X4) \cdot \overline{X1} + \overline{X2} \cdot \overline{X3} + X4$, на языке диаграмм функциональных блоков (FBD) с использованием специализированных программ. Реализация программы представлена на рисунке 6.5.

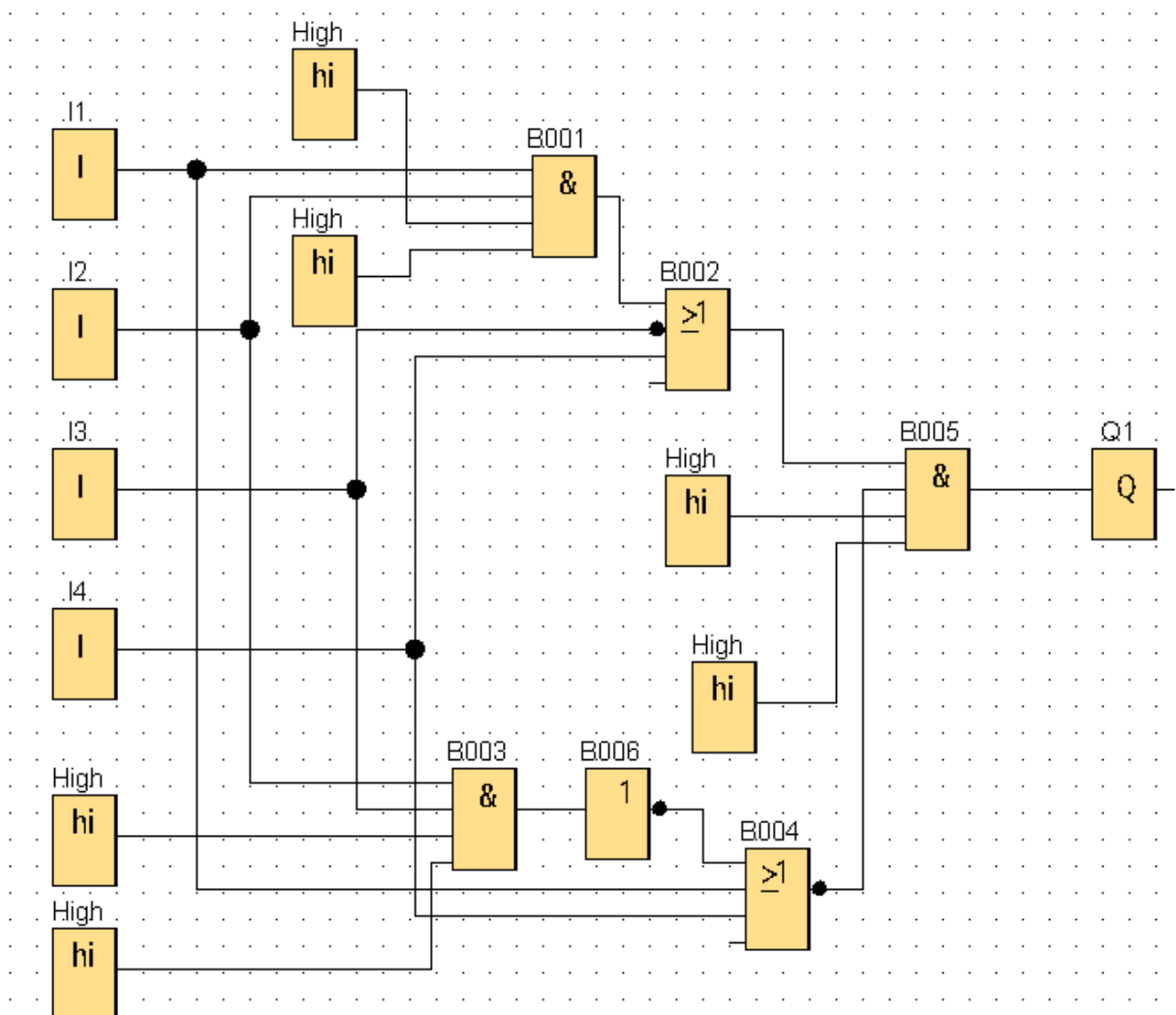


Рис. 6.4. Реализация программы на языке диаграмм функциональных блоков

Контрольные вопросы

1. Какие языки программирования ПЛК регламентирует МЭК 61131-3?
2. Какая форма записи программного кода языка релейно-контактных схем, LD?
3. Какая форма записи программного кода языка диаграмм функциональных блоков, FBD?
4. Преимущества программирования ПЛК на языке FBD от языка LD?

Практическая работа №7

Тема: «Автоматизированные схемы на аппаратной платформе Arduino»

Цель работы: Изучить основные принципы реализации схемного и программного решения микропроцессорных устройств.

Задание:

1. Изучить характеристики аппаратной платформы Arduino.
2. Провести анализ вариантов построения микропроцессорных устройств для реализации автоматизированных схем управления.
3. Получить практические навыки разработки автоматизированных систем с использованием аппаратной платформы Arduino.
4. Ответить на контрольные вопросы

Индивидуальное задание представлено в таблице 7.1.

Таблица 7.1

№ по порядку в списке группы	Задача
0	Разработать систему включения светодиода с двумя кнопками: Включается SB1, отключается SB2
1	Разработать систему включения светодиода с двумя кнопками: Включается SB2, отключается SB1 и SB2
2	Разработать систему визуального контроля уровня яркости светодиода на дисплее LCD1602 по I2C с потенциометром
3	Разработать систему визуального контроля уровня шума (с использование микрофона) по яркости светодиода
4	Разработать систему обратного отсчета времени (10 секунд) при запуске от кнопки и визуального контроля по светодиоду (светится)
5	Разработать систему контроля времени нажатого состояния кнопки и визуального контроля на дисплее LCD1602 по I2C времени
6	Разработать систему программной фиксации (запоминания) нажатого состояния кнопки SB1 и визуального контроля по состоянию светодиода (нажата-светится), SB2 сбрасывает фиксацию
7	Разработать систему работы светофора - светодиоды (красный-желтый-зеленый) включаются по очереди с К-1с-КЖ-0,5с-Ж-1с-З-1с-ЗЖ-0,5с
8	Разработать систему работы уровня звука (с использованием микрофона) - светодиоды (красный-желтый-зеленый) включаются по условию З (0-30%)-ЗЖ(30-60%)-ЗЖК(60-100%)
9	Разработать систему светоиндикации «бегущая волна» - светодиоды (1,2,3,4,5-5шт) включаются в следующей последовательности: 123-0,2с-234-0,2с-345-0,2с-451-0,2с-512-0,2с

Краткая теория

Arduino — это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и простоте языка программирования, а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов.

Arduino позволяет компьютеру выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.

Микроконтроллер на плате программируется при помощи языка Arduino (основан на языке Wiring) и среды разработки Arduino (основана на среде Processing). Проекты устройств, основанные на Arduino, могут работать самостоятельно, либо же взаимодействовать с программным обеспечением на компьютере (напр.: Flash, Processing, MaxMSP). Платы могут быть собраны пользователем самостоятельно или куплены в сборе. Программное обеспечение доступно для бесплатного скачивания. Исходные чертежи схем (файлы CAD) являются общедоступными, пользователи могут применять их по своему усмотрению.

Существует множество разновидностей плат Arduino, которые используются для разных целей. Некоторые платы отличаются от представленной ниже UNO, но большинство Arduino имеют много общих компонентов.

Uno снабжён различными типами выводов. Каждый вывод имеет маркировку на плате и используется для различных функций.

Выводы питания (USB / Power jack/Vin)

Контакты 5 В, 3,3 В, GND, аналоговый, цифровой, ШИМ, AREF

Все контакты UNO имеют чёрные пластиковые разъёмы. Они позволяют просто подключить провод прямо к плате без пайки.

Питание

Arduino Uno может получать питание через подключение USB или от внешнего источника питания. Источник питания выбирается автоматически.

Внешнее питание (не USB) может подаваться через преобразователь напряжения AC/DC (блок питания) или аккумуляторной батареей. Преобразователь напряжения подключается посредством разъема 2.1 мм с центральным положительным полюсом. Провода от батареи подключаются к выводам Gnd и Vin разъема питания.

Платформа может работать при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, при этом платформа может работать нестабильно. При использовании напряжения выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

Выводы питания:

VIN. Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB или другого регулируемого источника питания). Подача напряжения питания происходит через данный вывод.

5V. Регулируемый источник напряжения, используемый для питания микроконтроллера и компонентов на плате. Питание может подаваться от вывода VIN через регулятор напряжения, или от разъема USB, или другого регулируемого источника напряжения 5 В.

3V3. Напряжение на выводе 3.3 В генерируемое встроенным регулятором на плате. Максимальное потребление тока 50 мА.

GND. Выводы заземления.

Память

Микроконтроллер ATmega328 располагает 32 кБ флэш памяти, из которых 0.5 кБ используется для хранения загрузчика, а также 2 кБ ОЗУ (SRAM) и 1 Кб EEPROM (которая читается и записывается с помощью библиотеки EEPROM).

Входы и Выходы

Каждый из 14 цифровых выводов Uno может настроен как вход или выход, используя функции pinMode(), digitalWrite(), и digitalRead(), . Выводы работают при напряжении 5 В. Каждый вывод имеет нагрузочный резистор (по умолчанию отключен) 20-50 кОм и может пропускать до 40 мА. Некоторые выводы имеют особые функции:

Последовательная шина: 0 (RX) и 1 (TX). Выводы используются для получения (RX) и передачи (TX) данных TTL. Данные выводы подключены к соответствующим выводам микросхемы последовательной шины ATmega8U2 USB-to-TTL.

Внешнее прерывание: 2 и 3. Данные выводы могут быть сконфигурированы на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Подробная информация находится в описании функции attachInterrupt().

ШИМ: 3, 5, 6, 9, 10, и 11. Любой из выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции analogWrite().

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Посредством данных выводов осуществляется связь SPI, для чего используется библиотека SPI.

LED: 13. Встроенный светодиод, подключенный к цифровому выводу 13. Если значение на выводе имеет высокий потенциал, то светодиод горит.

На платформе Uno установлены 6 аналоговых входов (обозначенных как A0 .. A5), каждый разрешением 10 бит (т.е. может принимать 1024 различных значения). Стандартно выводы имеют диапазон измерения до 5 В относительно земли, тем не менее имеется возможность изменить верхний предел посредством вывода AREF и функции analogReference(). Некоторые выводы имеют дополнительные функции:

I2C: 4 (SDA) и 5 (SCL). Посредством выводов осуществляется связь I2C (TWI), для создания которой используется библиотека Wire.

Дополнительная пара выводов платформы:

AREF. Опорное напряжение для аналоговых входов. Используется с функцией analogReference().

Reset. Низкий уровень сигнала на выводе перезагружает микроконтроллер. Обычно применяется для подключения кнопки перезагрузки на плате расширения, закрывающей доступ к кнопке на самой плате Arduino.

Обратите внимание на соединение между выводами Arduino и портами ATmega328.

Связь

На платформе Arduino Uno установлено несколько устройств для осуществления связи с компьютером, другими устройствами Arduino или микроконтроллерами. ATmega328 поддерживают последовательный интерфейс UART TTL (5 В), осуществляемый выводами 0 (RX) и 1 (TX). Установленная на плате микросхема ATmega8U2 направляет данный интерфейс через USB, программы на стороне компьютера "общаются" с платой через виртуальный COM порт. Прошивка ATmega8U2 использует стандартные драйвера USB COM, никаких сторонних драйверов не требуется, но на Windows для подключения потребуется файл ArduinoUNO.inf. Мониторинг последовательной шины (Serial Monitor) программы Arduino позволяет посылать и получать текстовые данные при подключении к платформе. Светодиоды RX и TX на платформе будут мигать при передаче данных через микросхему FTDI или USB подключение (но не при использовании последовательной передачи через выводы 0 и 1).

Библиотекой SoftwareSerial возможно создать последовательную передачу данных через любой из цифровых выводов Uno.

ATmega328 поддерживает интерфейсы I2C (TWI) и SPI. В Arduino включена библиотека Wire для удобства использования шины I2C.

Программирование

Платформа программируется посредством ПО Arduino. Из меню Tools > Board выбирается «Arduino Uno» (согласно установленному микроконтроллеру). Подробная информация находится в справочнике и инструкциях.

Микроконтроллер ATmega328 поставляется с записанным загрузчиком, облегчающим запись новых программ без использования внешних программаторов. Связь осуществляется оригинальным протоколом STK500.

Имеется возможность не использовать загрузчик и запрограммировать микроконтроллер через выводы ICSP (внутрисхемное программирование). Подробная информация находится в данной инструкции.

Автоматическая (программная) перезагрузка

Uno разработана таким образом, чтобы перед записью нового кода перезагрузка осуществлялась самой программой Arduino на компьютере, а не нажатием кнопки на платформе. Одна из линий DTR микросхемы ATmega8U2, управляющих потоком данных (DTR), подключена к выводу перезагрузки микроконтроллеру ATmega328 через 100 нФ конденсатор. Активация данной линии, т.е. подача сигнала низкого уровня, перезагружает микроконтроллер. Программа Arduino, используя данную функцию, загружает код одним нажатием кнопки Upload в самой среде программирования. подача сигнала низкого уровня по линии DTR скоординирована с началом записи кода, что сокращает таймаут загрузчика.

Функция имеет еще одно применение. Перегрузка Uno происходит каждый раз при подключении к программе Arduino на компьютере с ОС Mac X или Linux (через USB). Следующие полсекунды после перегрузки работает загрузчик. Во время программирования происходит задержка нескольких первых байтов кода во избежание получения платформой некорректных данных (всех, кроме кода новой программы). Если производится разовая отладка скетча, записанного в платформу, или ввод каких-либо других данных при первом запуске, необходимо убедиться, что программа на компьютере ожидает в течение секунды перед передачей данных.

На Uno имеется возможность отключить линию автоматической перезагрузки разрывом соответствующей линии. Контакты микросхем с обоих концов линии могут быть соединены с целью восстановления. Линия маркирована «RESET-EN». Отключить автоматическую перезагрузку также возможно подключив резистор 110 Ом между источником 5 В и данной линией.

Задача №1

Условие: Разработать систему регулирования яркости светодиода двумя кнопками

Примечание: Интенсивность свечения будет регулироваться от 0 до 254 единиц, где 0 - светодиод выключен, а 254 - горит максимально (Например при 127 яркость будет на 50%).

Приборы и материалы:

Ардуино, Бредборд, Провода, Светодиод, Кнопки, Резисторы

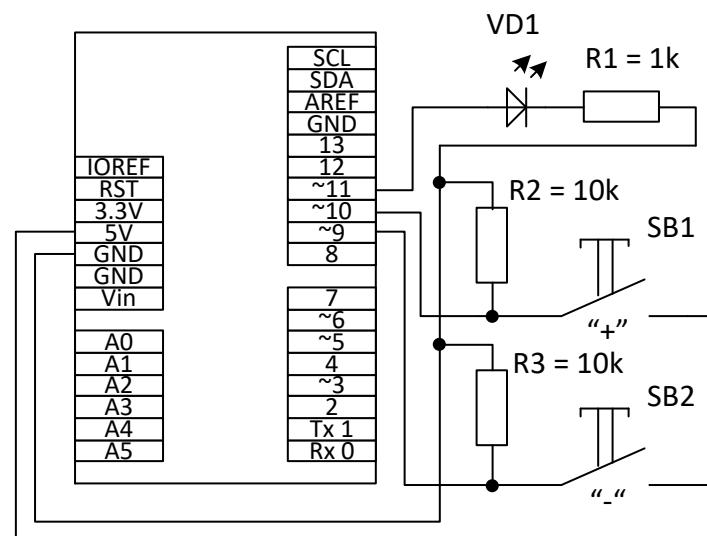


Рис. 7.2. Вариант схемы подключения по задаче №1

Листинг 7.1

Пример sketch для задачи №1

```
int led = 11;    // Номер Pin к которому подключен диод
int brightness = 0; //Переменная в которой хранится уровень яркости (От 0 до 254)
int buttonPlus = 9; // Номер Pin к которому подключена кнопка
int buttonMinus = 10; // Номер Pin к которому подключена кнопка -
```

```

void setup() {
  pinMode(led, OUTPUT); // Порт 11 (led) будет работать как Выход.
}
void loop() { // Этот цикл будет выполняться бесконечное количество раз.
  if (digitalRead(buttonPlus) == HIGH) {
    brightness += 5;
  } // Делаем проверку, если вход под номером 9 (buttonPlus) имеет состояние
5В. Увеличиваем значение переменной яркости на 5 единиц.
  if (digitalRead(buttonMinus) == HIGH) {
    brightness -= 5;
  } // Делаем проверку, если вход под номером 10 (buttonMinus) имеет
состояние 5 В. Уменьшаем значение переменной яркости на 5 единиц.
  brightness = constrain(brightness, 0, 254); // Эта функция контролирует, что бы
переменная brightness не стала больше 254 и меньше 0, если значение вылезит
за границу то функция 0 или 254
  analogWrite(led, brightness); // Устанавливаем состояние яркости
  delay(50); // Пауза 50 миллисекунд.
}

```

Задача №2

Условие: Разработать систему визуального контроля показания потенциометра на дисплее LCD1602 по I2C с Ардуино

Примечание: Интенсивность свечения будет регулироваться от 0 до 254 единиц, где 0 - светодиод выключен, а 254 - горит максимально (Например при 127 яркость будет на 50%).

Приборы и материалы:

Ардуино, Провода, LCD1602 по I2C, Потенциометр.

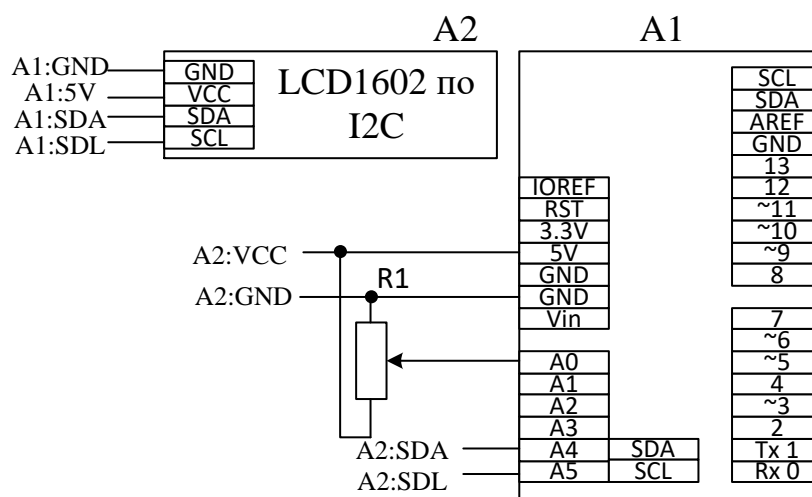


Рис. 7.3. Вариант схемы подключения по задаче №2

Листинг 7.2

Пример sketch для задачи №2

```
#include <Wire.h> //Подключаем библиотеку Wire для связи по I2C интерфейсу
```



```

#include <LiquidCrystal_I2C.h> //Подключаем библиотеку LiquidCrystal I2C для LCD
дисплеев.
int PotenciometrPin = A0; // Номер Pin к которому подключен потенциометр
int Value=0; //переменная для хранения данных о состоянии потенциометра
float Value_volt=0; //переменная для расчетного значения данных о состоянии
потенциометра
LiquidCrystal_I2C lcd(0x27,16,2); // Устанавливаем дисплей
void setup()
{
  lcd.init();
  lcd.backlight();// Включаем подсветку дисплея
  lcd.print("DSTU");
  lcd.setCursor(4, 1);
  lcd.print("Potenciometr");
  delay(1000);
}
void loop() // Этот цикл будет выполняться бесконечное количество раз.
{
  lcd.setCursor(0, 1);
  lcd.print("    - ____V  "); // Устанавливаем курсор на вторую строку и
нулевой символ.
  lcd.setCursor(0, 1); // Выводим на экран значение с аналогового входа (от 0 до
1024 пропорционально от 0 В. до 5 В.)
  Value = analogRead(PotenciometrPin);
  lcd.print(Value);
  lcd.setCursor(7, 1);
  Value_volt=(float)5/1024*Value;
  lcd.print(Value_volt);
  delay(300);
}

```

Контрольные вопросы

1. Какое отличие дискретного сигнала от аналогового?
2. Что такое датчик?
3. Как в среде Arduino определяются входные и выходные сигналы?
4. Какие еще электронные конструкторы вы знаете?

ЛИТЕРАТУРА

1. Единая система конструкторской документации. Обозначения буквенно-цифровые в электрических схемах. ГОСТ 2.710-81.
2. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения. - ГОСТ 34.003-90.
3. Бородин И.Ф., Судник Ю.А. Автоматизация технологических процессов.- М: КолосС., 2004.
4. Дудников Е.Г. Лёвин А.А. Промышленные автоматизированные системы управления. -М.: «Энергия», 1973.
5. Втюрин В.А. Автоматизированные системы управления технологическими процессами. Основы АСУТП. - СПб.: СПб ГЛА имени Кирова, 2006.
6. Захаров В.Н. , Поспелов Д.А., Хазацкий В.Е. Системы управления. Задание. Проектирование. Реализация. Изд. 2-е, перераб. и доп.- М.: «Энергия», 1977.
7. Сорокин С. Системы реального времени.- Современные технологии автоматизации, №2, 1997.
8. Тимофеев Б.Б., Козлик Г.А., Кулаков А.Ф., Мартыанов. А.И. Алгоритмизация в автоматизированных системах управления.- Киев, «Техніка», 1972
9. Черняховский Н.И. Элементы электрических схем.- Тольятти: ТГУ, 2007.
10. Пестриков В.М., Дудкин В.С., Петров Г.А. Дискретная математика.- СПб.: СПб ГТУРП, 2013.
11. Бармин А. Устройства локальной автоматики. Микроконтроллеры.- Современные технологии автоматизации, № 4, 2003
12. SIMATIC LOGO! Руководство пользователя. ред. A5E00119092–01 09/01.
13. Среда программирования OWEN Logic. Руководство пользователя. Москва, 2011